

OCR A Level Computing Coursework

Tireni Oluwabunmi

Analysis:	4
Problem Identification:	4
Features:	4
Stakeholders:	4
Reasons For Why It's Solvable By Computational Methods:	5
The need for simplification:	5
Breaking down the problem:	6
Need for divide and conquer:	7
The Output:	7
Interview:	7
Questions:	8
Answers:	9
Analysis Of Answers:	10
Researching existing similar problems:	11
Product 1:	11
Possible adaptations:	11
Possible adaptations:	13
Product 3:	13
Possible adaptations:	13
Product 4:	14
Possible adaptations:	14
Product 5:	15
Possible adaptations:	15
Initial Concept Of Solution:	16
Limitations of my solution:	16
Interview 2:	17
Email Sent:	17
Replies:	18
Analysis Of Answers:	18
Abstract Diagram:	19
Analysis Of Abstraction Diagram:	19
Stakeholder Requirements:	20
Design:	20
Functionality:	21
Hardware And Software :	24
Success Criteria:	25
Design:	26
Brief Overview:	26
Sub Program Algorithms:	31
Sign Up:	31
Log In:	32

Room Status:	33
Pin:	33
Pins:	34
Search:	35
Route Creation:	36
Menu:	37
Screen Designs:	38
Key Variables:	48
Data Structures Explained:	89
Validation Table:	90
Test Table Alpha(During Development):	90
Test Table Beta (Post Development):	95
Development:	100
Breakdown:	100
Task 1:	101
Development:	101
Client Review:	120
Review Modifications:	122
Task 2:	124
Development:	124
Client Review:	146
Review Modifications:	148
Task 3:	148
Development:	148
Client Review:	172
Review Modifications:	175
Task 4:	177
Development:	177
Client Review:	203
Review Modifications:	208
Evaluation:	208
Beta Testing:	209
Usability features:	238
Questionnaire:	238
Questionnaire Feedback:	238
Possible Improvements:	239
Evaluation:	240
Success Criteria CheckList	240
Additional Features:	242
Maintenance:	243
Limitations:	247
Final Code	248

Analysis:

Problem Identification:

As I returned to school in September I witnessed first hand that nearly 80% of the new external students managed to get lost over 3 times within the first day and only really got familiar with the school layout after 2 weeks of staying at the school. This experience has brought to my attention that newcomers who are unfamiliar with the school layout tend to get lost a significant amount of times before familiarising themselves with the layout after a period of around two week of trial and error when trying to locate areas and classrooms , so I wish to create a solution that will solve the problem of locating and travelling to places within the school.

Features:

To solve this problem the software will be required to:

- 1) Be able to locate the user
- 2) Allow users to input their requested destination
- 3) Be able to guide the user on the most efficient path to the users destinations
- 4) Show an accurate layout of the inside of the school buildings
- 5) Show an accurate layout of the site
- 6) Authenticate the user

To solve this problem the hardware will be required to:

- 1) Be able to display an image
- 2) Be able to perform complex maths calculations
- 3) Be able to store necessary data
- 4) Be lightweight and transportable

Stakeholders:

The target audience will be within the population of the school as the problem itself is specifically stating the school as the area of the problem and the residents of that area will obviously be involved ,but to narrow it more my main target audience will be in the age range of 11-16 as the problem mainly affects students as they make up the majority of the school's

population with a ratio around 1:16 for teachers to students and majority of the time spend a shorter period of time within the school.

To be more specific, there will be 3 main stakeholders:

Stakeholder A being year 12s externals as they will affect the development of the program as they are the priority users as they are the ones that have a severe consequence when it comes to the problem of navigating the school as them being lost could result in them missing time of there a level subjects which would reduce their learning experience which is crucial at their stage of education.

Stakeholder B being year 7s as although the consequences of not having this problem solved isn't as sever as stakeholder A they are still able to affect the program as their experience and request are also useful in the development of the program as they are also affected by the problem of navigation as they are new to school layout and plan on travelling it on a regularly basis.

Stakeholder C being a teacher as although they are not my main target audience as they are most likely not affected directly by this problem they have a role of admin in the solution in order to keep the solution up to date to maintain its efficiency and thus will need to input their ideas when it comes admin based functions.

Reasons For Why It's Solvable By Computational Methods:

My problem is better fitted to be solved by a computer program as the majority of the features of the problem fall under solutions of computational methods such as:

The need for simplification:

You could make the argument that this could be solvable without computational methods as there are multiple examples of tangible maps but I'd disagree as in this scenario there would be loads of variables and locations to highlight so much it would be overwhelming to the user. This is where I believe this feature could be solved by computational methods as through the use of abstraction we can hide unnecessary details to highlight the main details that allow the user to easily navigate the site. To be specific in this instance we will be using data abstraction since we are trying to hide unnecessary variables such as from the user so they focus on the main details, in context being the user's location, destination and the path to the destination. This solution is useful to the stakeholders as it allows them to easily focus on key details when it comes to visualisation and processes such as the path to their destination and what part of the school their destination is at.

From this we can see that although the problem can be solved by different methods, such as physical maps, it is more beneficial to use a computer to produce a solution as it allows the user to focus on their unique route based on their specified destination which non computational methods ,such as physical maps, can't do as they can not change the details they are abstracting while a computer can to give a more detailed path to their destination or reveal more information to the user if needed ,to put this into context a map could abstract the site to show only the routes ,but a computer could abstract further to show only the needed routes based and context based on the users destination and if needed also the interior routes inside the buildings of the site or revealing a more accurate layout to give a better understanding of where the user is going , displaying how a solution produced from a computer would give a more efficient solution than other methods in this scenario.

Breaking down the problem:

Breaking down the problem allows us to see the features and split them up in order of priority and not only does this allows us to efficiently manage the development of the project to solve the problem it also allows us to solve subproblems with 1 subroutine each.This can display why it's beneficial to solve by computational method for this problem as by using problem decomposition not only can we attain the benefits as stated before we can then improve it by having each subroutine that is designated to a problem to have its own display making it simple to develop and understand.

From this we can see that although the problem can be solved by different methods, such as physical maps, it is more beneficial to use a computer to produce a solution as it allows the developer to create a solution that is easier to edit and manage,based on users request and needs. While other methods can edit and manage their solutions it would not be efficient as a solution produced from a computer as they would most likely have to reproduce their entire solution while a programmer can just edit the code based of the section of the code that is not doing it's part effectively or needs to be changed due to user request without having to redo their entire development for the solution.To put this into this context it would split the problem up to 3 subproblems being:

- 1) Displaying the pixelated map
- 2)Allowing the user to input their location and destination
- 3) Finding the shortest route to the destination and displaying it

From these subproblems we can create a subroutine as stated before to individually solve each problem. This would reduce the development and edit time as we have efficiently organised the code allowing us to accurately edit or produce sections of the code without having to redevelop the entire solution ,unlike other methods, displaying why using a computer is better than other methods in this context.

Need for divide and conquer:

As the problem at hand is required to be efficient ,as we want to find the shortest routes from the user's current location to their specified destination, we need to narrow down the data we need. This can be done through a computational problem known as divide and conquer which is a technique that reduces the size of the problem with each successive iteration by taking the problem and applying the rules or specification narrowing down the data to only the data that is required. This technique is beneficial to solving the problem as it can provide us with an algorithm that narrows it down to the best path for the user and this idea can also link to another computational method being a **theoretical approach** as this would enable the program to use a mathematical equation that would help in the process of producing an efficient route.

From this we can see that although the problem can be solved by different methods, such as physical maps , it is more beneficial to use a computer to produce a solution as the solution can be more efficient as it can narrow down the data based on input unlike other methods.This is useful as this reduces the time needed to produce the solution as it would reduce the time to produce a solution as there's less data to search.To put this into context we could use the A* or dijkstra algorithm being the theoretical approach to coordinate the shortest path from the users location to their requested destination and to increase the efficiency we could narrow the data by narrowing the possible paths by restricting it to actual existing paths in the school reducing the time of the calculation.

The Output:

The output from the solution to the problem should produce a 2d pixelated map that displays clearly the shortest and most efficient route from the user's specified location to the specified destination. The most efficient route is useful to the stakeholders being students ,specifically year 7s and 12s, as this would reduce the time of education lost in transport either due to being unfamiliar with the layout or not knowing the most efficient route.The 2d pixelated display is valuable to the stakeholders being students,specifically year 7s and 12s, as this design of the map provides a user friendly experience as it's aesthetics is marketed towards the stakeholders,but this design is also valuable as on a pixelated map the route can be easily highlighted for the user to see.

Interview:

I intend to interview my stakeholders directly affected by the problem being A and B via questionnaires. The intent of the first interview will be to get a statistic of how many people

need the solution to justify the development of the program which is why I've decided to use a questionnaire in this instance as it gives an easy way to see what the majority thinks.

Questions:

- 1) Have you ever been lost on school grounds?
- 2) If applicable, how many times have you gotten lost?
- 3) How many times have you been late?
- 4) If applicable, why have you been late?
- 5) If applicable, why did you get lost?
- 6) What year of education were you late the most on this site?
- 7) What type of map do you think is the easiest to follow:
 - .Satellite
 - .Abstracted(only shows roads)
 - .Pixel based map(eg.pokemon red's map)
- 8) What type of map do you think is the most user friendly/interactive:
 - .Satellite
 - .Abstracted(only shows roads)
 - .Pixel based map(eg.pokemon red's map)
- 9) If applicable, how many weeks did it take you to get used to the school's layout?

Question one, two and three establish if the stakeholders get lost and this factor/data is important as if a large proportion of the stakeholders say yes it shows the severity of the problem giving the justification for the development of the program.

Question four and five establishes why people get lost and this data is important as it dictates the solution to the problem as by looking at the most common answer it gives us an idea of the common problem allowing us to make a common solution for it and also shows if the problem really is needed to be solved by computational solutions based on the common problem derived from the question.

Question six establishes if the stakeholders chosen were correct by identifying the most common year and comparing it to the stakeholders chosen.

Question seven and eight establish the type of map that is best suited for the user based on their preference and the efficiency of their use.

Question nine can establish the need of the solution if it's evident that it takes a long period of time for the user to get used to the schools layout

Answers:

22 students have answered the survey

1) Have you ever been lost on school grounds?

Yes [17]

No [5]

2) If applicable, how many times have you gotten lost?

1[3]

2[5]

3[4]

3+[5]

3) How many times have you been late?

Once[6]

Twice[6]

Three times[5]

3+ [5]

4) If applicable, why have you been late?

6 answered that they were late due to being confused with the pips

16 answered that they were late due to being lost

5) If applicable, why did you get lost?

14 stated that it was due to not being familiar or knowing the schools layout

2 answered that it was due to misdirections

6) What year of education were you late the most this site?

7 [12]

8 [0]

9 [0]

10 [0]

11 [0]

12 [10]

13 [0]

7) What type of map do you think is the easiest to follow:

1) Satellite

2) Abstracted(only shows roads)

3) Pixel based map(eg.pokemon red's map)

1 [3]

2 [11]

3 [8]

8) What type of map do you think is the most user friendly/interactive:

1) Satellite

2) Abstracted(only shows roads)

3) Pixel based map(eg.pokemon red's map)

- 1 [2]
- 2 [5]
- 3 [15]

9) If applicable, how many weeks did it take you to get used to the school's layout?

- 1[1]
- 2[4]
- 3[12]

Analysis Of Answers:

From this survey I have established the justification for the solution as the survey has made it evident that the problem is affecting a majority of people on the site.

Question 1 2 and 3 provides valuable data as it's from the highest population on the site being the students and from this data we can see the majority have gotten lost multiple times establishing the need and justification for the solution

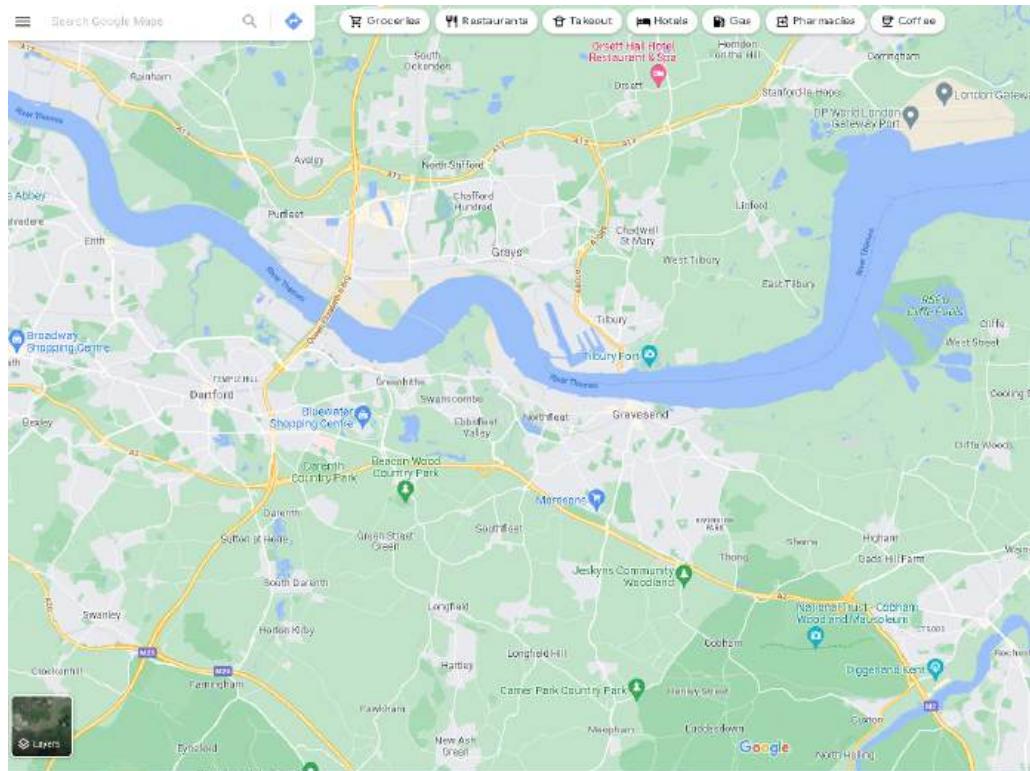
Questions 4, 5 and 6 provide valuable data as it displays the common core of the problem being the fact that it was due to lack of familiarity with the site due to being new and is useful as it proves the problem identification to be correct also providing justification for the solution. This is established from the fact both year 7 and 12 are the majority of people being lost , which we can tell from question 6, with both causes being stated to be due to lack of familiarity.

Question 7 provides valuable data as it establishes the most preferred display of the solution by asking the highest population affected being an abstract map while question 8 displays the map that the target audience finds most enticing.This is important as user retention is important in this scenario as the aim of the solution is to familiarise themselves with the layout so they need to memorise the map which is only possible if they're constantly drawn to the map which can be done through appeal.

Question 9 provides valuable information as it can be used as justification for the solution as for some problems although there's a problem computational methods aren't always necessarily needed but from question 9 it's clearly shown that computational methods are needed as it's taking an extended period of time for the stakeholders to solve the problem which is negative in this case as this affects their education for an extended period of time.

Researching existing similar problems:

Product 1:



Google maps is a web mapping website and app coded in javascript that allows the user to perform a variety of map based features such as: finding the route to a user suggested location based on their mode of transport , saving locations , adding stops on your route and predicting and changing the route based on interrupts such as traffic just to name a few features. The features are possible mainly due to the interaction between machine learning and the data provided , such as location, historical traffic patterns, local government data, and real-time feedback from users , to form the shortest route based on the user's need.

Possible adaptations:

My solution could implement the use of feedback from users to make the solution “**be able to show an accurate layout of the inside of the school buildings**” by making it known to the user that certain destinations are unavailable.

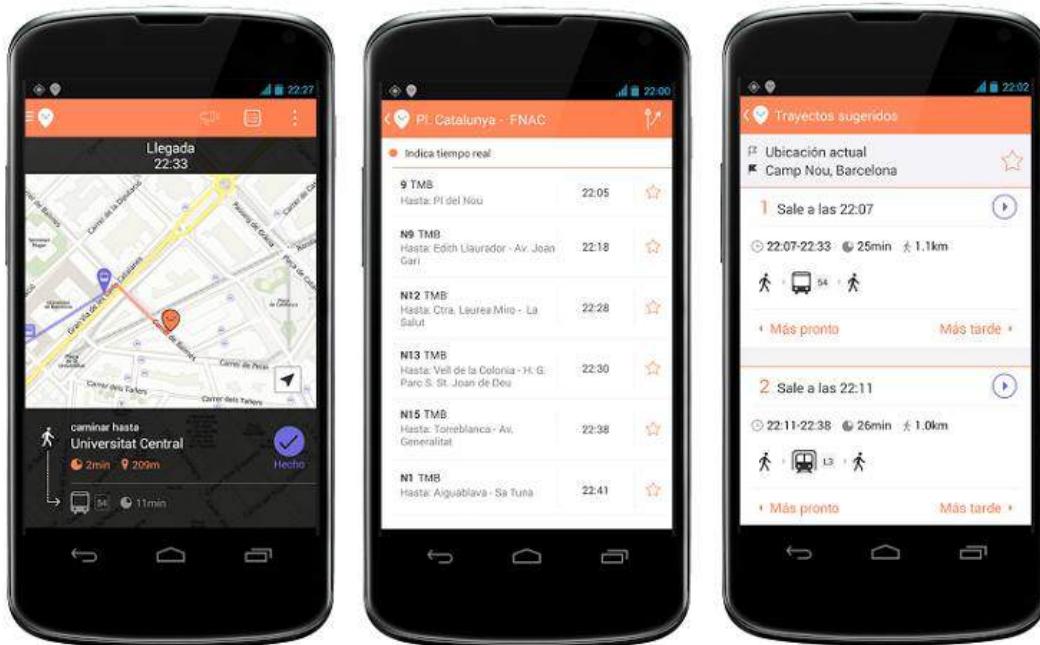
.my solution could also implement machine learning to predict the routes unavailable based

on the time for example the during lunch the program can adjust routes based to encoute for lunch rushes interference which is useful as stated before due to the fact my program software must have the feature of **“being able to guide the user on the most efficient path to the users destinations”** and this enables the path to be more efficient by taking interrupts into account.

.My solution could use the addition of the layer feature for different variations of the map which can be seen as useful as my program must be able to **“show an accurate layout”** which can me accomplished more efficiently by having multiple layouts

.My solution could use the side menu to keep all features neatly organised so that the display isn't overly crowded with features which would enable the program to **“show an accurate layout of the site”** as stated in my software features.

Product 2:

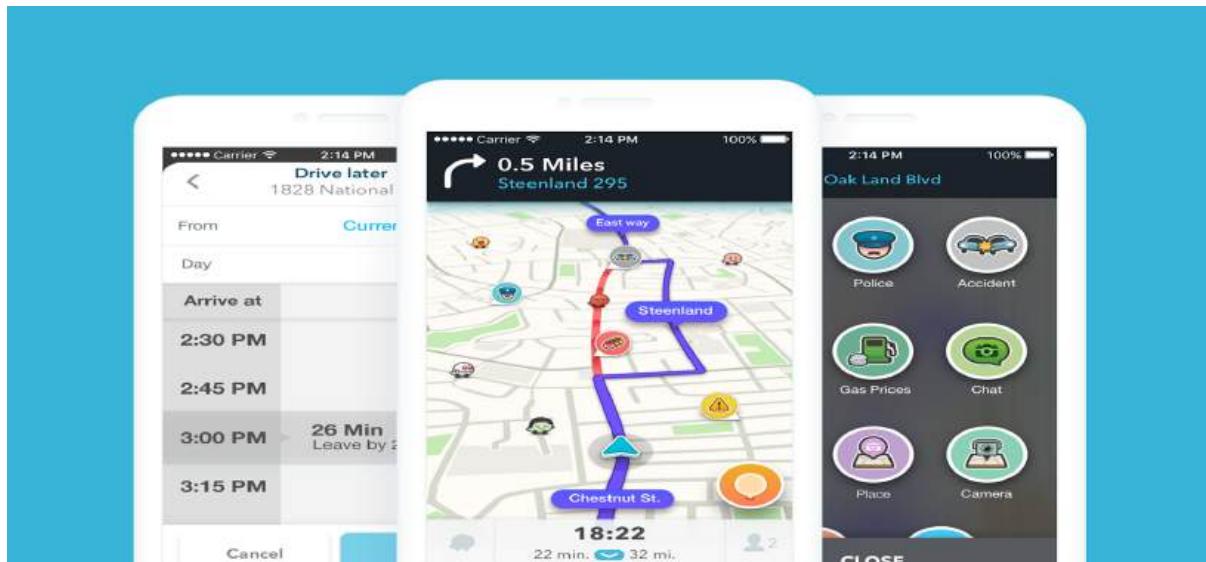


Moovit is a mobile web mapping app similar to google maps but differs in the fact that it's routes are centralised around public transport and unlike google maps which creates routes through the interaction between machine learning and the data give uses pre created routes and combine them to get the user to their destination based on the time of their departure or desired arrival.

Possible adaptations:

I could implement the estimated time feature of the app which can give you the approximate time needed to get to the location based on your desired arrival time or the approximate time of the journey time which can be helpful in this context as it can give a student a proper sense on when to leave to go to class fitting in the program feature of “**being able to guide the user on the most efficient path to the users destinations**”.

Product 3:

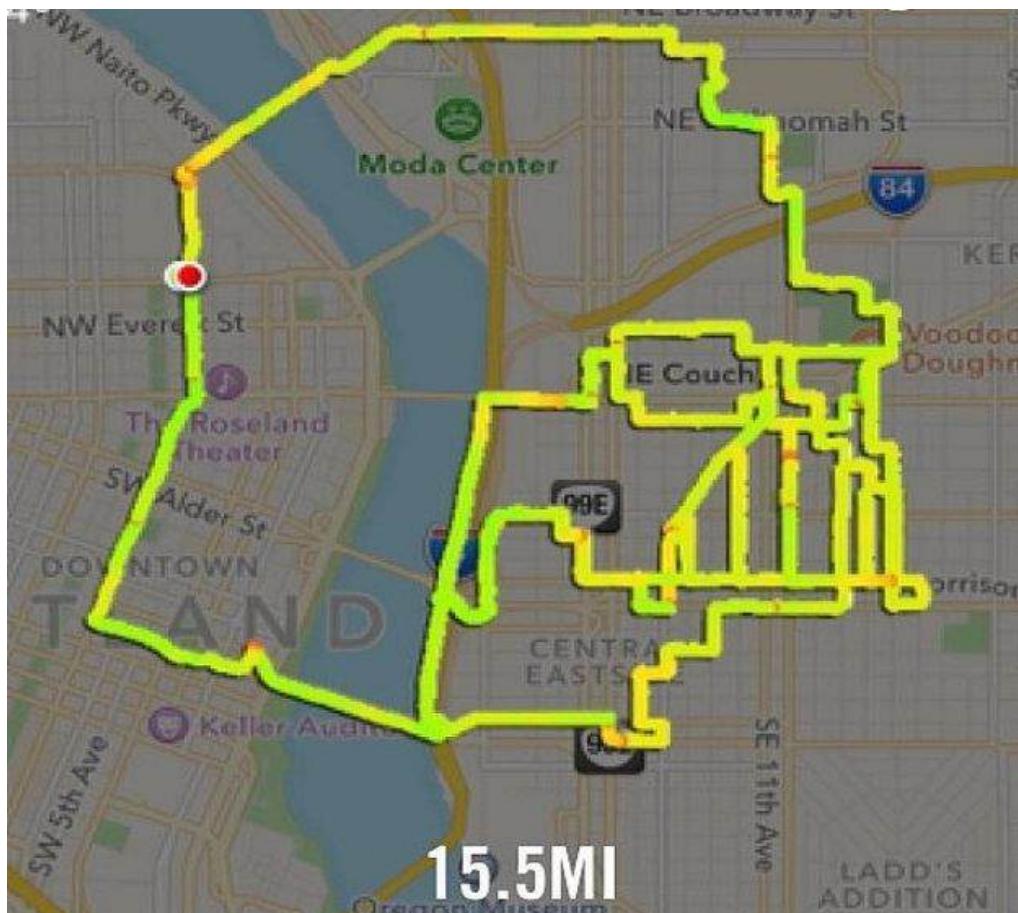


Waze is another web mapping app but is centralised around car routes using a similar method to google being the interaction between machine learning and the data provided , although waze utilises real-time feedback from users much more than google by allowing users to identify specific things such as police , gas stations and landmarks through the use of their pin features.

Possible adaptations:

My solution could implement the feature of pins to increase real-time feedback from users to “**show an accurate layout of the site** ” and make the solution “**able to guide the user on the most efficient path to the users destinations**” by having more data to reduce the time of the route and also take account for any more interferences that might not have been predicted by the system

Product 4:

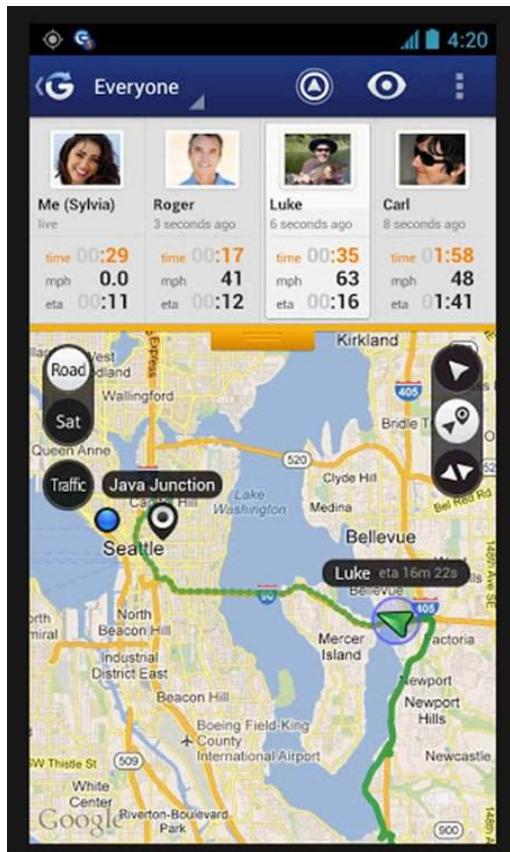


Nike+ Running is an app that uses a map centralised around pedestrian routes to allow the user to create their own personal drawn map for a personalised route. This app has a feature to track the speed of the user and notify them when they're going slow and can keep track of their average speed.

Possible adaptations:

.My solution could implement the speed tracking feature to notify the user when they're walking too slow to class and will be late or use it to give them an approximate time to leave for break or lunch so that they are not late to their lessons which would fulfil the hardware feature to “**be able to perform complex maths calculations**”.

Product 5:



Glympse is another web mapping app that allows the user to create routes based on their desired destination although it differs from other apps as it has the ability to reveal the users location to other specified users allowing them to easily track them down and meet up with them.

Possible adaptations:

.My solution could implement the use of displaying locations of teachers ,as this would make the users destination more specific allowing the solution to “**guide the user on the most efficient path to the users destinations**” and show “**an accurate layout of the sight**”

. To put this into context if a teacher was in their classroom the room could have a green light to represent that and red when they're not with a note of where they are.

Initial Concept Of Solution:

My program will be an application run on a mobile device as it has been shown that this is the most common and accessible way of using an online map through the product analysis with all products doing so or at least having a mobile variation.

When the application is started it will display a birds eye view of the site as shown in most applications researched so that it's clear to the user exactly where they are giving an "**accurate layout of the site**" so that they can make get more familiar with the site itself and make a more informative decision on their destination.

On the bottom of the screen there will be a small list of user set pins

On the top left hand corner there will be 3 lines to represent a menu where the user can access more features such as:

- .average speed

- .Pins which will allow the user to customise their map by adding a pin on the map reminding the user of their important locations

- .Log in function in order for the user to be able to store and access specific pins and other functions

- .Status function if the user that logs in has high enough authorization in order to change the status of the room

On the top right hand there'll be a magnifying glass to represent a search bar where the user enters their location and desired destination in order to create an 'efficient path' with the added inbuilt functions of showing a teachers location ,if they allow it, to help the user find them incase they're not at their designated room and another inbuilt function being notes which can be displayed to the user when they click the down arrow to see any information about the room that the owner/teacher has set such as if it's locked or who the classroom is currently assigned to.

Limitations of my solution:

The solution's main problem is that it's accuracy of the route as it can't rely on real time feedback ,as this would require the program to utilise a network in order to pass updates on information about routes , which I currently do not know how to do nor have the time to learn and implement into the solution due to time constraints, and thus there is no user feedback and thus the quality of the solution is reduced as the route can not account for any possible disturbances.

Another limitation may be the lack of use of gps as this reduces the program's ability to locate the user accurately and create a path as I am coding this with a limited budget and

time so being able to find and convert a gps that would be able to be translate onto a custom made map would not be possible due to these restrictions and thus the solution shall not be as accurate as other graphical solutions that make use of a gps.

Interview 2:

The intent of this interview will be to get an idea of the features that people believe that are more beneficial when improving their journey when travelling across the site ,with consideration to limitations and features when asking questions.Unlike the first interview we will need reasons rather than a range of opinions to properly justify each added feature so this interview will be conducted by contacting individual stakeholders , being Alvin Roy , Joseph Baker and Mr Cooker , to get their suggestions and critiques when it comes to the initial concept of the solution so I can improve it based on their opinions

Email Sent:

"To Whom This May Concern,

After more consideration on what the product goal is I have gotten a range of specific features that may be involved in the solution and I would appreciate your input on which features get developed and any suggestions you have for features.

When the application loads the user will have a birds eye view of the site with a line of bubbles with locations near the bottom of the screen storing the users pins , which is a feature that stores user requested locations. This feature is intended to quicken the process by removing the process of searching for the user's location and destination.

The startup screen will also include 3 lines in a box at the top left to represent the menu feature which will include a range of features such as:

- average speed
- Log in function in order for the user to be able to store and access specific pins and other functions
 - Pin function which allows the users to save locations that will appear on the main display for easier use to increase efficiency
 - Status function if the user that logs in has high enough authorization in order to change the status of the room

The startup screen also includes a magnifying glass in a box located in the top right corner to show the search function which allows users to locates specific rooms with the added inbuilt functions of showing a teachers location ,if they allow it, to help the user find them

incase they're not at their designated room and another inbuilt function being notes which can be displayed to the user when they click the down arrow to see any information about the room that the owner/teacher has set such as if it's locked or who the classroom is currently assigned to.

Thanks

Yours Faithfully , Tireni Oluwabunmi"

Replies:

Alvin Roy(Yr12):

"This sounds promising for an initial concept as it efficiently does it's main purpose , although I would say that the note feature is a bit unnecessary as in most cases there's not that much extra detail needed to know about a room/area , with only real piece of information needed to know being if it's locked."

Joseph Baker(Yr7):

"I'm satisfied with all the features stated as they seem easy to identify and use , although I think that displaying the users average speed might be unnecessary and a waste of space and can be replaced by a more useful function."

Mr Cooker(Teacher):

"I'm pleased with this concept but in regards to space I do worry that this application may take to much of my memory as I need a significant portion for my school documents"

Analysis Of Answers:

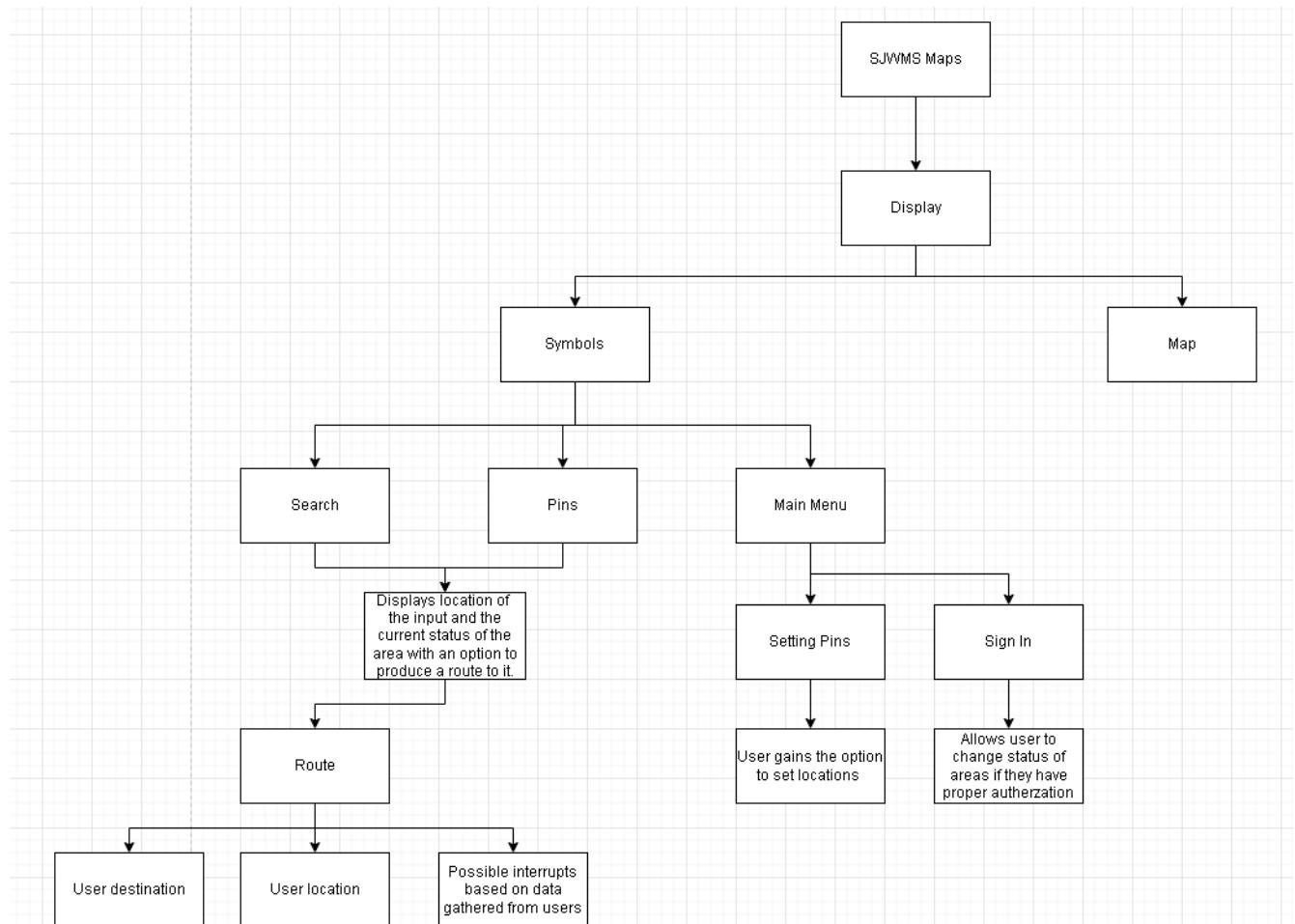
From these responses it's made evident that the stakeholders are pleased with the majority of features suggested with a few that need to be improved or removed such as:

-The addition of a status feature ,that indicates whether a room is available based on the highlighted colour on the rooms description, in exchange for the removal of the note feature as it completes the note features task more efficiently and is easier to implement.

-The removal of the display for average speed due to the users finding it unnecessary to display and might cause an unnecessary amount of background memory being used making the app undesirable for some users who require a lot of memory for other purposes.

Giving me a clearer idea of what I can do to improve the solution by getting a better idea of how to improve features by finding out user limitations and preferences.

Abstract Diagram:



Analysis Of Abstraction Diagram:

The abstraction diagram is a visualisation of the solution through the use of divide and conquer by splitting the solution into smaller sub categories to give a clearer idea on how the system works , with it being used in this scenario to display the systems and data used.From this diagram I was able to realise that it will be necessary for me to use text files in order to store the data necessary for the solution to function.

Stakeholder Requirements:

Design:

Requirement	Explanation	Limitation
Contain all rooms on the site	This is necessary as stated in the problem identification the main goal of this application is to make the user more familiar with the site as it's evident that newcomers are unfamiliar with it shown best through the survey conducted in the 1st interview with the majority of the people who answered that they have been late was stated to be caused by the fact that they were unfamiliar with the site during the time and by having the entire site within the application it ensures the possibility of the user becoming familiar with the site.	It won't be able to contain all rooms in high quality detail due to me having a limited budget and time and it was also made clear to me that the majority of users won't be able to handle high quality due to their limited storage as made evident to me in the 2nd client interview .
Be able to show the status of all rooms	This is to aid the user when it comes to decision making when travelling as from my product analysis I had Identified that it would be useful if the program could show the status of a room in order to further complete the software feature of "showing an accurate layout of the sight" which would allow the user to	Can't be overly specific as this would overflow the user's display not allowing the program to clearly display the layout of the school making it unable to complete the software feature of "showing an accurate layout of the sight" and also the fact that some users wouldn't have the storage required as

	make an informative decision on if they should use the path to go to their inputted destination.	made evident to me in the 2nd client interview .
Clearly display and highlight a route to the user inputted destination from the users current location	This is to make it clear to the user what route to follow which is useful as displayed in the research it used commonly to make excessively clear to the user the route to follow allowing the program to “be able to guide the user on the most efficient path” which my clients agreed upon within the 2nd client interview .	Route can't be abstracted too much or it will remove surrounding areas that can be used for the user to familiarise themselves with the site reducing the efficiency of the solution as it's not following the software feature of “displaying an accurate layout of the site” not allowing the program to accomplish its goal being for the user to familiarise themselves with the site
Be able to display user saved locations on the starting page	From my product analysis I was able to see that this function would allow the user to quickly access their common locations making the solution more efficient by reducing the time it takes to produce a route accomplishing the software feature of “Being able to guide the user on the most efficient path to the users destinations” .	Limited number of pins as if there's too many user's display may be overflowed and it will remove the purpose of the function and disable the potential of the solution to have the software feature of “Show an accurate layout of the site” and “Be able to guide the user on the most efficient path to the users destinations” , while also having the risk of taking too much of the user's memory as some users wouldn't have the storage required as made evident to me in the 2nd client interview .

Functionality:

Requirement	Explanation	Limitation
Not take over 1GB of memory	As the primary target audience is students they	The solution will still need to take some background

	<p>are most likely in need of storage for school so the solution taking a lot of memory would be unattractive to them , this view being backed up through the 2nd Client Interview .</p>	<p>memory to function properly with the reasons so being best displayed by its hardware features.</p>
Be usable on mobile devices	<p>Primary target audience is students and the most common device would be mobile phones and mobile phones would be more ideal than most other devices as it's transportable ,as shown through the mobile applications in my product analysis, which is required in this scenario as stated in the hardware features the device must be "light and transportable"</p>	<p>Outdated mobile devices may not be able to run the app with the program using up to date software, which might limit the range of students that are able to use the app as they won't all have the same level of software.</p>
Be able to locate user inputted destination	<p>Solution requires a destination to calculate the most efficient route to accomplish the software feature of "being able to guide the user on the most efficient path".</p>	<p>As I am a student I would be unable to integrate something as complex as a gps within my solution like the applications in my product analysis have done.</p>
Be able to locate user inputted starting location	<p>Solution requires a starting point to calculate the most efficient route to accomplish the software feature of being able to guide the user and also accomplishing the hardware feature of using gps to locate the user.</p>	<p>As I am a student I would be unable to integrate something as complex as a gps within my solution like the applications in my product analysis have done.</p>
Be able to change the status of all rooms if user has necessary authorization	<p>Status of rooms may change and the user needs to be able to make an informed decision on their destination to increase the efficiency of the solution ,accomplishing the required feature of guiding them on "the most efficient path" which is displayed through its use within similar programs within my product analysis.</p>	<p>Can't be overly specific as this would overflow the user's display not allowing the program to clearly display the layout of the school not completing the software feature of "Show an accurate layout of the site" and also run the risk of taking too much of the user's storage which is not ideal as some users might not have the required space</p>

		as stated within my 2nd client interview .
Be able to save users requested location	This increases the efficiency of the solution by reducing the time it takes to produce a route by predicting the route a user may use, as displayed by similar programs within my product analysis . This would allow the solution to accomplish the software feature of “Being able to guide the user on the most efficient path to the users destinations”	Limited number of saved locations due to storage and overflowing the user as the app is designed to be run on a mobile device owned by a student as stated within my 2nd client interview .
Be able to suggest necessary corrections to input	This increases the solutions efficiency to accomplish its goal as it suggest necessary corrections for the user to make in order to use it the application in it's ideal manner	Can not predict every possible outcome and thus the corrections given will have to be broad.
Be able to log in to an account	This allows the pins saved by a user to be stored, increasing the efficiency of the solution by allowing the solution to accomplish the feature of “authenticating the user” and thus be able to utilise other functions based on the user's data like the programs within the product analysis .	As I am a student on a limited budget and time the accounts will be stored in the device rather than a network and thus the users would have to make an account per device
Be able to log out of an account	This allows pins saved by a different user to be accessed allowing the solution to accomplish the feature of “authenticating the user” by making authorization necessary by allowing the user to log back out into a non account.	N/A
Be able to create and store new accounts with authorization	This allows the pins saved by a new user to be stored allowing the solution to accomplish the hardware feature of “storing	As I am a student on a limited budget and time the authorization used will not be overly complex and extremely secure.

	<p>necessary data" as it allows the solution to create and store accounts which may hold the user's pins which increase the efficiency of the solution as shown through the programs within the product analysis ,while also enabling the solution to be able to accomplish the software feature of "Being able to guide the user on the most efficient path to the users destinations". It also allows the solution to accomplish the feature of "authenticating the user" by allowing there to be a user.</p>	
Be able to create and store new accounts	This allows the pins saved by a new user to be stored allowing the solution to accomplish the hardware feature of " storing necessary data " as it allows the solution to create and store accounts which may hold the user's pins which increase the efficiency of the solution ,as shown in the programs form the product analysis , and enables the solution to be able to accomplish the software feature of " Being able to guide the user on the most efficient path to the users destinations "	As I am a student on a limited budget and time the accounts will be stored in the device rather than a network and thus the users would have to make an account per device

Hardware And Software :

Requirement	Explanation
Android 6.0+ OS or iOS 11 OS	In order to run the app the user's OS must be able to provide a suitable amount of storage for it to run and the processor must be able to run the app within a good

	enough time range
Python Interpreter	The code will be written in python so an interpreter is necessary to run the code.
Touchscreen	Touchscreen is necessary as the user must be able to interact with the application so that it can make appropriate decisions when creating a route

Success Criteria:

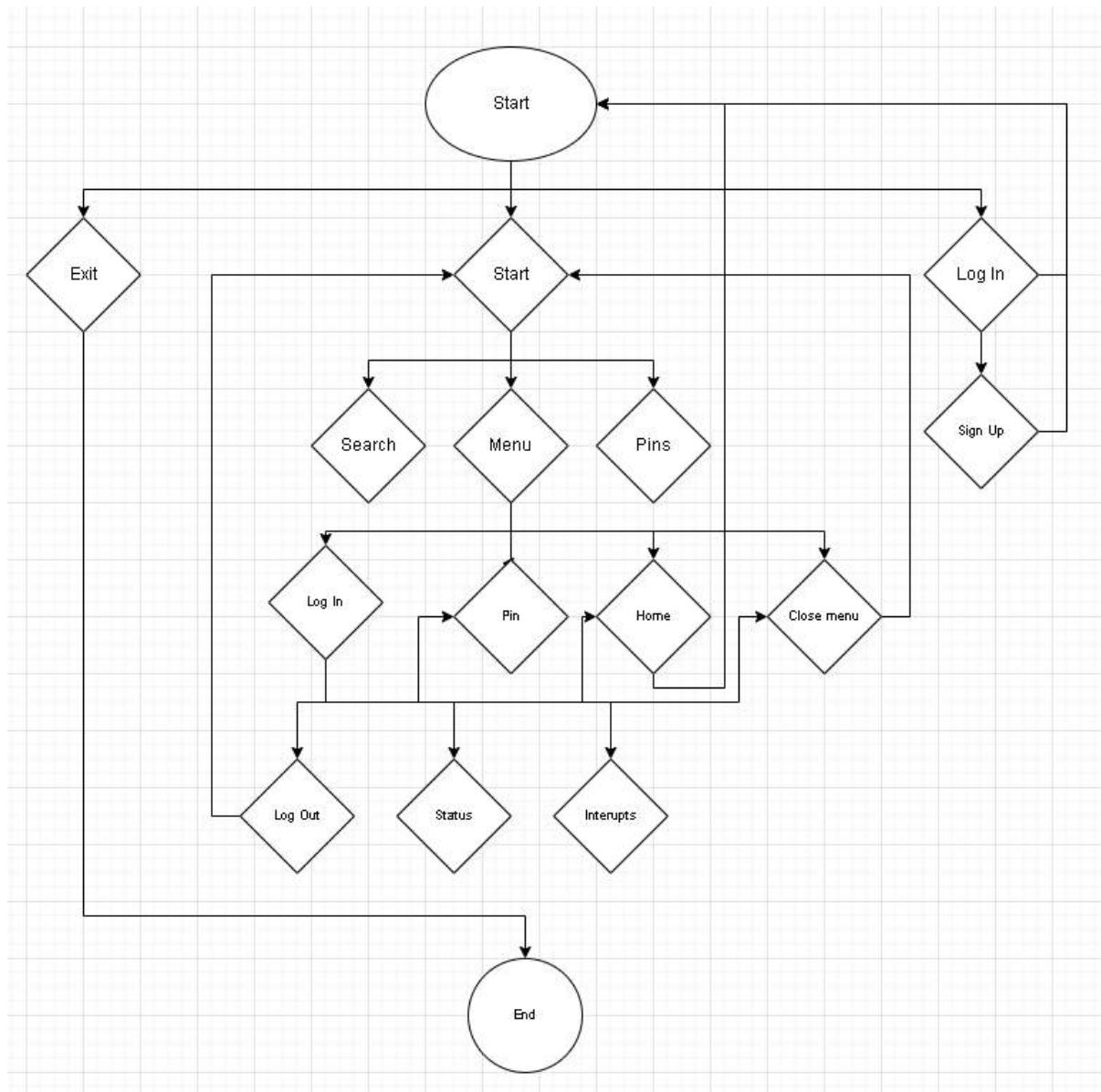
Number	Criteria	Proof
SC1	Be able to display a starting screen	Screenshot of a starting screen containing a start , exit and log in button.
SC2	Be able to exit the application	Screenshot before and after of pressing exit button
SC3	Be able to switch screens	Screenshot before and after of pressing a button that switches a screen
SC4	Be able to log in to an account	Screenshot before and after of logging into an account
SC5	Be able to log out of an account	Screenshot before and after of logging out of an account
SC6	Be able to create and store new accounts with authorization	Screenshot before and after of text file containing accounts after creating an account with authorization
SC7	Be able to create and store new accounts	Screenshot before and after of text file containing accounts after creating an account
SC8	Be able to locate user inputted destination	Screenshot destination being shown on the map
SC9	Be able to locate user inputted starting point	Screenshot starting point being shown on the map
SC10	Be able to clearly display and highlight a route from the user inputted starting point to	Screenshot created route

	the user inputted destination.	
SC11	Contain all rooms on the site	Screenshot site and screenshot map
SC12	Be able to show status of a room	Screenshot status of searched room
SC13	Be able to change status of a room if user has necessary authorization	Screenshot status of searched room before and after room's status is changed after signing in to an authorised account
SC14	Be able to change and store user saved locations	Screenshot pins before and after changing saved locations
SC15	Be able to display user saved locations	Screenshot pins on main screen
SC16	Be able to suggest necessary corrections to input	Screenshot error screen after making an error
SC17	Take under 5 MB of storage	Screenshot amount of storage taken by application
SC18	Be able to use a mobile device	Take a photo of application being used on mobile device

Design:

Brief Overview:

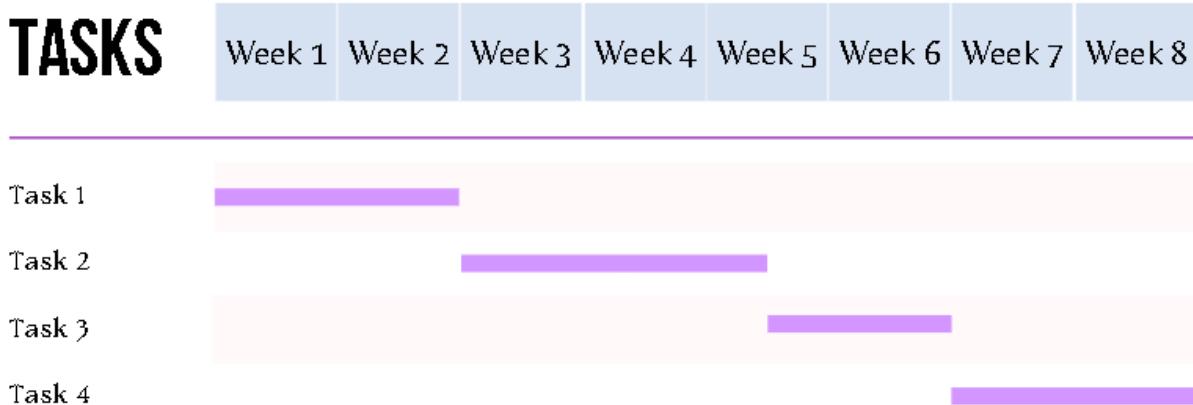
My problem, as stated in the problem identification , is" that newcomers who are unfamiliar with the school layout tend to get lost" and to solve this I have decided to design a mobile program that allows the user to be guided by a map of the site with the system being displayed below:



Sub Program	Task	Success Criteria	Description
Start	Task 1	SC1 ,SC3	This sub program's purpose is to give the user access to the main window via pressing its displayed button.
Log In	Task 1	SC3 , SC4 ,SC5,SC16	This sub-program's purpose is to provide access to other sub-programs if the user's access is high enough. To do so the program will be required to take in 2 inputs from the

			user a username and password the program will then check the username and password to see if both are valid if both are valid by comparing it to the stored username and passwords in a text file. It will then give access to more sub programs based on the username entered as different user will have different levels of access
Sign Up	Task 1	SC3,SC6 ,SC7,SC16	This sup program purpose is to allow the user to be able to use the log in subprogram. To do so the user will need to enter an appropriate username and password ,that's compared to a set of parameters to make sure they're appropriate, and the appropriate access key. With the access key being pre-set giving them users different levels of access. If all are acceptable a new account is created that can be used with the log in sub program by storing the new account into a text file containing other accounts.
Exit	Task 1	SC2	This sub program allows the user to exit the app completely via pressing it's button
Search	Task 2	SC8 , SC9	This sub program is designed to allow the user to search for the rooms on the site in order for them to be capable of inputting a start location and destination for the route creation function to utilise. It'll do this by producing a list of similar values compared to the inputted value with each value in the list also displaying their status increasing the chance of the user getting their desired room in the instance they inputted the room incorrectly and also increasing their ability to make an informed decision based on the status of the

			room.
Menu	Task 3	SC1, SC2, SC3	This sub-program, if clicked, opens a rectangular window on the side of the app allowing the user to access more subprograms.
Close Menu	Task 3	SC1, SC2	This sub-program if clicked closes the menu returning to the main window.
Pins	Task 3	SC14,SC15	This sub program's designed to increase the efficiency of the route creation sub program by reducing the time it takes to create a route as these sub programs display saved routes essentially having the route already planned significantly reducing the time to create a route.
Pin	Task 3	SC14,SC15	This sub program is designed to allow the user to customise the pins acces by changing the path the pin saves.
Status	Task 4	SC13	This sub programs is one gained through authorization and its purpose is to allow the user to make an informative decision on if they want to go to their destination based on the status for the room getting rid of pointless journeys.
Route Creation	Task 4	SC8,SC9,SC10,SC11	This sub program will be the function used to create an efficient path for the user. To do so it will need 2 inputs minimum being destination and start location , which can be inputted in through using the search sub program. From these inputs we can create a path from each point while considering the site's layout through the use of an a* path algorithm.



Possible Inputs:

Outputs

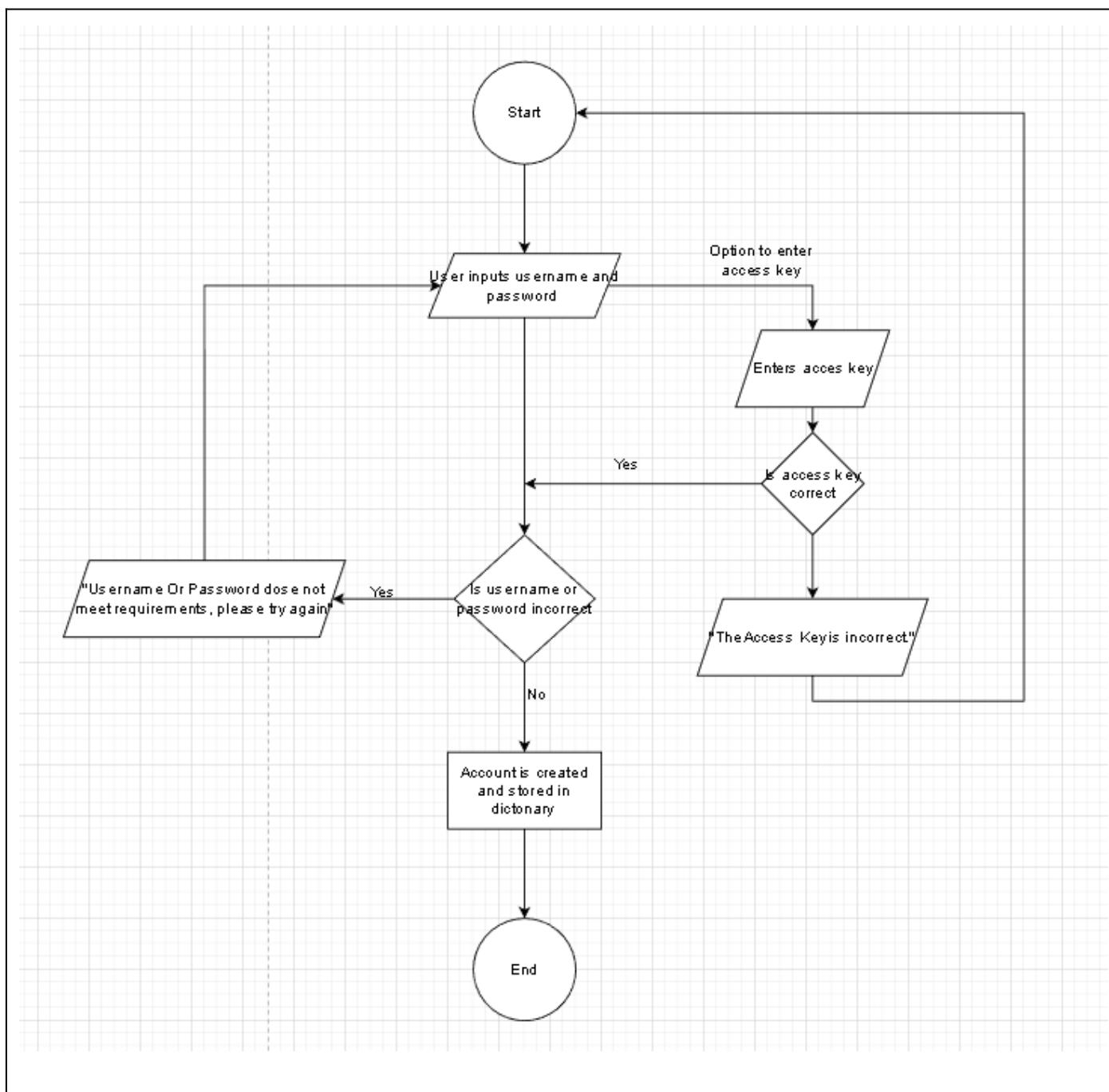
Stored Data

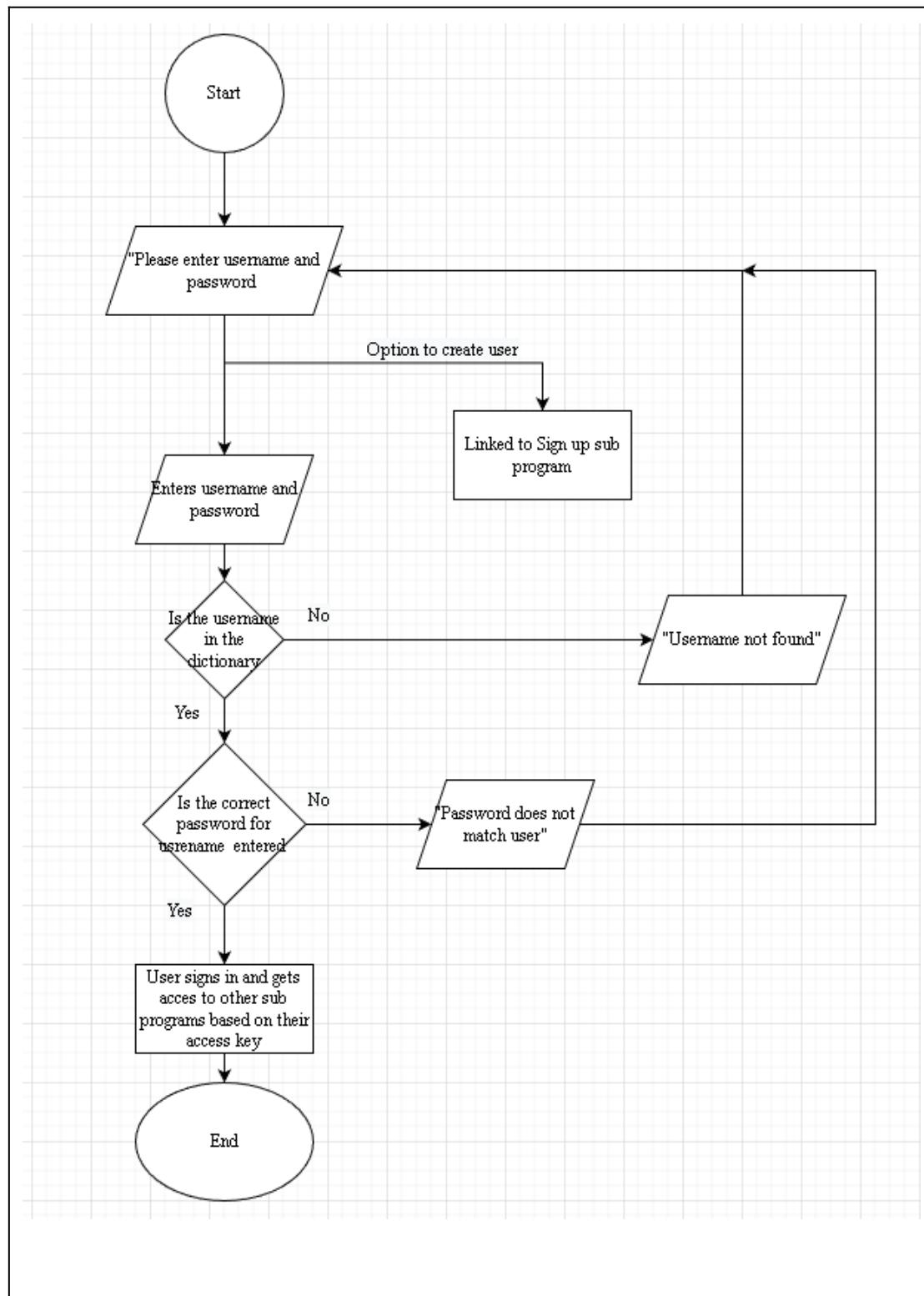
Inputs	Process
Destination	Generates route to destination using A* algorithm or Djikstar's algorithm.
Pins	Stores requested locations and in the future is used to generate route to pinned location using A* algorithm or Djikstar's algorithm.
Username	Stores entered username and gives access to more sub programs based on the account.
Password	Stores entered password to username and give access to the account linked to the username.
Access Key	Entered access key is compared to pre set access keys giving the created account access to more sub programs based on the access key entered.
Room Status	Authorised users change the status of the room with their being 3 status open , teaching and closed these are then outputs

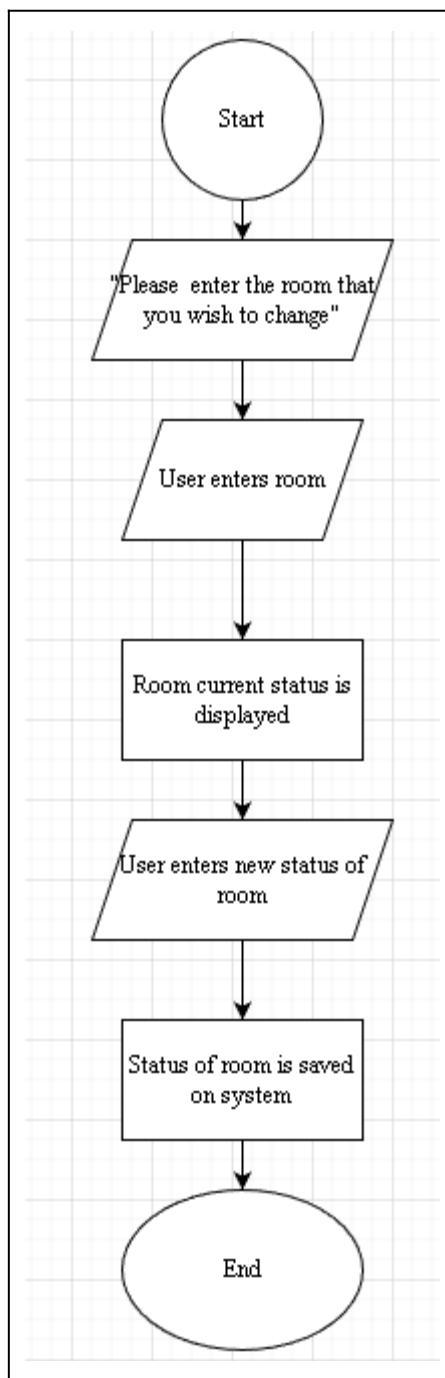
	and stored on the system in the form of stop lights that appear on the card of the room once searched up with green representing open , yellow representing teaching and red representing closed.
--	---

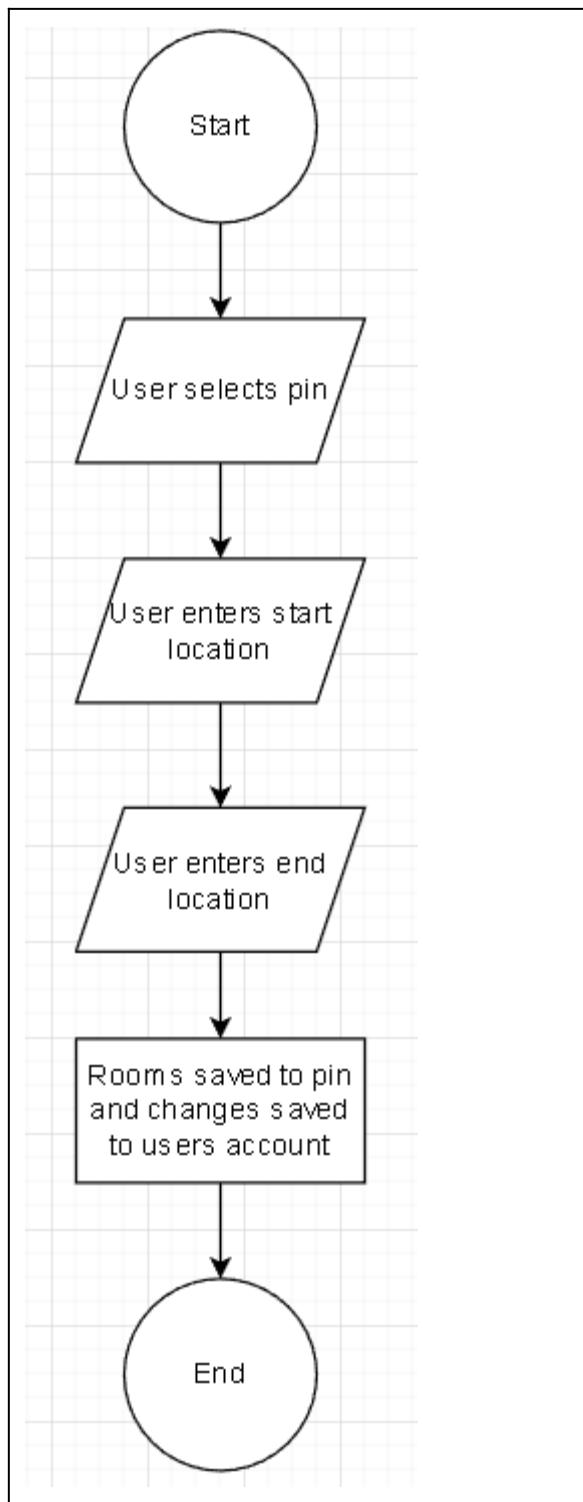
Sub Program Algorithms:

Sign Up:

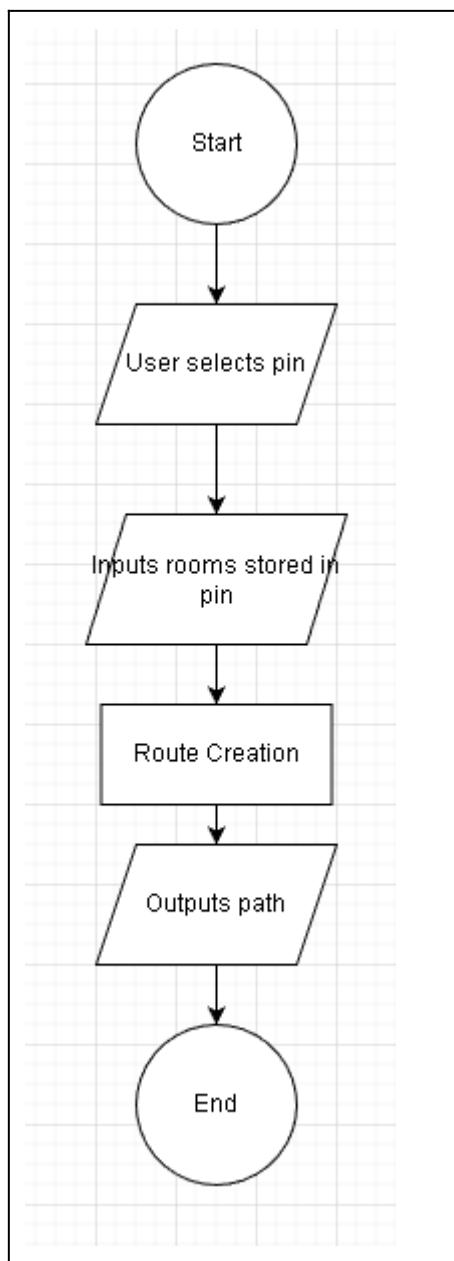


Log In:

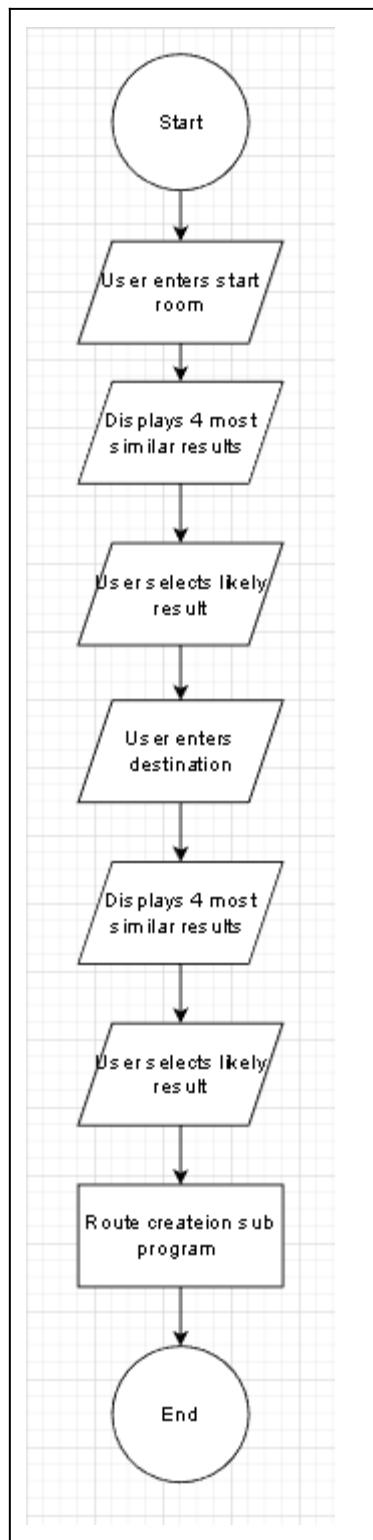
Room Status:**Pin:**



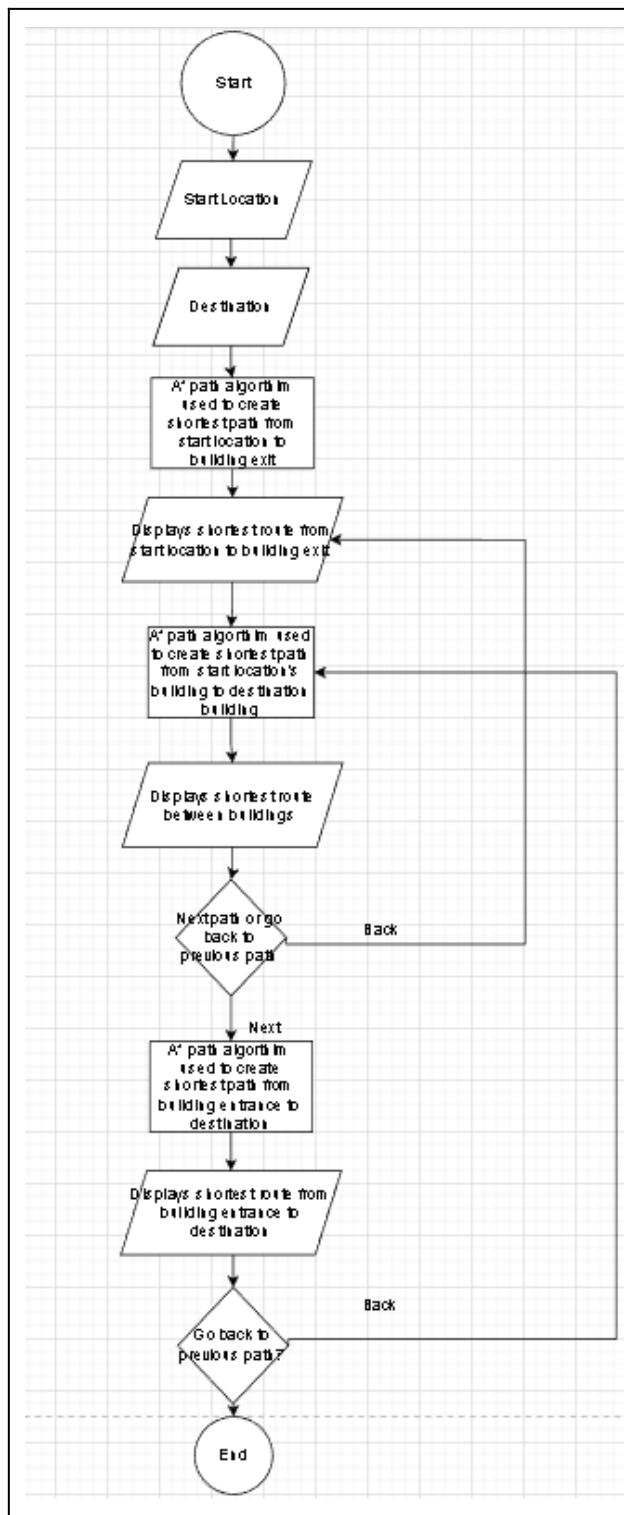
Pins:



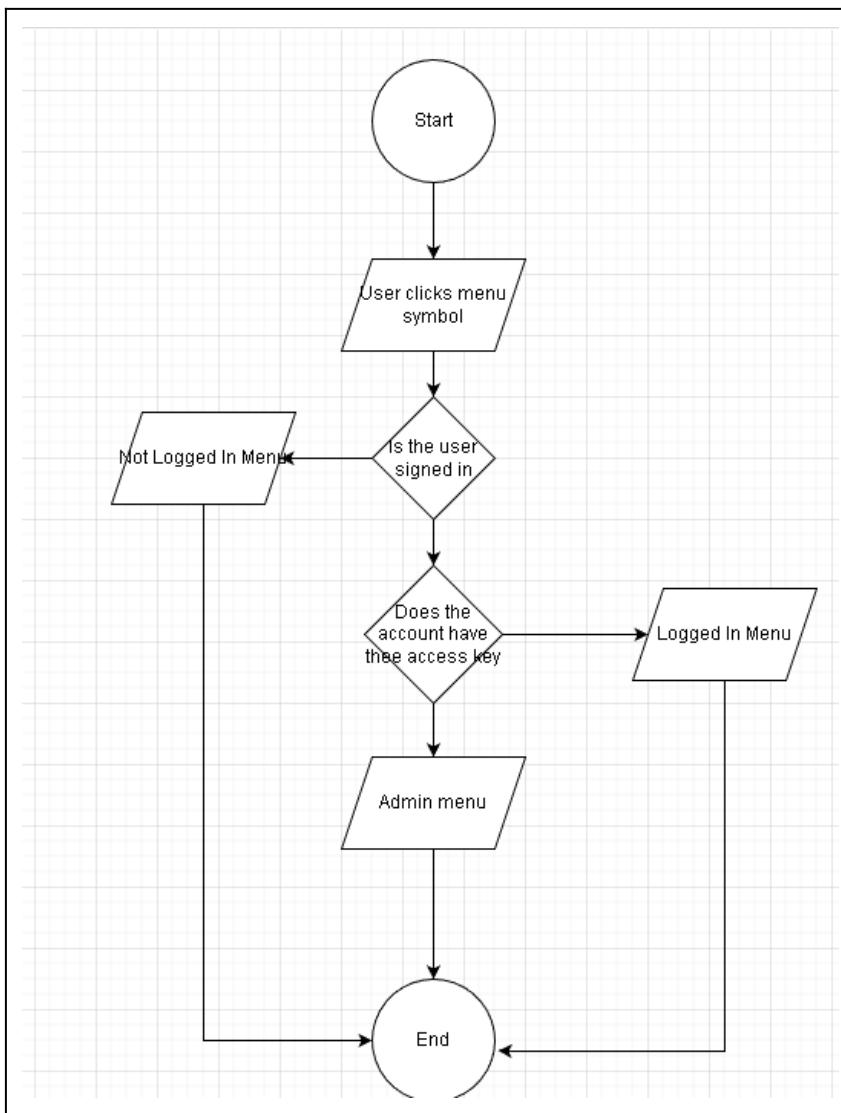
Search:



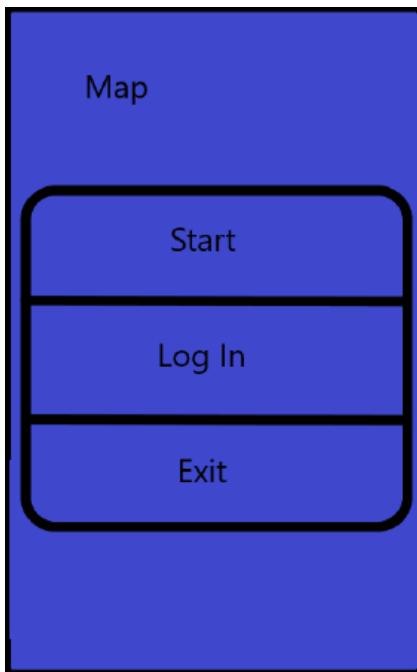
Route Creation:



Menu:



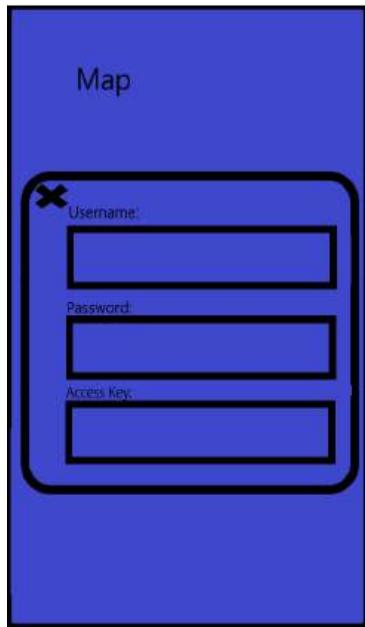
Screen Designs:



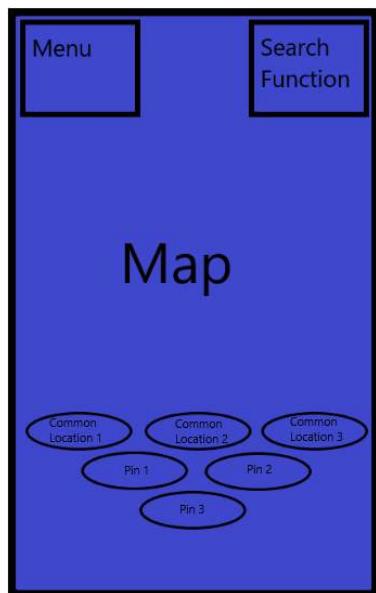
Upon loading into the application this shall be the first screen presented to the user. I decided for the background to be the map of the site to clearly indicate to the user the purpose of the application of familiarising themselves with the site. The central item of interest is a window displaying 3 functions Start , Log In and Exit in order to comply with **SC1**.



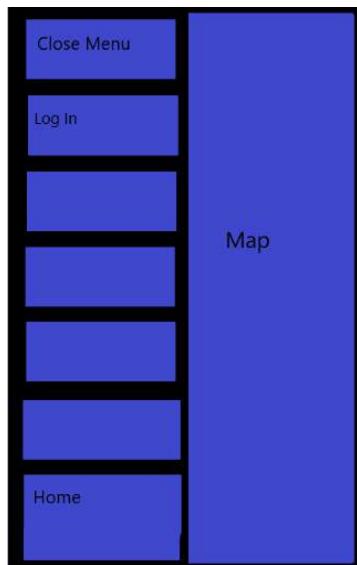
If the user clicks log in the window shall be overlapped by another window containing the necessary input areas for the user to enter their login details so that **SC4** can be achieved. In the event that a user doesn't have a log in I've added a clearly identified area that allows them to create an account being the "CREATE ACCOUNT " button clearly displaying its purpose.



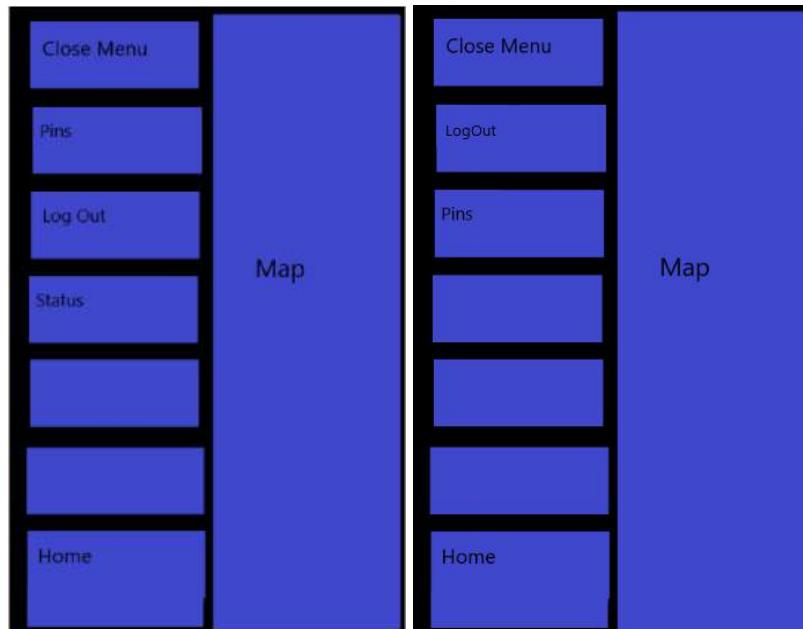
If the user presses the create an account button this window will now overlap the previous allowing them to input the necessary details to create an account so that **SC6** and **SC7** can be achieved.



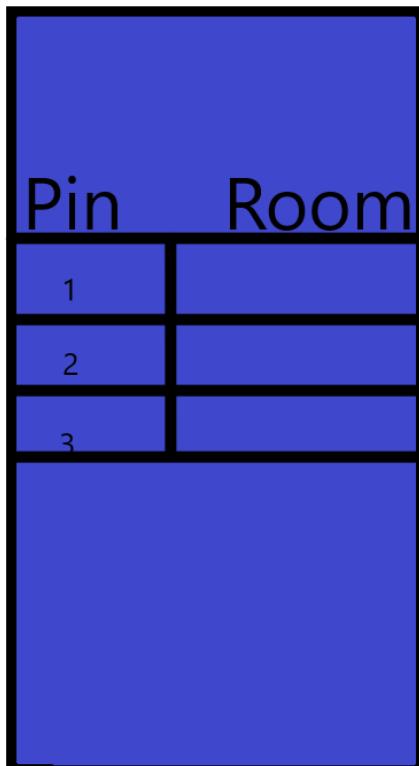
If the user were to click start on the main menu they would be transported to this screen having the map, which is not blocked by buttons so that they can get a true appreciation of the site in order to complete the program's main goal of familiarising the user with the site, and 3 functions displayed in front of them being the: search function , pins and menu function as these 3 functions are the main functions of the code as they link to multiple other functions.



If they were to press the menu button a window would slide across the left of the screen revealing more functions. If they aren't signed in they will only be presented to 2 functions being the home function and log in function due to there being different levels of authorization in order to comply with **SC17**.



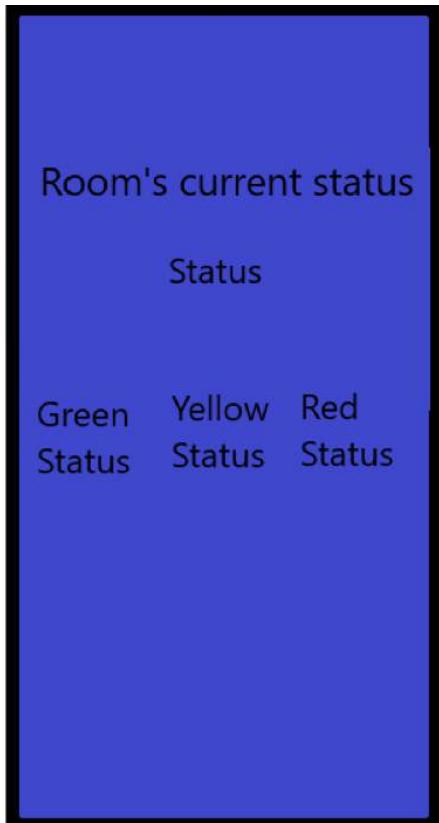
If they are logged in they gain access to the functions logout and pins, but if the account they're logged into has the correct access key the menu that is displayed is the one below giving them an extra function ,being status, due to there being different levels of authorization in order to comply with **SC17**.



If they were to then press the pins button a new screen would open displaying a table showing the 3 pins the user has access to allowing them to edit the path the pin links to in order to comply with **SC15**.



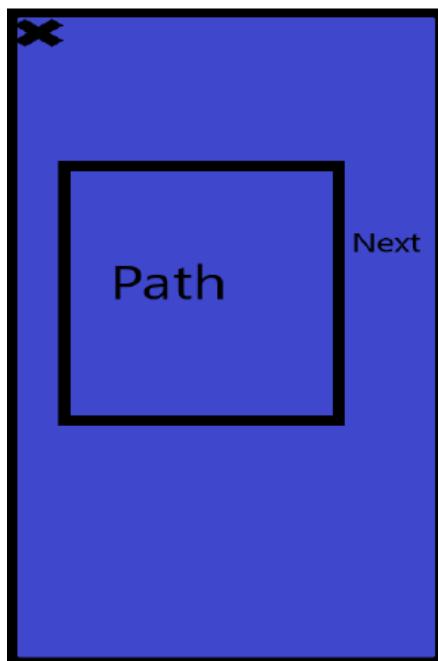
If they were to press on the room linked to a pin they would be linked to the search function allowing them to search for rooms to create a path and store it in the pin complying with **SC14**.



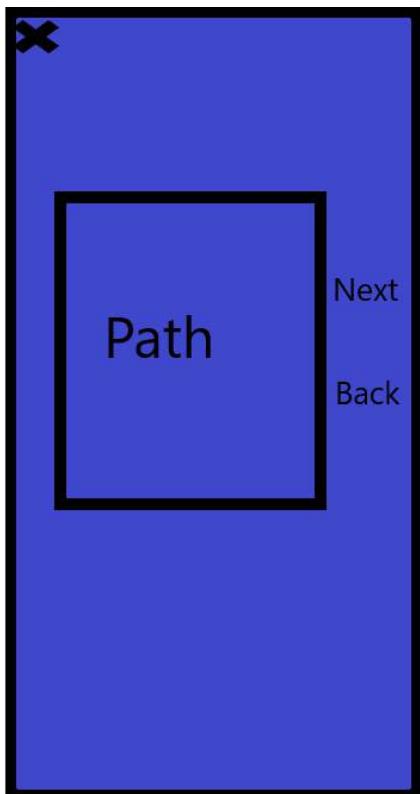
If the user clicks on status they'll be able to search up a room with the search function and once they click on the room they'll be linked to this screen giving them options to change the status of the room by clicking on the appropriate new status allowing the program to comply with **SC13**.



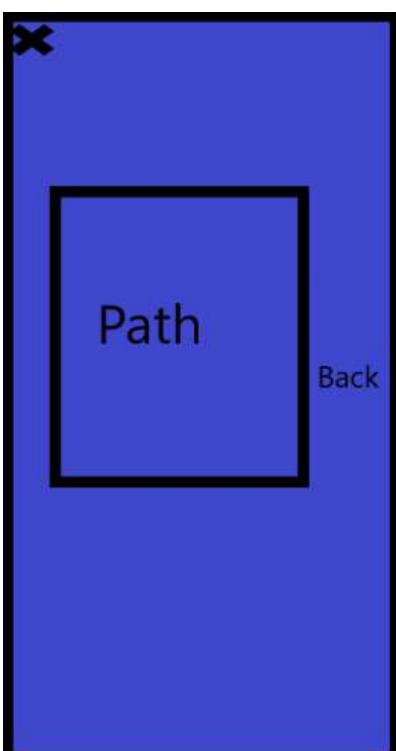
If the user clicks upon the search icon it swaps to the search screen allowing them to utilise the search function in order to get the start and end location for a path to be formed and thus the program can comply with **SC8** and **SC9**



When the pin or search function are used to form a path it first links to the 1st path screen which details the path required for the user to exit the building that they have stated they're in, allowing **SC10** and **SC8** to be achieved.



If next is pressed on path 1 or back is pressed on path 3 it links to the 2nd path screen which details the path required for the user to traverse from the building they were originally in to the building their destination room is in helping achieve **SC10** and further accomplishing **SC8** and **SC9**.



If next is pressed on path2 it links to the 3rd path screen which details the path required for the user to traverse from the entrance of the building their destination is into the destination room helping achieve **SC10** and **SC8**.



If an error of unknown origins has occurred this screen is displayed to the user.



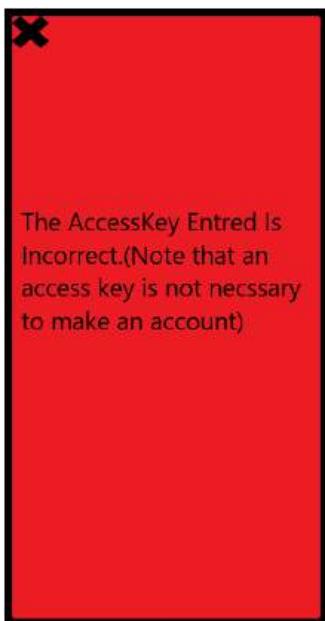
If the user attempts to use a pin with a null value the system displays this screen inorder to teach the user the correct way to use the pin function in order for the system to be able to achieve **SC16**.



If the user clicks upon a likely result that contains a null value it states that there was no matches for the location picked by displaying the screen above in order for the system to be able to achieve **SC16**.



If the user has entered a username or password that does not match the parameters the system advises the user to change it by displaying this screen in order for the system to be able to achieve **SC16**.



If the user enters the incorrect accesskey when creating an account the system displays this fact to the user by displaying this screen and makes sure that they know that it's not necessary to have an access key in order to create an account in order for the system to be able to achieve **SC16**.

Key Variables:

Name	Data Type	What It Stores	Why it is needed	Success Criteria Achieved
Astroturf Entrance (AE1)	Integer	Store the cords corresponding to the node of the Astroturf's Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Astroturf's Entrance in order to accomplish the following success criteria.	SC8,SC9,SC10,SC11
Pavilion Entrance 1 (PE1)	Integer	Store the cords corresponding to the node of the Pavilion's Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Pavilion's Entrance in order to accomplish the	SC8,SC9,SC10,SC11

			following success criteria	
Pavilion Exit 1 (PEX1)	Integer	Store the cords corresponding to the node of the Pavilion's Exit	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Pavilion's Exit in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Davinci Block Entrance 1 (DE1)	Integer	Store the cords corresponding to the node of one of the Davinci Block entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Davinci Block entrance in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Davinci Block Entrance 2 (DE2)	Integer	Store the cords corresponding to the node of one of the Davinci Block entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Davinci Block entrances in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Davinci Block Exit 1 (DEX1)	Integer	Store the cords corresponding to the node of the Davinci Block Exit	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Davinci Block Exit in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Main Hall Entrance 1 (HE1)	Integer	Store the cords corresponding to the node of one of the Main Hall Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Main Hall Entrances in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Main Hall Entrance 2 (HE2)	Integer	Store the cords corresponding to the node of one of the Main Hall Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another	SC8,SC9,SC10,SC11

			one of the Main Hall Entrances in order to accomplish the following success criteria	
Main Hall Exit 1 (HEX2)	Integer	Store the cords corresponding to the node of the Main Hall Exit	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Main Hall Exit in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Sports Hall Entrance 1 (SE1)	Integer	Store the cords corresponding to the node of the Sports Hall Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Sports Hall Entrance in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
English Block Entrance 1 (EE1)	Integer	Store the cords corresponding to the node of the English Block Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the English Block Entrance in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
English Block Exit 1 (EEX1)	Integer	Store the cords corresponding to the node of the English Block Exit	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the English Block Exit in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Maths Block Entrance 1 (ME1)	Integer	Store the cords corresponding to the node of the Maths Block Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Maths Block Entrance in order to accomplish the following success criteria	SC8,SC9,SC10,SC11
Maths Block Exit 1 (MEX1)	Integer	Store the cords corresponding to the node of one of the	So that we can use the path creation algorithm to create and display a path on the user's phone to	SC8,SC9,SC10,SC11

		Maths Block Exit	guide them to in order to accomplish the following success criteria	
Maths Block Exit 2 (MEX2)	Integer	Store the cords corresponding to the node of one of the Maths Block Exit	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to accomplishing the one of the Maths Block Exits following success criteria	SC8,SC9,SC 10,SC11
Main Block Entrance 1(MNE1)	Integer	Store the cords corresponding to the node of one of the Maths Block entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another one of the Maths Block entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Main Block Entrance 2(MNE2)	Integer	Store the cords corresponding to the node of one of the Maths Block entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Maths Block entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Main Block Exit 1 (MNEX 1)	Integer	Store the cords corresponding to the node of the Maths Block exit.	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another the Maths Block exits. in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Language Block Entrance 1(LE1)	Integer	Store the cords corresponding to the node of the Language Block Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Language Block Entrance in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Lower School	Integer	Store the cords corresponding to the node of	So that we can use the path creation algorithm to create and display a path	SC8,SC9,SC 10,SC11

Office Entrance 1(LOE1)		the Lower School Office Entrance .	on the user's phone to guide them to the Lower School Office Entrance in order to accomplish the following success criteria	
Study Centre Entrance 1(SCE1)	Integer	Store the cords corresponding to the node of the Study Centre Entrance	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Study Centre Entrance in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Study Centre Exit 1(SCEX1)	Integer	Store the cords corresponding to the node of one of the Study Centre exits	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Study Centre exists in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Study Centre Exit 2(SCEX2)	Integer	Store the cords corresponding to the node of one of the Study Centre exits	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another one of the Study Centre exists in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Silent Study Entrance 1(SSE1)	Integer	Store the cords corresponding to the node of one of the silent study's entrances	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Silent Study's Entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Silent Study Entrance 1(SSE2)	Integer	Store the cords corresponding to the node of one of the Silent Study's Entrances	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another one of the Silent Study's Entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11

Science Block Entrance 1 (SE1)	Integer	Store the cords corresponding to one of the Science Block Entrances	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to one of the Science Block Entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Science Block Entrance 2 (SE2)	Integer	Store the cords corresponding to the node of a one of the Science Block Entrances	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to another one of the Science Block Entrances in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
Library Exit 1 (LBEX1)	Integer	Store the cords corresponding to the Library's Exit.	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to the Library's Exit in order to accomplish the following success criteria	SC8,SC9,SC 10,SC11
S1	String	Store the cords,status and the closest exit corresponding to S1 within the science block map	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to S1 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success	SC8,SC9,SC 10,SC11,SC13

			criteria	
S2	String	Store the cords,status and the closest exit corresponding to S2 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S2 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
S3	String	Store the cords,status and the closest exit corresponding to S3 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S3 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success 	SC8,SC9,SC10,SC11,SC13

			criteria	
S4	String	Store the cords,status and the closest exit corresponding to S5 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S4 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
S5	String	Store the cords,status and the closest exit corresponding to S5 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S5 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success 	SC8,SC9,SC10,SC11,SC13

			criteria	
S6	String	Store the cords,status and the closest exit corresponding to S6 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S6 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
S7	String	Store the cords,status and the closest exit corresponding to S7 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S7 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success 	SC8,SC9,SC10,SC11,SC13

			criteria	
S8	String	Store the cords,status and the closest exit corresponding to S8 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S8 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
S9	String	Store the cords,status and the closest exit corresponding to S9 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S9 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success 	SC8,SC9,SC10,SC11,SC13

			criteria	
S10	String	Store the cords,status and the closest exit corresponding to S10 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S10 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
S11	String	Store the cords,status and the closest exit corresponding to S11 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S11 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
S12	String	Store the cords,status and the closest exit corresponding to S12 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S12 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
S13	String	Store the cords,status and the closest exit corresponding to S13 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S13 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success 	SC8,SC9,SC10,SC11,SC13

			criteria	
S14	String	Store the cords,status and the closest exit corresponding to S14 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S14 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
S15	String	Store the cords,status and the closest exit corresponding to S15 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S15 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
S16	String	Store the cords,status and the closest exit corresponding to S16 within the science block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to S16 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
604	String	Store the cords,status and the closest exit corresponding to 604 within the language block	<p>So that we can:</p> <ul style="list-style-type: none"> use the path creation algorithm to create and display a path on the user's phone to guide them to 604 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
605	String	Store the cords,status and the closest exit corresponding to 605 within the language block	<p>So that we can:</p> <ul style="list-style-type: none"> .use the path creation algorithm to create and display a path on the user's phone to guide them to 605 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings", "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
606	String	Store the cords,status and the closest exit corresponding to 606 within the language block	<p>So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to 606 in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
607	String	Store the cords,status and the closest exit corresponding to 607 within the language block	<p>So that we can:</p> <ul style="list-style-type: none"> use the path creation algorithm to create and display a path on the user's phone to guide them to 607 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	
608	String	Store the cords, status and the closest exit corresponding to 608 within the language block	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to 608 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
609	String	Store the cords, status and the closest exit corresponding to 609 within the language block	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to 609 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	
610	String	Store the cords, status and the closest exit corresponding to 610 within the language block	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to 610 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
611	String	Store the cords, status and the closest exit corresponding to 611 within the language block	So that we can: use the path creation algorithm to create and display a path on the user's phone to guide them to 611 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
612	String	Store the cords,status and the closest exit corresponding to 612 within the language block	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to 612 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
613	String	Store the cords,status and the closest exit corresponding to 613 within the language block	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to 613 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	
614	String	Store the cords, status and the closest exit corresponding to 614 within the language block	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to 614 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
615	String	Store the cords, status and the closest exit corresponding to 615 within the language block	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to 615 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	
E1	String	Store the cords,status and the closest exit corresponding to E1 within the english block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to E1 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
E2	String	Store the cords,status and the closest exit corresponding to E2 within the english block map	So that we can: . use the path creation algorithm to create and display a path on the user’s phone to guide them to E2 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	
E3	String	Store the cords,status and the closest exit corresponding to E3 within the english block map	So that we can: . use the path creation algorithm to create and display a path on the user’s phone to guide them to E3 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
E4	String	Store the cords,status and the closest exit corresponding to E4 within the english block map	So that we can: . use the path creation algorithm to create and display a path on the user’s phone to guide them to E4 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
E5	String	Store the cords,status and the closest exit corresponding to E5 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to E5 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
E6	String	Store the cords,status and the closest exit corresponding to E6 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> .use the path creation algorithm to create and display a path on the user’s phone to guide them to E6 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
E7	String	Store the cords,status and the closest exit corresponding to E7 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to E7 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
E8	String	Store the cords,status and the closest exit corresponding to E8 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to E8 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
E9	String	Store the cords,status and the closest exit corresponding to E9 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to E9 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
E10	String	Store the cords,status and the closest exit corresponding to E10 within the english block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to E10 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
P1	String	Store the cords,status and the closest exit corresponding to P1 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to P1 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
P2	String	Store the cords,status and the closest exit corresponding to P2 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to P2 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
P3	String	Store the cords,status and the closest exit corresponding to P3 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to P3 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria 	SC8,SC9,SC10,SC11,SC13
P4	String	Store the cords,status and the closest exit corresponding to P4 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> use the path creation algorithm to create and display a path on the user’s phone to guide them to P4 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
P5	String	Store the cords,status and the closest exit corresponding to P5 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user’s phone to guide them to P5 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
P6	String	Store the cords,status and the closest exit corresponding to P6 within the music block map	<p>So that we can:</p> <ul style="list-style-type: none"> .use the path creation algorithm to create and display a path on the user’s phone to guide them to P6 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient 	SC8,SC9,SC10,SC11,SC13

			path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	
P7	String	Store the cords, status and the closest exit corresponding to P7 within the music block map	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to P7 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
P9	String	Store the cords, status and the closest exit corresponding to P9 within the music block map	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to P9 . display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
MA01	String	Store the cords,status and the closest exit corresponding to MA01 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA01</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
MA02	String	Store the cords,status and the closest exit corresponding to MA02 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA02</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient</p>	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
MA03	String	Store the cords,status and the closest exit corresponding to MA03 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA03</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
MA04	String	Store the cords,status and the closest exit corresponding to MA04 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA04</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient</p>	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
MA05	String	Store the cords,status and the closest exit corresponding to MA05 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA05</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
MA11	String	Store the cords,status and the closest exit corresponding to MA11 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA11</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient</p>	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
MA12	String	Store the cords,status and the closest exit corresponding to MA12 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA12</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
MA13	String	Store the cords,status and the closest exit corresponding to MA13 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA13</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient</p>	SC8,SC9,SC10,SC11,SC13

			<p>path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	
MA14	String	Store the cords,status and the closest exit corresponding to MA14 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA14</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site”</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
MA15	String	Store the cords,status and the closest exit corresponding to MA15 within the maths block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user’s phone to guide them to MA15</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient</p>	SC8,SC9,SC10,SC11,SC13

			path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	
MA16	String	Store the cords,status and the closest exit corresponding to MA16 within the maths block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to MA16 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
11	String	Store the cords,status and the closest exit corresponding to room 11 within the main block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to room 11 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	
12	String	Store the cords,status and the closest exit corresponding to room 12 within the main block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to room 12 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
13	String	Store the cords,status and the closest exit corresponding to room 13 within the main block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to room 13 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	
14	String	Store the cords,status and the closest exit corresponding to room 14 within the main block map	So that we can: . use the path creation algorithm to create and display a path on the user’s phone to guide them to room 14 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient path to the users destinations” ,”Show an accurate layout of the inside of the school buildings”, ”Show an accurate layout of the site” in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
15	String	Store the cords,status and the closest exit corresponding to room 15 within the main block map	So that we can: use the path creation algorithm to create and display a path on the user’s phone to guide them to room 15 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of “being able to guide the user on the most efficient	SC8,SC9,SC10,SC11,SC13

			path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	
16	String	Store the cords, status and the closest exit corresponding to room 16 within the main block map	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to room 16 in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
21	String	Store the cords, status and the closest exit corresponding to room 21 within the main block map	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to room 21 in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
22	String	Store the cords, status and the closest exit corresponding to room 22 within the main block map	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to room 22 in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
23	String	Store the cords, status and the closest exit corresponding to room 23 within the main block map	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to room 23 in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
24	String	Store the cords, status and the closest exit corresponding to room 24 within the main	So that we can use the path creation algorithm to create and display a path on the user's phone to guide them to room 24 in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13

		block map		
25	String	Store the cords,status and the closest exit corresponding to room 25 within the main block map	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to room 25 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13
26	String	Store the cords,status and the closest exit corresponding to room 26 within the main block map	So that we can: . use the path creation algorithm to create and display a path on the user's phone to guide them to room 26 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success	SC8,SC9,SC10,SC11,SC13

			criteria	
DT1	String	Store the cords,status and the closest exit corresponding to DT1 within the davinci block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user's phone to guide them to DT1</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site"</p> <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
DT2	String	Store the cords,status and the closest exit corresponding to DT2 within the davinci block map	<p>So that we can:</p> <p>use the path creation algorithm to create and display a path on the user's phone to guide them to DT2</p> <p>.display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site"</p> <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
DT3	String	Store the cords,status and the closest exit corresponding to DT3 within the davinci block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to DT3 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
DT4	String	Store the cords,status and the closest exit corresponding to DT4 within the davinci block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to DT4 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
DT5	String	Store the cords,status and the closest exit corresponding to DT5 within the davinci block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to DT5 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success criteria</p>	SC8,SC9,SC10,SC11,SC13
A1	String	Store the cords,status and the closest exit corresponding to A1 within the davinci block map	<p>So that we can:</p> <ul style="list-style-type: none"> . use the path creation algorithm to create and display a path on the user's phone to guide them to A1 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" <p>in order to accomplish the following success</p>	SC8,SC9,SC10,SC11,SC13

			criteria	
A2	String	Store the cords,status and the closest exit corresponding to A2 within the davinci block map	So that we can: .use the path creation algorithm to create and display a path on the user's phone to guide them to A2 .display the status of the room so that it can allow the user to make an informative decision on whether or not to travel to the room allowing the solution to accomplish the software features of "being able to guide the user on the most efficient path to the users destinations" , "Show an accurate layout of the inside of the school buildings" , "Show an accurate layout of the site" in order to accomplish the following success criteria	SC8,SC9,SC10,SC11,SC13

Data Structures Explained:

The data structure needed is an dictionary which will allow the solution to:

-store the data of created accounts by reading a text file where each key represents one user with the first value being the account name, second value being the account password, the 3rd -6th values being the pins stored and the 7th value being the access key if they have one.

-Stores data of rooms to dictionaries named after the building they're in with the data of the buildings being stored in a separate text file . Each key in the building dictionaries represent a room in the building with the first value being the room's x cord on the maps ,the second value being the room's y value on the map , the third value being the status of the room and the last value being the closest exit to the room

Validation Table:

In order to reduce the chance of errors I will add these validations in order to ensure that the data being entered has a low chance of causing errors.

Validation	Explanation
Text Boxes	The text boxes in which text is inputted from the user limits the input to a word limit of 16 to ensure that data being saved isn't overly long so that solution complies with SC17
Likely result	When entering an input into the search function the search function shall output the 4 most likely results so that in case the user has mistyped their input they can most likely correct themselves allowing the solution to compile with SC16
Log In Error Screen	When making an account or signing into an account if the user enters the wrong details the relevant error screen is screen blit instructing the user about where they went wrong when inputting their login details allowing the solution to compile with SC16 and properly complete SC6,SC7
Pin Error Screen	When trying to use a pin with a null value an error screen is screen blit notifying the user that the pin is empty and to use they need to fill the pin allowing the solution to compile with SC16

Test Table Alpha(During Development):

Test Number	Test Data	Success Criteria	Explanation

Room Search			
DT1	Search: S13	SC8,SC9,SC16	To check if the system outputs the correct room when searched for.
DT2	Search: S1	SC8,SC9	Checks to make sure program does not output any corrections to an input that wouldn't have any
Pins			
DT3	Save S12 to E2 to pin1	SC14,SC15	Makes pin1 = S12 to E2 and then reloads the program to see if it saved the location.
Route Creation			
DT4	Search: S12 to E2	SC8,SC9,SC10,SC16	Makes the system calculate a route from users location to room 13 to see if it makes a highlighted route within the given amount of time.
DT5	Click a Pin with the value s12 to E2	SC8,SC9,SC10	Makes the system calculate a route and makes a highlighted route within a given amount of time from the already stored starting position and end position
DT6	Search: Room 16 to Dt2	SC8,SC9,SC10	Makes the system calculate a route from users location to room 13 to see if it can make a highlighted route

			even if the route contains stairs.
DT7	Click Pin with the value Room 16 to Dt2	SC8,SC9,SC10	Makes the system calculate a route from users location to room 13 to see if it can make a highlighted route even if the pin's stored route contains stairs.
Status			
DT8	Room 12 Status = Red And Search: Room 12	SC12,SC13	Checks to see if the system saves the new status of room 12.
Buttons			
DT9	Click Start	SC1,SC3	To check if program can open the main screen
DT10	Click Exit	SC1,SC2	To check if program closes as expected and
DT11	Click Log In	SC1,SC3	To check if program can open the log in screen
DT12	Click a Pin with no value	SC14,SC15,SC16	To see if the program produces correct error message when trying to produce a route for no value
DT13	Click Search Icon	SC3	To check if program opens search screen
DT14	Click Menu Button while logged out	SC3	To check if program opens appropriate menu

DT15	Click Menu Button while logged in	SC3	To check if program opens appropriate menu
DT16	Click Menu Button while logged into an admin account	SC3,SC13	To check if program opens appropriate menu
DT17	Click Close Menu Button	SC3	To check if program closes menu upon the button being clicked
DT18	Click Home Button	SC3	To check if the program goes to the start screen upon the button being clicked
DT19	Click Pins Button	SC3,SC14	To check if program swaps to the pin search screen in order to search for a start and end position
DT20	Click Status Button	SC3,SC13	To check if program swaps to status search screen in order to search for a room to switch the status of
DT21	Click Menu LogIn Button	SC3,SC4	To check if program swaps to the login screen from the menu in order for the user to log in
DT22	Click LogOff Button	SC3,SC5	To check if the program swaps to the starting screen from the menu screen and signs the users out

DT23	Click Next Button	SC3	To check if the program can go to the correct next path screen
DT24	Click Back Button	SC3	To check if the program can go back to the correct previous path screen
Text Length			
DT25	Attempt to write “ProofThatThere IsAMaxLength”	SC17	To check if the program has a maximum character length of 16 in order to compile with the text box validation
Log In			
DT26	Log into a test user	SC4,SC7,SC3	To check if the program signs into account correctly and updates the necessary details of the system from the accounts details
DT27	Log into an account that has not been created	SC4,SC7,SC3,SC16	To check if the program produces the correct error message when user tries to sign into an unknown account
Sign Up			
DT28	Create an account with no access key	SC6	To check if an account can be created and if the account gets the correct level of authorization
DT29	Create an	SC6,SC7	To check if an

	account with access key		account can be created and if the account gets the correct level of authorization
DT30	Create an account with incorrect access key	SC6,SC16	To check if program produces the correct error message when the access key inputted is incorrect.

Test Table Beta (Post Development):

Test Number	Test Data	Success Criteria	Explanation
Room Search			
PT1	Search: MA01	SC8,SC9,SC16	To check if the system outputs the correct room when searched for.
PT2	Search: E1	SC8,SC9	Checks to make sure program does not output any corrections to an input that wouldn't have any
Pins			
PT3	Save MA01 to P2 pin1	SC14,SC15	Makes pin1 = MA12 to P2 and then reloads the program to see if it saved the location.
Route Creation			
PT4	Search: MA01 to	SC8,SC9,SC10,S	Makes the system

	P2	C16	calculate a route from users location to room 13 to see if it makes a highlighted route within the given amount of time.
PT5	Click a Pin with the value MA01 to P2	SC8,SC9,SC10	Makes the system calculate a route and makes a highlighted route within a given amount of time from the already stored starting position and end position
PT6	Search: MA12 to P2	SC8,SC9,SC10	Makes the system calculate a route to see if it can make a highlighted route even if the route contains stairs.
PT7	Click Pin with the value MA12 to P2	SC8,SC9,SC10	Makes the system calculate a route to see if it can make a highlighted route even if the pin's stored route contains stairs.
Status			
PT8	MA12 Status = Red And Search: MA12	SC12,SC13	Checks to see if the system saves the new status of room 12.
Buttons			
PT9	Click Start	SC1,SC3	To check if program can open the main screen
PT10	Click Exit	SC1,SC2	To check if

			program closes as expected and
PT11	Click Log In	SC1,SC3	To check if program opens a new screen
PT12	Click a Pin with no value	SC14,SC15,SC16	To see if the program produces correct error message when trying to produce a route for no value
PT13	Click Search Icon	SC3	To check if program opens search screen
PT14	Click Menu Button while logged out	SC3	To check if program opens appropriate menu
PT15	Click Menu Button while logged in	SC3	To check if program opens appropriate menu
PT16	Click Menu Button while logged into an admin account	SC3,SC13	To check if program opens appropriate menu
PT17	Click Close Menu Button	SC3	To check if program closes menu upon the button being clicked
PT18	Click Home Button	SC3	To check if the program goes to the start screen upon the button being clicked
PT19	Click Pins Button	SC3,SC14	To check if program swaps to the pin search screen in order to search for a start

			and end position
PT20	Click Status Button	SC3,SC13	To check if program swaps to status search screen in order to search for a room to switch the status of
PT21	Click Menu LogIn Button	SC3,SC4	To check if program swaps to the login screen from the menu in order for the user to log in
PT22	Click LogOff Button	SC3,SC5	To check if the program swaps to the starting screen from the menu screen and signs the users out
PT23	Click Next Button	SC3	To check if the program can go to the correct next path screen
PT24	Click Back Button	SC3	To check if the program can go back to the correct previous path screen
Text Length			
PT25	Attempt to write “ProofThatThere IsAMaxLength”	SC17	To check if the program has a maximum character length of 16 in order to compile with the text box validation
Log In			
PT26	Log into a test user	SC4,SC7,SC3	To check if the program signs into account correctly

			and updates the necessary details of the system from the accounts details
PT27	Log into an account that has not been created	SC4,SC7,SC3,SC16	To check if the program produces the correct error message when user tries to sign into an unknown account
Sign Up			
PT28	Create an account with no access key	SC6	To check if an account can be created and if the account gets the correct level of authorization
PT29	Create an account with access key	SC6,SC7	To check if an account can be created and if the account gets the correct level of authorization
PT30	Create an account with incorrect access key	SC6,SC16	To check if program produces the correct error message when the access key inputted is incorrect.
PT31	Screenshot the properties of the application	SC17	To check if the application takes under 5MB of storage
PT32	Screenshot the text file containing the rooms	SC11	To check if the application contains all rooms on the site

Development:

Breakdown:

As displayed in the gantt chart and X I have split the development section into the following 4 tasks:

Task 1:Consists of sub programs made in order to make a functioning main menu

- A)Buttons
- B)Start Screen
- C)Exit
- D)Log In Screen
- E)Log In Function
- F)Sign Up Screen
- G)Account save function
- H>Main Screen

Task 2:Consists of 1 sub program

- A)Search Screen
- B)Search Function
- C)RoomStatusFinder
- D)Destination Screen

Task 3: Consists of subprograms that form the menu choices based on authorisation

- A) Menu Button
- B)Close Menu Button
- C)Home Button
- D)LogIn
- E)Log Off
- F)Pins
- G>Status

Task 4:Consists of subprograms used to manipulate path finder visualizer

- A)Map Identifier Function
- B)Node Identifier Function
- C)Inside Path Creator Function
- D)Outside Path Creator Function
- E)Path Screens

In order to be able to develop iteratively via the agile paradigm as this development process will enable me to develop in an efficient manner and get necessary feedback from clients at appropriate time intervals.

Task 1:

Development:

15/04/22:

Although task 1 only consists of the main menu there are multiple sub programs needed to make the main menu functioning as shown in the top down diagram in design and thus my first priority was to develop a function that would enable me to efficiently create these sub programs as buttons in order to fulfil **SC1,SC2,SC3**.

To do this I first imported the python library pygame and entered the required details as it gave me the necessary landscape for such a function to run on as it gives access to useful ui creating functions such as “pygame.display.Info()” and “pygame.display.set_mode” which enabled me to fit the program to the user’s screen allowing it to run on all devices fulfilling **SC18**.

```
import pygame, sys # Pygame is the Library we're using to display the maps upon and sys lets us access system-specific parameters and functions  
|  
## -----Pygame Setup-----##  
pygame.init() #initializes pygame  
info = pygame.display.Info() #gets info about user's screen  
Width = info.current_w #screen's width  
Height = info.current_h# screen's height  
Center_X = Width/2#X cords for the center of the screen  
Center_Y = Height/2#Y cords for the center of the screen  
Screen = pygame.display.set_mode((Width,Height)) #app now fit's to user's screen  
clock = pygame.time.Clock()
```

Although pygame gives ui functions it however does not provide a function that could efficiently create my desired buttons so to work around this first I needed to create the

button class to allow me to create a ui interface to start the subprograms.

```
## -----Screen Creator-----
def get_font(size):#Returns text in desired size
    return pygame.font.Font("BACKTO19.ttf", int(size))

class Button():#class made to effentially create buttons
    def __init__(self, image, pos, text_input, font, base_color, hovering_color):
        self.image = image
        self.x_pos = pos[0]
        self.y_pos = pos[1]
        self.font = font
        self.base_color, self.hovering_color = base_color, hovering_color
        self.text_input = text_input
        self.text = self.font.render(self.text_input, True, self.base_color)
        if self.image is None:
            self.image = self.text
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))
        self.text_rect = self.text.get_rect(center=(self.x_pos, self.y_pos))

    def update(self, screen):#puts image and text of button upon the screen
        if self.image is not None:
            screen.blit(self.image, self.rect)
            screen.blit(self.text, self.text_rect)

    def checkForInput (self, position):#checks if client is clicking the button
        if position[0] in range (self.rect.left, self.rect.right) and position[1] in range (self.rect.top, self.rect.bottom):
            return True
        return False

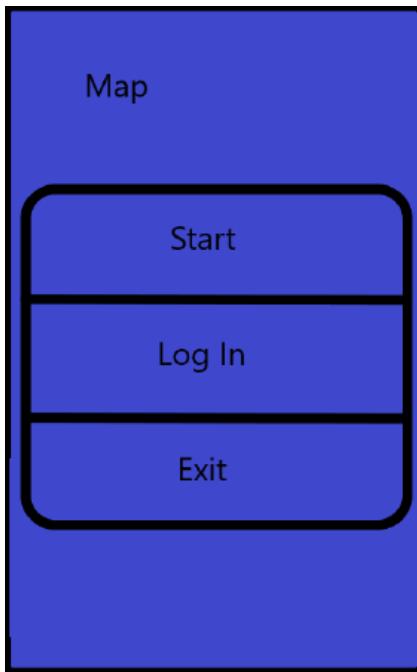
    def changeColor(self, position):#checks if client is hovering over the button and if so the button changes colour
        if position[0] in range(self.rect.left, self.rect.right) and position[1] in range(self.rect. top, self.rect.bottom):
            self.text = self.font.render(self.text_input, True, self.hovering_color)
        else:
            self.text = self.font.render(self.text_input, True, self.base_color)
```

Sub Modules of Task1	Completed
A	Yes
B	No
C	No
D	No
E	No
F	No
G	No
H	No

Success Criteria	Achieved partially or fully	Explanation
S18	Partially	By making the screen size dependent on the devices width and height the application can be used on a mobile device.

16/04/22 -18/04/22:

I applied the button function by attempting to create the first necessary screen for the app benign the starting screen:



```

def Start_Sc():
    while True:
        Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
        Screen.blit(MainMap,(0,0))

        Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

        START_Button = Button(image = None, pos = (Center_X + (Width)/5 ,Center_Y +(Height)/5)
                               ,text_input = "START", font = get_font((Height)/20), base_color = "#000317", hovering_color = "Yellow")

        LOGIN_Button = Button(image = None, pos = (Center_X, Center_Y) ,text_input = "LOGIN", font = get_font((Height)/20)
                               ,base_color = "#000317", hovering_color = "Yellow")

        EXIT_Button = Button(image = None, pos =(Center_X -(Width)/5 , Center_Y -(Height)/5)
                               ,text_input = "Exit", font = get_font((Height)/20), base_color = "#000317", hovering_color = "Yellow")

        for Button in [START_Button, LOGIN_Button, EXIT_Button]:# goes through each button and uploads it onto the screen
            Button.changeColor(Start_Mouse_Pos)
            Button.update(Screen)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
                if START_Button.checkForInput(Start_Mouse_Pos):
                    Main_Sc()
                if LOGIN_Button.checkForInput(Start_Mouse_Pos):
                    Login_Sc()
                if EXIT_Button.checkForInput(Start_Mouse_Pos):
                    pygame.quit()
                    sys.exit()

        pygame.display.update()
        clock.tick(60)
    
```

however when attempting to use the buttons the screen would freeze as an error had occurred:

```
Traceback (most recent call last):
  File "\sjw-fs-stu01.latrust.org.uk\SJWStudentsData$\09oluwti01\Documents\SJWM
MAPS\Test2.py", line 95, in <module>
    Start_Sc()
  File "\sjw-fs-stu01.latrust.org.uk\SJWStudentsData$\09oluwti01\Documents\SJWM
MAPS\Test2.py", line 65, in Start_Sc
    START_Button = Button(image = None ,pos = (Center_X +(Width)/5 ,Center_Y +(H
ight)/5) ,text_input = "START", font = get_font((Height)/20), base_color = "#000
17", hovering_color = "Yellow")
UnboundLocalError: local variable 'Button' referenced before assignment
```

The error occurs as the buttons were looked at as a local variable and thus could not be defined within the modules for the screens and thus I fixed it by simply creating the buttons outside the modules and referring to the needed ones inside the module screens which they were needed

```
#Buttons
START_Button = Button(image = None ,pos = (Center_X +(Width)/5 ,Center_Y +(Height)/5) ,text_input = "START", font = get_font((Height)/20), base_color = "#000117", hovering_color = "Yellow")
LOGIN_Button = Button(image = None, pos = (Center_X, Center_Y) ,text_input = "LOGIN", font = get_font((Height)/20), base_color = "#000117", hovering_color = "Yellow")
EXIT_Button = Button(image = None, pos = (Center_X -(Width)/5 , Center_Y -(Height)/5) ,text_input = "Exit", font = get_font((Height)/20), base_color = "#000117", hovering_color = "Yellow")
```

After fixing the error I was able to confirm that the button function works as I was able to recreate the screen design as shown below:



I then coded the “log in” screen after the “exit” sub program as the “log in” screen is recalled upon in the sub programs following the “start” subprogram as shown in the top down design. I did so by creating text boxes for the

```

def Login_Sc():
    global Username_Input , Password_Input , Username_Rect, Password_Rect , Createaccount_Button , X_Button , Check_Button ,Username_Type ,Password_Type ,loggedin
    while True:
        Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

        for Button in [Username_Button,Password_Button,Createaccount_Button , X_Button , Check_Button]:# goes through each button and uploads it onto the screen
            Button.changeColor(Start_Mouse_Pos)
            Button.update(Screen)

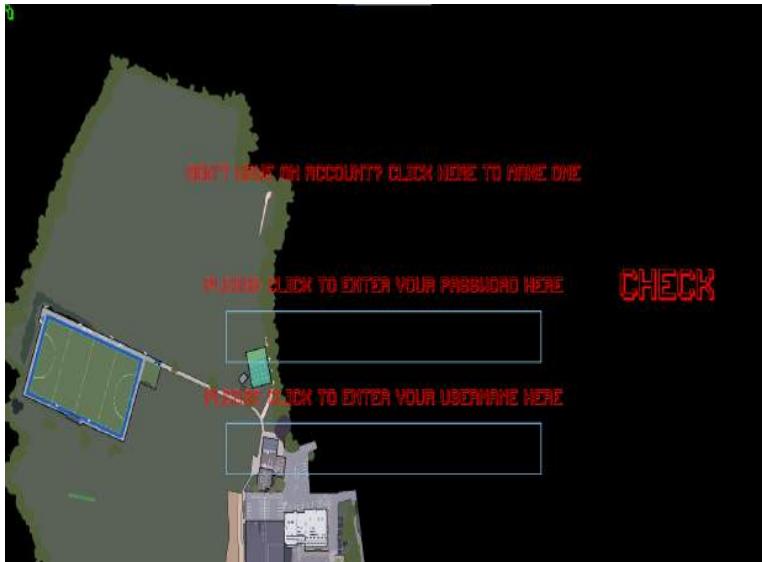
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
                if Username_Button.checkForInput(Start_Mouse_Pos) and Password_Type == False:#Allows user to type if clicked unless the password button has also been clicked
                    Username_Type = True
                if Password_Button.checkForInput(Start_Mouse_Pos) and Username_Type == False:#Allows user to type if clicked unless the username button has also been clicked
                    Password_Type = True
                if Createaccount_Button.checkForInput(Start_Mouse_Pos):#Goes to sign up screen if clicked
                    SignUp_Sc()
                if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
                    Start_Sc()

            if Username_Type == True:#if the username button has been clicked the user can now type into the username rectangle
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_BACKSPACE:
                        Username_Input = Username_Input[0:-1] #Username input is now from the initial character to the 2nd to last character essentially deleting the last char
                    elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
                        Username_Type = False
                    elif int(len(Username_Input)) > 16:#Stops them from writing outside the text box's width
                        Username_Type = False
                    else:
                        Username_Input += event.unicode
                        Username_Input = Username_Input.upper()

            if Password_Type == True:#if the password button has been clicked the user can now type into the password rectangle
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_BACKSPACE:
                        Password_Input = Password_Input[0:-1] #Username input is now from the initial character to the 2nd to last character essentially deleting the last char
                    elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
                        Password_Type = False
                    elif int(len(Password_Input)) > 16:#Stops them from writing outside the text box's width
                        Password_Type = False
                    else:
                        Password_Input += event.unicode
                        Password_Input = Password_Input.upper()

        pygame.display.update()
        clock.tick(60)
    
```

Test Performed	Test Evidence	Pass Or Fail
DT10		Pass

DT11	 	Pass
DT25		Pass

Sub Modules of Task1	Completed
-----------------------------	------------------

A	Yes
B	Yes
C	Yes
D	Yes
E	No
F	No
G	No
H	No

Success Criteria	Achieved partially or fully	Explanation
SC1	Partially	As shown in the DT11 and DT10 both the exit and log in button are functioning and in their correct position within the start screen.
SC2	Fully	As shown in test DT10 application ends when exit button prest
SC3	Fully	As show in the test DT11 the application is capable of switching screens if the button is pressed
SC17	Partially	As shown in DT25 the text box complies with the text box validation of preventing the user writing inputs over 16 characters

20/04/22-23/04/22:

As stated in the design I will be reading and storing data from text files to store accounts in order to fulfil **SC4,SC5,SC6,SC7** and thus the first thing I needed to do is create a text file for the accounts as shown below:



File Edit Format View Help

TESTUSER PASSWORD123 Key321 SC12 N/a N/a

I filled the text file with only a test user so that I can use the account for testing with the order of the text being purposeful so that I can read each line as a separate user's data as I stated in the design's data structure allowing the following code to be used to read the file:

```

#Textfiles:
User_List = [] #list that stores all usernames
Password_List = [] #list that stores all user's passwords to compare to the entered username to see if they're allowed to sign in or not
AccessKey_List = [] #list that stores all user's access keys to identify their lvl of authorization
NumberOfPins_List = [] #list that stores all user's pins so they can be applied when the respective user is signed in
Pins_List = []
with open("Accounts.txt") as f:
    for line in f:#for each line in the file assign the account data to the array accordingly
        (username,password,key,pin1, pin2 , pin3) = line.split()
        User_List.append(username)
        Password_List.append(password)
        AccessKey_List.append(key)
        Pins_List.append(pin1)
        Pins_List.append(pin2)
        Pins_List.append(pin3)

```

I then created a function in order to compare the inputted username and password to the one's in the list with the program checking the username list first and then the respective password of the username is correct so that the correct error message can be produced in the instance of an error as shown in the code below:

```

## -----Log In-----##
Username_Input = ""
Password_Input = ""
AccessKey_Input = ""

def LOGIN_Check(Username, Password):
    global Username_Check
    global Password_Check
    x = str(Username) in User_List#X becomes true if the inputed username is within the usernames array in the accounts dictionary
    y = str>Password_Input) in Password_List#Y becomes true if the inputed password is within the passwords array in the accounts dictionary
    if x == True:#Username and password check allows us to see if the username is the problem or the password is the problem
        Username_Check = True
    if y == True:
        Password_Check = True

```

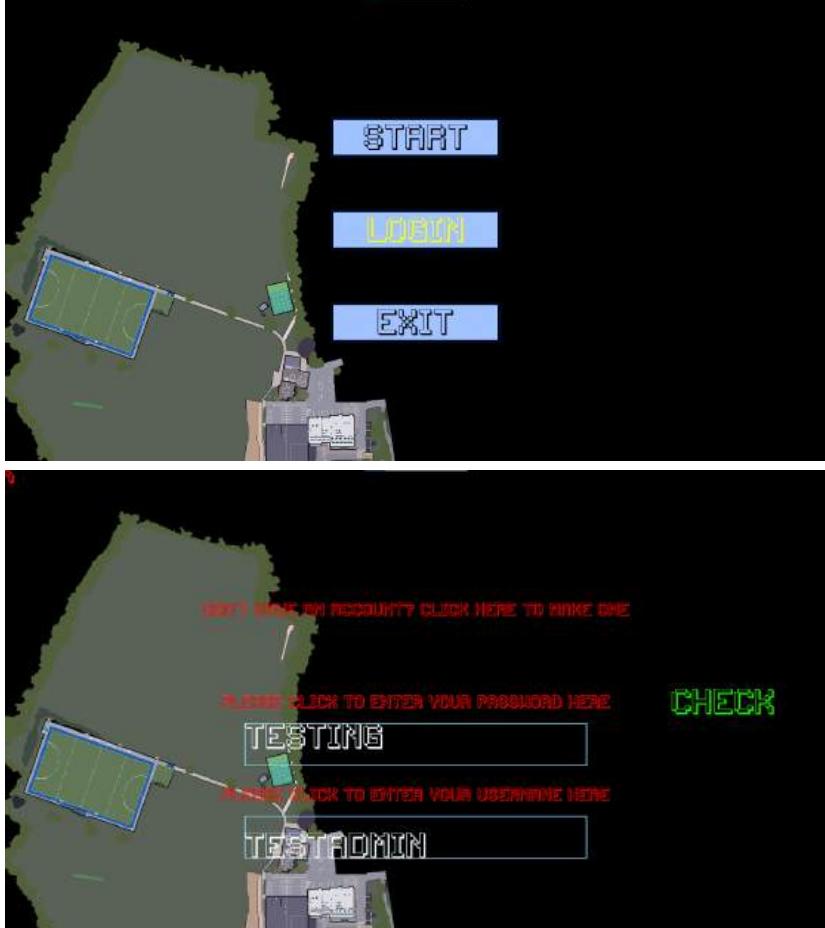
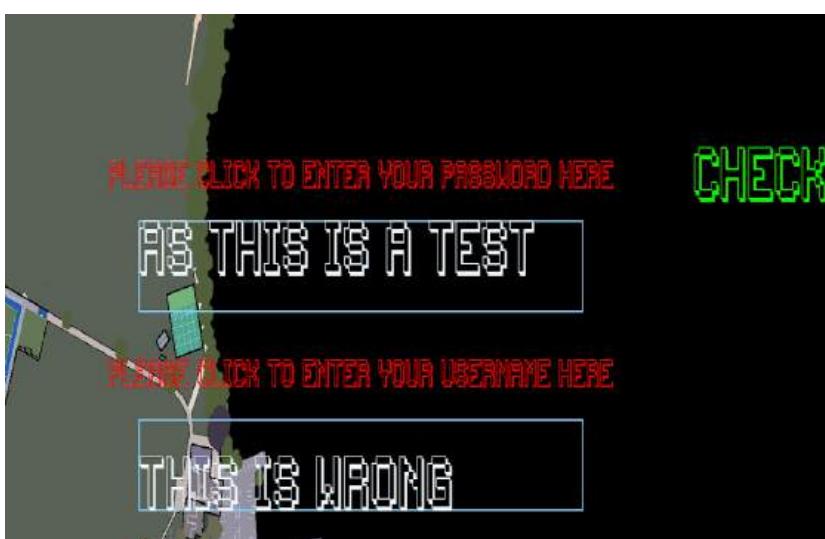
I then implemented the LOGIN_Check function into the log in screen as shown below:

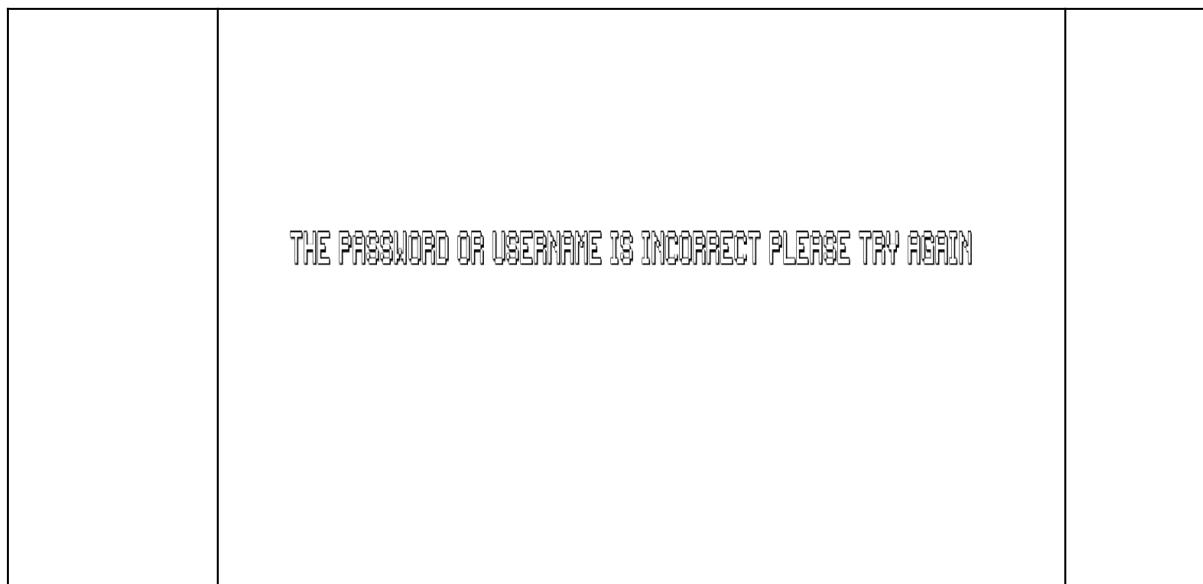
```

~~~~~
if Check_Button.checkForInput(Start_Mouse_Pos):#Checks if mouse is clicking check button to start stament below
    LOGIN_Check(Username_Input,Password_Input)#Checks if the username and password is correct and if so it signs the user in
    if Username_Check == True and Password_Check == True:
        LoggedIn = True
        Main_Sc()#sends to main screen if both the username and password are true
        break
    elif Password_Check == False or Username_Check == False:
        Fail_Sc()#sends to error message screen telling user that they've inputed the incorrect data requirements
    else:
        Error_Sc()#displays an error if an unkown error occurs

```

Test Performed	Test Evidence	Pass Or Fail
----------------	---------------	--------------

DT26		Pass
DT27		Pass



Success Criteria	Achieved partially or fully or again	Explanation
SC3	Again	As shown in the test DT26 the application is capable of switching screens if the button is pressed as it switches to an error message screen after check is pressed
SC16	Fully	As shown in the test DT27 the application is capable of completing the Log In Error Screen Validation of advising the user in correcting their mistakes displayed as the system tries to correct the user as they entered the wrong log in details.

Sub Modules of Task1	Completed
A	Yes
B	Yes
C	Yes
D	Yes

E	Yes
F	No
G	No
H	No

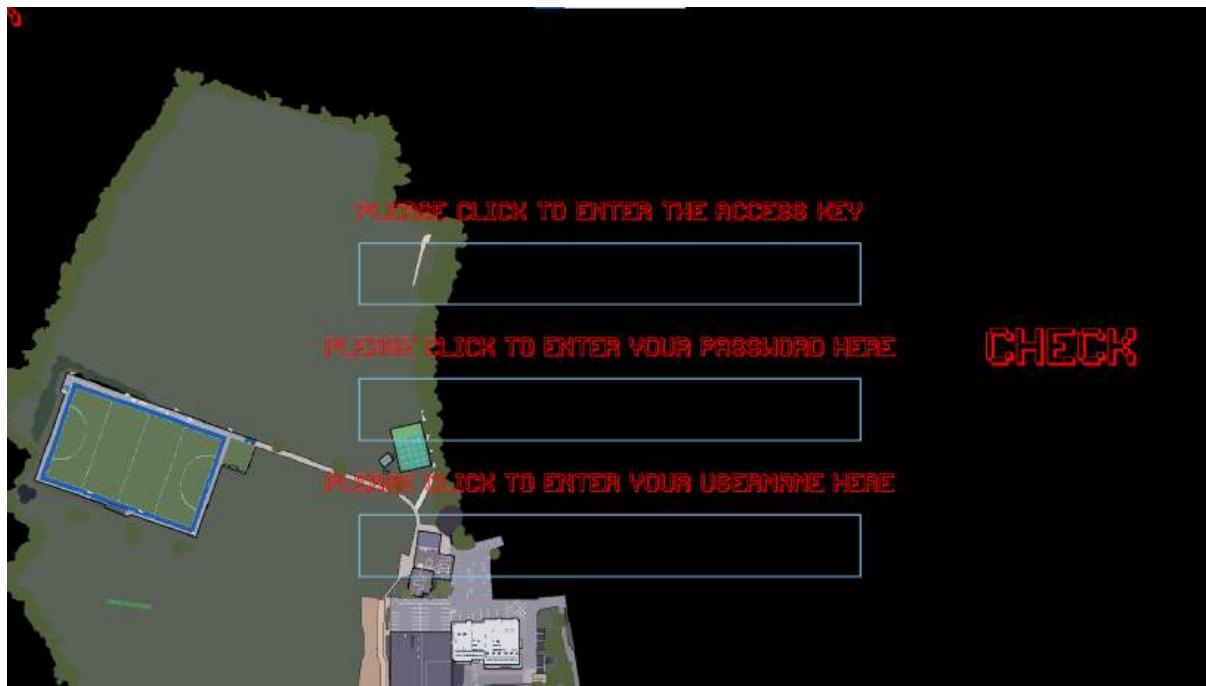
24/04/22-26/04/22:

Used button class to code sign in screen in the same manner as the log in screen as theres only two major differences being the purpose of the check function and the new area to enter the access key. The check function now simply appends the user's inputs to the respective arrays for the username password and potential access key given that the access key is correct or hasn't been given in order for the code to comply with **SC6** as shown in the code below:

```

if Check_Button.checkForInput(Start_Mouse_Pos);#Checks if the username and password is correct and if so it signs the user into the new account after saving the account details
    if accessKey_Input == "" or accessKey_Input == "ADMIN56":
        User_List.append(username_Input)
        Password_List.append(password_Input)
        AccessKey_List.append(accessKey_Input)
        Pins_List.append("0")
        Pins_List.append("0")
        Pins_List.append("0")
        Store_Accounts()
        LoggedIn = True
        Main_Sc()
    elif Password_Input == "" or Username_Input == "":
        False_Sc2()
    else:
        False_Sc()

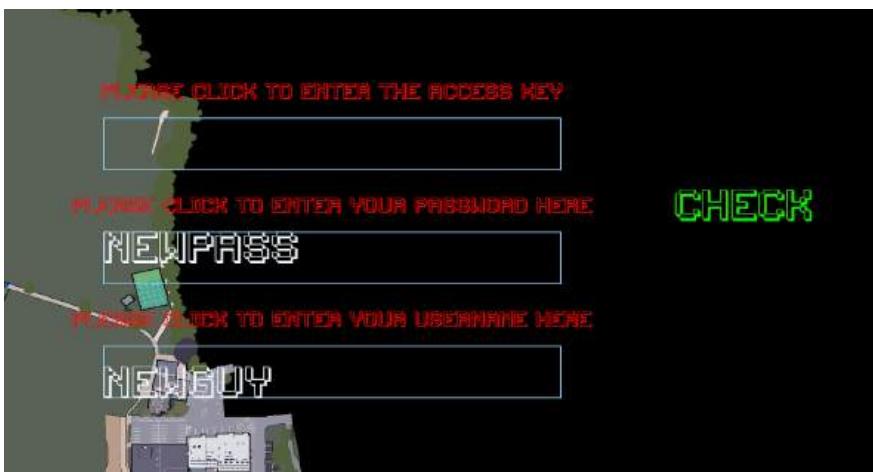
```

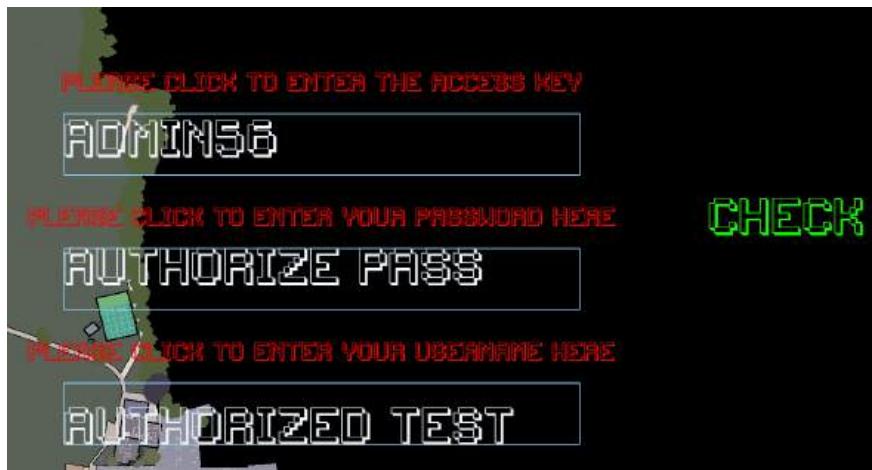
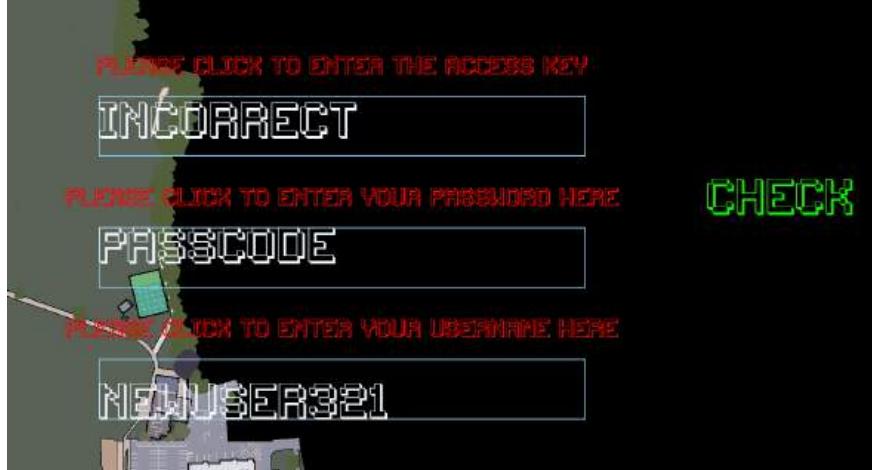


But when it came to saving the new account's data I had to wipe the account text file so I could overlay it with the new account and the previous accounts stored in the arrays as shown in the code below:

```
def Store_Accounts():
    global User_List
    global Password_List
    global AccessKey_List
    global NumberOfPins_List
    global Pins_List
    with open("accounts.txt") as f:
        f.truncate(0) #text file gets wiped
    with open("accounts.txt", 'a') as f:
        for i in range(len(User_List)):#accounts's respective username ,password ,access key and pins are stored as words in the text file on the same line
            f.write(str(User_List[i - 1]) + " " + str(Password_List[i - 1]) + " " + str(AccessKey_List[i - 1]) + " " + str(Pins_List[i - 1]) + " " + str(Pins_List[i]) + " " + str(Pins_List[i + 1]))
        f.write("\n")#new line for new account
```

Test Performed	Test Evidence	Pass Or Fail
----------------	---------------	--------------

DT28	 <p>Traceback (most recent call last):</p> <pre>File "C:\Users\Tireni\Downloads\SJWMS MAPS\sjwms maps code.py", line 41, in module> (username,password,key,pin1, pin2 , pin3) = line.split() ValueError: not enough values to unpack (expected 6, got 5)</pre> <p>Accounts - Notepad</p> <p>File Edit Format View Help</p> <p>NEWGUY NEWPASS 0 SC12 N/a</p> <p>TESTUSER PASSWORD123 Key321 SC12 N/a N/a</p>	Fail
------	--	------

DT29	 <p>PLEASE CLICK TO ENTER THE ACCESS KEY</p> <p>ADMIN56</p> <p>PLEASE CLICK TO ENTER YOUR PASSWORD HERE</p> <p>AUTHORIZE PASS</p> <p>PLEASE CLICK TO ENTER YOUR USERNAME HERE</p> <p>AUTHORIZED TEST</p> <p>CHECK</p> <pre>Accounts - Notepad File Edit Format View Help AUTHORIZED TEST AUTHORIZE PASS ADMIN56 0 0 0 TESTUSER PASSWORD123 AD321 0 0 0 Traceback (most recent call last): File "\\\sjw-fs-stu01.latruct.org.uk\sjwStudentsData\\$\\09oluwti01\Documents\SJWMS\sjwms Maps Code.py", line 41, in <module> (username,password,key,pin1, pin2 , pin3) = line.split() ValueError: too many values to unpack (expected 6)</pre>	Fail
DT30	 <p>PLEASE CLICK TO ENTER THE ACCESS KEY</p> <p>INCORRECT</p> <p>PLEASE CLICK TO ENTER YOUR PASSWORD HERE</p> <p>PASSCODE</p> <p>PLEASE CLICK TO ENTER YOUR USERNAME HERE</p> <p>NEWUSER321</p> <p>CHECK</p> <p>THE ACCESSKEY ENTERED IS INCORRECT (NOTE THAT AN ACCESS KEY IS NOT NECESSARY TO MAKE AN ACCOUNT)</p>	Pass



Sub Modules of Task1	Completed
A	Yes
B	Yes
C	Yes
D	Yes
E	Yes
F	Yes
G	No
H	No

27/04/22:

As shown from the test completed during the dates of **24/04/22-26/04/22** an error occurs as when unpacking the text file the program is expecting 6 words per line as the 6 variables to store but when I created the new user the new user uploaded 8 words into a line as show below:



File Edit Format View Help

```
AUTHORIZED TEST AUTHORIZE PASS ADMIN56 0 0 0
TESTUSER PASSWORD123 AD321 0 0 0
```

```

Traceback (most recent call last):
  File "\sjw-fs-stu01.latrust.org.uk\SJWStudentsData\$\\09oluwti01\Documents\S
S MAPS\Sjwms Maps Code.py", line 41, in <module>
    (username,password,key,pin1, pin2 , pin3) = line.split()
ValueError: too many values to unpack (expected 6)

```

This was due to the fact that the user is able to create spaces causing there to be multiple words for their username and password and thus to fix this error I simply had to remove their option of being able to have spaces in their username ,password and access key by replacing the spaces they have with no spaces as shown below:

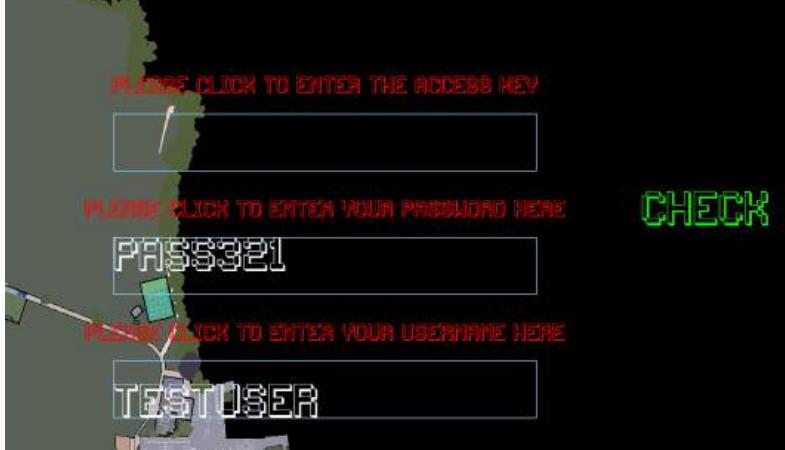
```

AccessKey_Input = AccessKey_Input.replace(" ", "")#removes spaces

Username_Input = Username_Input.replace(" ", "")#removes spaces

Password_Input = Password_Input.replace(" ", "")#removes spaces

```

Test Performed	Test Evidence	Pass Or Fail
DT28	 CHECK	Pass

DT29	 <p>PLEASE CLICK TO ENTER THE ACCESS KEY ADMIN56</p> <p>PLEASE CLICK TO ENTER YOUR PASSWORD HERE PASSCODETEST</p> <p>PLEASE CLICK TO ENTER YOUR USERNAME HERE TESTUSER321</p> <p>CHECK</p> <p>Accounts - Notepad</p> <p>File Edit Format View Help</p> <p>TESTUSER321 PASSCODETEST ADMIN56 0 0 0 TESTUSER PASS321 0 0 0 0 TESTUSER PASSWORD123 AD321 0 0 0</p>	Pass
------	---	------

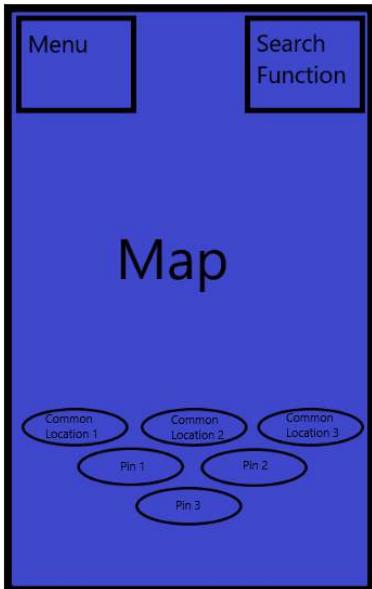
Sub Modules of Task1	Completed
A	Yes
B	Yes
C	Yes
D	Yes
E	Yes
F	Yes
G	Yes
H	No

Success Criteria	Achieved partially or fully or again	Explanation
SC6	Fully	As shown in DT27 the code is no capable of creating and storing authorised accounts shown by the fact that it stores the new account with an access key in the accounts file

SC7	Fully	As shown in DT28 the code is now capable of creating and storing accounts shown by the fact that it stores the new account in the accounts file
-----	-------	---

28/04/22:

Used Button class to code main screen shown in my screen design, although the pins are not there as the pin function is not coded in task1, so that the start function can now swap to this screen upon being pressed in order to fulfil **SC3** as displayed below:



```

def Main_Sc():
    while True:
        Screen.fill("black") #overlaps any previous screen to give an illusion of a new screen
        Screen.blit(MainMap, (0,0- (Height)/2))

        if LoggedIn == False:
            Start_Mouse_Pos = pygame.mouse.get_pos() #finds mouse position

        for Button in [Menu_Button, Search_Button]:# goes through each button and uploads it onto the screen
            Button.changeColor(Start_Mouse_Pos)
            Button.update(Screen)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
                if Menu_Button.checkForInput(Start_Mouse_Pos):
                    print("1")
                    Menu_Sc()

        pygame.display.update()
        clock.tick(60)
    
```



Test Performed	Test Evidence	Pass Or Fail
DT9	<p>START</p> <p>END</p> A screenshot from a navigation application showing a map of the SJWMS campus. The map includes a green field with a blue soccer field and a white track. Two points are marked: 'START' in yellow at the top left and 'END' in blue at the top right. A yellow icon with a target symbol is located near the bottom left corner of the map.	Pass

Success Criteria	Achieved partially or fully or again	Explanation
SC1	Fully	As shown in DT9 the start function is working and thus all buttons required for the start screen are working and therefore the start the screen is completed.
SC3	Again	Able to switch to main screen from starting screen through the use of the start function as shown in DT9

Sub Modules of Task1	Completed
A	Yes
B	Yes
C	Yes
D	Yes
E	Yes
F	Yes
G	Yes
H	Yes

Client Review:

After completing my first attempt of task 1 I sent an overview email to my client so that I can get feedback to make modifications and improvements to the product and thus following the iterative development as shown below:

"To Whom This May Concern,

I have currently completed the first task of the development of the application which consists of sub programs made in order to make a functioning starting screen. The starting screen itself consists of 3 buttons being the exit button , login button and start button, which respectively exits the program , switches the screen to a login screen allowing the user to sign in or swap to a screen that allows the user to make a new account and swaps the starting screen to the main screen, with screenshots provided below evidencing theses functions:

Start screen



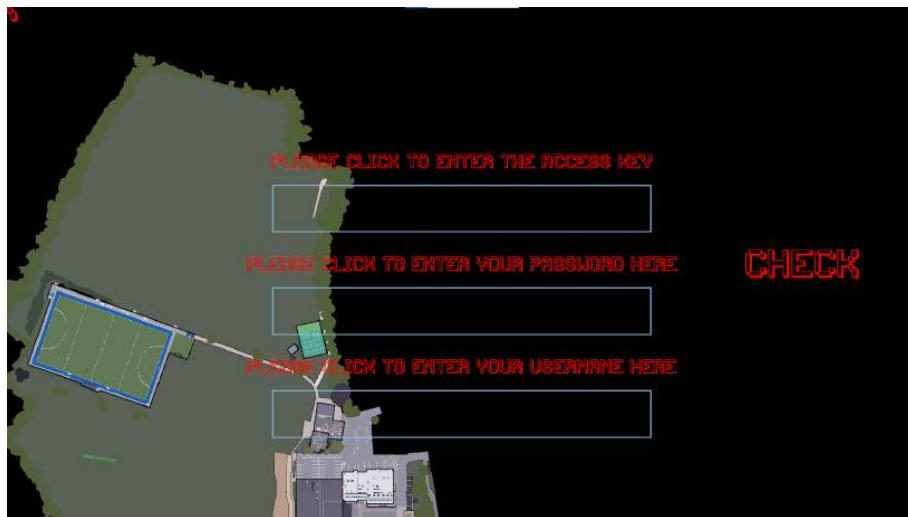
Main Screen



LogIn Screen



SignUp Screen



I would appreciate if you could suggest any potential additions

Yours Faithfully , Tireni Oluwabunmi”

Replies:

Joseph Baker: “Could you please the visibility of the buttons in the start screen as it’s rather hard to see with the background being black”

Mr Cooker: “I believe that you should either change the position of the menu button in the main screen or the map as the button hides part of the astroturf which is still part of the site.”

Review Modifications:

29/04/22:

From my client review I have been made aware that:

- A)the visibility of the buttons could be a problem to some so to fix this I will be adding a boxes behind the buttons
- B)the position of the map is inappropriate as the main menu button would be blocking part of the map so I'll be moving it more towards the centre

A: Simply used the button function to upload a box behind the image by adding an image within the provided parameters of the button class as shown below:

```
#Main Menu Buttons:
START_Button = Button|image = FilledRectangle ,pos = |Center_X ,Center_Y +|Height|/5| ,text_input = "START", font = get_font(|Height|/20), base_color = "#000317", hovering_color = "Yellow"
LOGIN_Button = Button|image = FilledRectangle, pos = |Center_X, Center_Y| ,text_input = "LOGIN", font = get_font(|Height|/20), base_color = "#000317", hovering_color = "Yellow"
EXIT_Button = Button|image = FilledRectangle, pos =(Center_X, Center_Y +|Height|/5) ,text_input = "Exit", font = get_font(|Height|/20), base_color = "#000317", hovering_color = "Yellow"
```



B: Screen blot the map at the ((center_x)/4 , 0) as by dividing the center_x variable ,which is the halfway point, by 4 we can consistently keep the menu button in the corner without the main screen overlapping it as shown below:

```
def Main_Sc():
    Screen.fill("black") #overlaps any previous screen to give an illusion of a new screen
    Screen.blit(MainMap, ((Center_X)/4,0))
    while True:
        if LoggedIn == False:
            Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

        for Button in [Menu_Button, Search_Button]:# goes through each button and uploads it onto the screen
            Button.changeColor(Start_Mouse_Pos)
            Button.update(Screen)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
                if Menu_Button.checkForInput(Start_Mouse_Pos):
                    Menu_Sc()
                    print("1")

        pygame.display.update()
        clock.tick(60)|
```



Task 2:

Development:

30/04/22:

Task2 only has 1 function being the search function but it still requires a screen for the user to be able to interact with it so firstly I needed to create and link the that screen to the main screen, I did so by using the search button created on the main screen to switch it to a screen I defined as the “LocationSearch_Sc” upon being pressed, further fulfilling **SC3**, as shown below

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()
    if Search_Button.checkForInput(Start_Mouse_Pos):#Switches user from main screen to the location search screen upon being clicked
        LocationSearch_Sc()
```

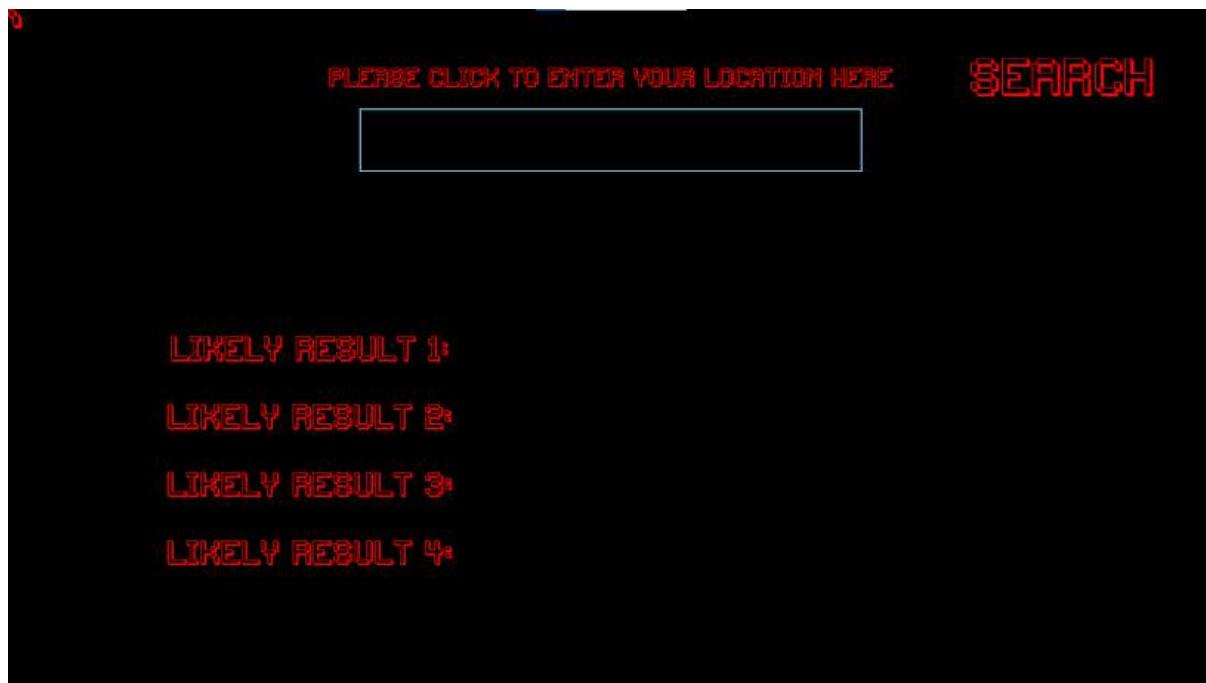


which would display the screen design below through the use of the button class

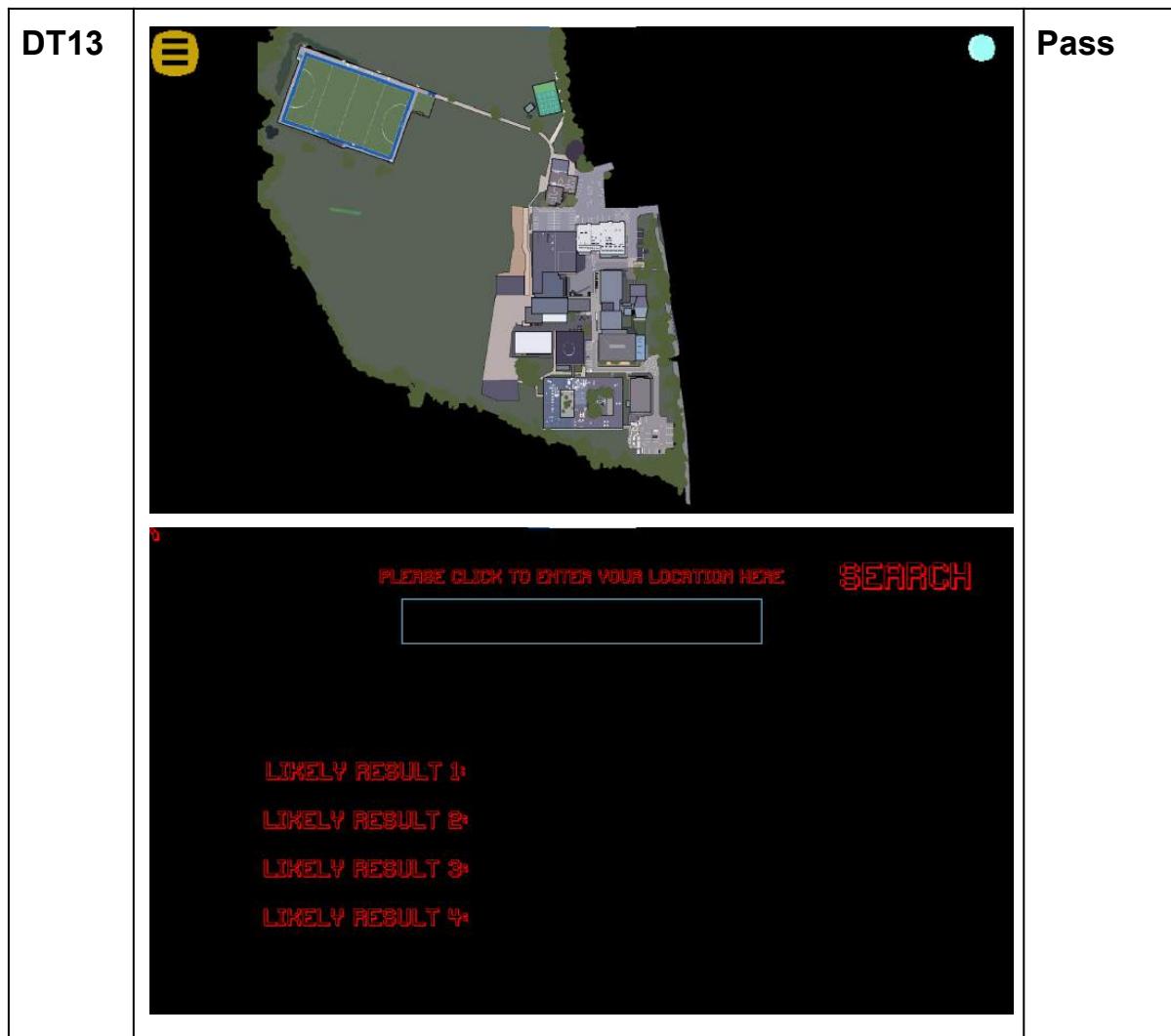


```
def LocationSearch_Sc():
    global Location_Input ,Location_Rect ,X_Button , Location_Type ,Result1_Button ,Result2_Button ,Result3_Button ,Result4_Button ,Result1_Input , Result2_Input,Result3_Input,Result4_Input
    while True:
        Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

        for Button in [X_Button, location_Button, Locate_Button, Result1_Button , Result2_Button , Result3_Button ,Result4_Button ]:# goes through each button and uploads it onto the screen
            Button.changeColor(Start_Mouse_Pos)
            Button.update(Screen)
```



Test Performed	Test Evidence	Pass Or Fail

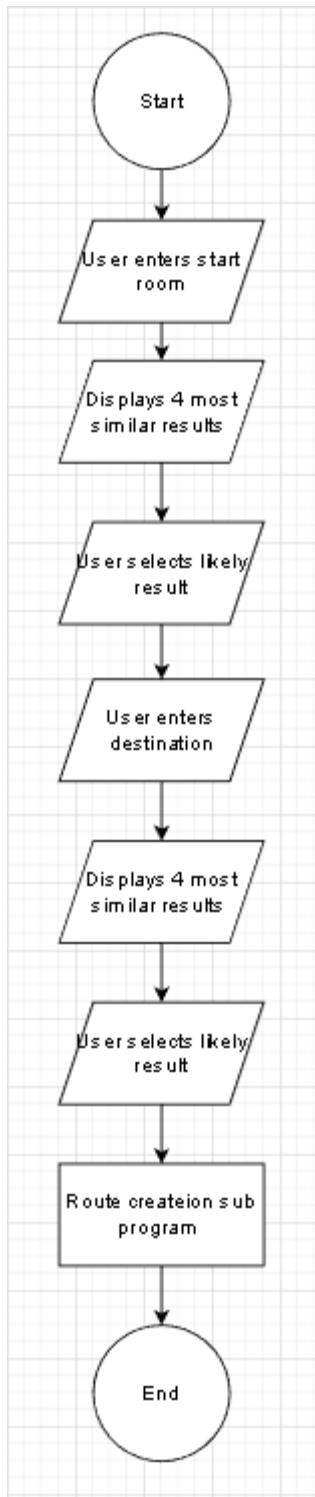


Sub Modules of Task2	Completed
A	Yes
B	No
C	No
D	No

Success Criteria	Achieved partially or fully or again	Explanation
SC3	Again	As show in the test DT13 the application is capable of switching screens if the button is pressed

01/05/22 -03/05/22:

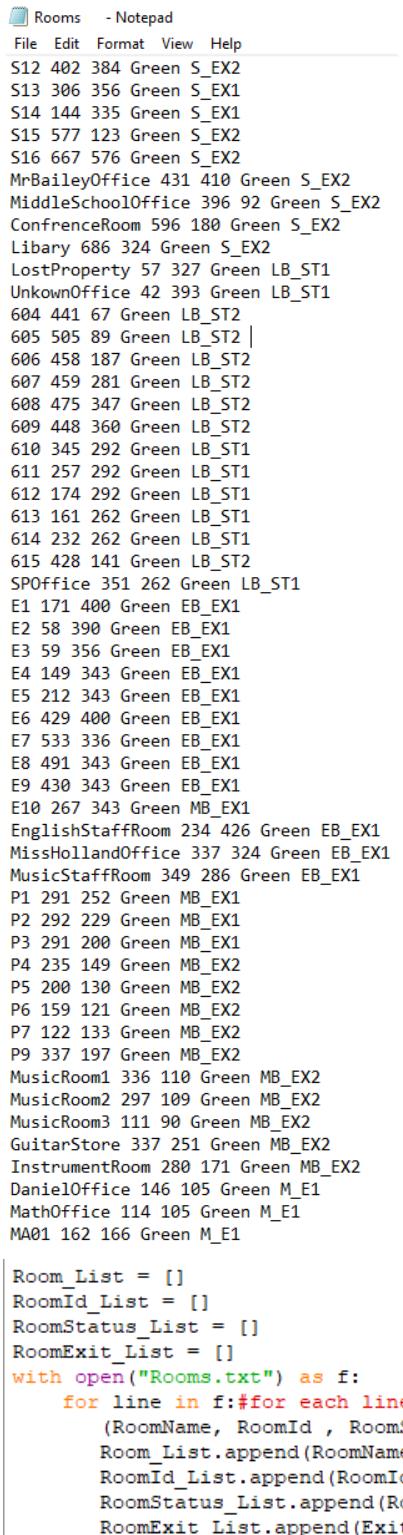
As shown in the search function's top down design below the function was initially supposed to check a dictionary in order to fulfil **SC8,SC9,SC11,SC14,SC15**



,but after researching further into how searching a dictionary would work in this scenario I decided against it as it became evident that using lists would be better. This is due to the fact that lists keep the order of elements while dictionaries do not, making it less efficient when trying to search up values from their link to another which is necessary in this scenario as the status and cords of the room must also be stored together with the room name.

As the data would be sorted into a list I was able to use the same way of loading the data on

the Accounts text file to a list with the Rooms text file by simply splitting the lines of the text file more to composite for the extra variables needed to be stored as shown in the screenshots below:



```

Rooms - Notepad
File Edit Format View Help
S12 402 384 Green S_EX2
S13 306 356 Green S_EX1
S14 144 335 Green S_EX1
S15 577 123 Green S_EX2
S16 667 576 Green S_EX2
MrBaileyOffice 431 410 Green S_EX2
MiddleSchoolOffice 396 92 Green S_EX2
ConfrenceRoom 596 180 Green S_EX2
Libary 686 324 Green S_EX2
LostProperty 57 327 Green LB_ST1
UnkownOffice 42 393 Green LB_ST1
604 441 67 Green LB_ST2
605 505 89 Green LB_ST2 |
606 458 187 Green LB_ST2
607 459 281 Green LB_ST2
608 475 347 Green LB_ST2
609 448 360 Green LB_ST2
610 345 292 Green LB_ST1
611 257 292 Green LB_ST1
612 174 292 Green LB_ST1
613 161 262 Green LB_ST1
614 232 262 Green LB_ST1
615 428 141 Green LB_ST2
SPOffice 351 262 Green LB_ST1
E1 171 400 Green EB_EX1
E2 58 390 Green EB_EX1
E3 59 356 Green EB_EX1
E4 149 343 Green EB_EX1
E5 212 343 Green EB_EX1
E6 429 400 Green EB_EX1
E7 533 336 Green EB_EX1
E8 491 343 Green EB_EX1
E9 430 343 Green EB_EX1
E10 267 343 Green MB_EX1
EnglishStaffRoom 234 426 Green EB_EX1
MissHollandOffice 337 324 Green EB_EX1
MusicStaffRoom 349 286 Green EB_EX1
P1 291 252 Green MB_EX1
P2 292 229 Green MB_EX1
P3 291 200 Green MB_EX1
P4 235 149 Green MB_EX2
P5 200 130 Green MB_EX2
P6 159 121 Green MB_EX2
P7 122 133 Green MB_EX2
P9 337 197 Green MB_EX2
MusicRoom1 336 110 Green MB_EX2
MusicRoom2 297 109 Green MB_EX2
MusicRoom3 111 90 Green MB_EX2
GuitarStore 337 251 Green MB_EX2
InstrumentRoom 280 171 Green MB_EX2
DanielOffice 146 105 Green M_E1
MathOffice 114 105 Green M_E1
MA01 162 166 Green M_E1

Room_List = []
RoomId_List = []
RoomStatus_List = []
RoomExit_List = []
with open("Rooms.txt") as f:
    for line in f:#for each line in the file assign the room data to the dictioanry accordingly
        (RoomName, RoomId , RoomStatus ,Exit) = line.split()
        Room_List.append(RoomName)
        RoomId_List.append(RoomId)
        RoomStatus_List.append(RoomStatus)
        RoomExit_List.append(Exit)

```

Afterwards I developed the search function that would display the inputted room if it existed

in the list ,but also made sure that the function took into consideration spelling errors and suggested the most likely input in order to achieve SC16, as shown below:

```
def Search(Place):
    global Result1_Input,Result2_Input ,Result3_Input ,Result4_Input,
    for i in range(len(Place)-1):#repeats for the number of characters within the input
        for x in range(len(Room_List)):
            #repeats for the number of rooms in total to compare each room
    #with the input reducing the number of characters each time to see which ones are most similar and then storing them in results
        if Place[0:-i] == (Room_List[x])[0:-i]:
            Result1_Input = Room_List[x]
        if Place[0:-i] == (Room_List[x])[0:-i]:
            Result2_Input = Room_List[x]
        if Place[0:-i] == (Room_List[x])[0:-i]:
            Result3_Input = Room_List[x]
        if Place[0:-i] == (Room_List[x])[0:-i]:
            Result4_Input = Room_List[x]
```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>513</p> <p>LIKELY RESULT 1</p> <p>LIKELY RESULT 2</p> <p>LIKELY RESULT 3</p> <p>LIKELY RESULT 4</p>	Fail
DT2	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>51</p> <p>LIKELY RESULT 1</p> <p>LIKELY RESULT 2</p> <p>LIKELY RESULT 3</p> <p>LIKELY RESULT 4</p>	Fail

Sub Modules of Task2	Completed
A	Yes
B	No
C	No
D	No

05/05/22:

As shown in the screenshot provided when completing test X the search function failed to output the likely results expected. This was due to the fact that Result_Inputs were made equivalent ,==, and not equal to as shown below:

```
def Search(Place):
    global Result1_Input,Result2_Input ,Result3_Input ,Result4_Input,
    for i in range(len(Place)-1):#repeats for the number of characters within the input
        for x in range(len(Room_List)):
            #repeats for the number of rooms in total to compare each room with the input reducing the number
            #of characters each time to see which ones are most similar and then storing them in results
            if Place[0:-i] == (Room_List[x])[0:-i]:
                Result1_Input = Room_List[x]
            if Place[0:-i] == (Room_List[x])[0:-i]:
                Result2_Input = Room_List[x]
            if Place[0:-i] == (Room_List[x])[0:-i]:
                Result3_Input = Room_List[x]
            if Place[0:-i] == (Room_List[x])[0:-i]:
                Result4_Input = Room_List[x]
```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>LIKELY RESULT 1: 812</p> <p>LIKELY RESULT 2: 812</p> <p>LIKELY RESULT 3: 812</p> <p>LIKELY RESULT 4: 812</p>	Fail

DT2	 <p>The screenshot shows a search interface with a text input field containing 's12'. To the right of the input field is a 'SEARCH' button. Below the input field, there are four entries labeled 'LIKELY RESULT 1', 'LIKELY RESULT 2', 'LIKELY RESULT 3', and 'LIKELY RESULT 4', each followed by 's12'.</p>	Fail
-----	---	------

Sub Modules of Task2	Completed
A	Yes
B	No
C	No
D	No

07/05/22:

Although the search function now provides an output it's incorrect. This is due to the fact that the text file loaded did not contain all the values needed so all I needed to do was add all the

required rooms as shown below

 Rooms - Notepad
File Edit Format View Help

```

S1 68 450 Green S_EX1
S2 68 334 Green S_EX1
S4 66 107 Green S_EX1
S5 111 61 Green S_EX1
S6 179 95 Green S_EX1
S7 274 93 Green S_EX2
S8 353 92 Green S_EX2
S9 356 158 Green S_EX2
S10 517 335 Green S_EX2
S11 356 256 Green S_EX2
S12 402 384 Green S_EX2
S13 306 356 Green S_EX1
S14 144 335 Green S_EX1
S15 577 123 Green S_EX2
S16 667 576 Green S_EX2
604 441 67 Green LB_ST2
605 505 89 Green LB_ST2
606 458 187 Green LB_ST2
607 459 281 Green LB_ST2
608 475 347 Green LB_ST2
609 448 360 Green LB_ST2
610 345 292 Green LB_ST1
611 257 292 Green LB_ST1
612 174 292 Green LB_ST1
613 161 262 Green LB_ST1
614 232 262 Green LB_ST1
615 428 141 Green LB_ST2
E1 171 400 Green EB_EX1
E2 58 390 Green EB_EX1
E3 59 356 Green EB_EX1
E4 149 343 Green EB_EX1
E5 212 343 Green EB_EX1
E6 429 400 Green EB_EX1
E7 533 336 Green EB_EX1
E8 491 343 Green EB_EX1
E9 430 343 Green EB_EX1
E10 267 343 Green EB_EX1
P1 291 252 Green EB_EX1
P2 292 229 Green EB_EX1
P3 291 200 Green EB_EX1
P4 235 149 Green EB_EX1
P5 200 130 Green EB_EX1
P6 159 121 Green EB_EX1
P7 122 133 Green EB_EX1
P9 337 197 Green EB_EX1
MA01 162 166 Green M_E1
MA02 162 228 Green M_EX1
MA03 82 165 Green M_E1
MA04 82 302 Green M_EX1
MA05 82 343 Green M_EX1
MA11 158 480 Green M_ST1
MA12 158 422 Green M_ST1
MA13 158 164 Green M_ST2
MA11 158 480 Green M_ST1
MA12 158 422 Green M_ST1
MA13 158 164 Green M_ST2
MA14 331 85 Green M_ST2
MA15 330 216 Green M_ST2
MA16 330 419 Green M_ST1
11 59 123 Green C_ST2
12 157 107 Green C_ST2
13 261 108 Green C_ST2
14 261 224 Green C_ST2
15 111 236 Green C_ST2
16 89 211 Green C_ST2
21 31 137 Green G_ST2
22 121 120 Green G_ST2
23 233 123 Green G_ST2
24 232 232 Green G_ST2
25 81 250 Green G_ST2
26 61 225 Green G_ST2
DT1 206 178 Green DT_E1
DT2 206 124 Green DT_E1
DT3 510 189 Green DT_EX1
DT4 492 145 Green DT_ST2
DT5 491 303 Green DT_ST2
SEN 510 103 Green DT_EX1
PrepRoom 684 59 Green DT_EX1
TechRoom 713 93 Green DT_EX1
DTLab 684 188 Green DT_EX1
A1 103 257 Green DT_ST2
A2 100 190 Green DT_ST2
MA13 158 164 Green M_ST2

```

,but even with not all necessary values being stored it made evident the fact that the likely result outputs the same values even though in this instance it should have outputted S12 S13 S14 as they would be the next 3 likely results.

I came to the conclusion that this was due to the fact that the search function allowed the values to be stored repeatedly so I added a parameter that the result must be empty for a value to be stored as shown below

```

---+-----+
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result1_Input == "":
        Result1_Input = Room_List[x]
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result2_Input == "":
        Result2_Input = Room_List[x]
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result3_Input == "":
        Result3_Input = Room_List[x]
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result4_Input == "":
        Result4_Input = Room_List[x]

```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>A screenshot of a search interface. At the top, there is a placeholder text "PLEASE CLICK TO ENTER YOUR LOCATION HERE". Below it is a search bar containing the text "S13". To the right of the search bar is a red "SEARCH" button. The main area displays four identical results, each consisting of the text "LIKELY RESULT" followed by a number and a location name: "LIKELY RESULT 1: S12", "LIKELY RESULT 2: S12", "LIKELY RESULT 3: S12", and "LIKELY RESULT 4: S12".</p>	Fail
DT2	 <p>A screenshot of a search interface. At the top, there is a placeholder text "PLEASE CLICK TO ENTER YOUR LOCATION HERE". Below it is a search bar containing the text "S1". To the right of the search bar is a red "SEARCH" button. The main area displays four identical results, each consisting of the text "LIKELY RESULT" followed by a number and a location name: "LIKELY RESULT 1: S12", "LIKELY RESULT 2: S12", "LIKELY RESULT 3: S12", and "LIKELY RESULT 4: S12".</p>	Fail

Sub Modules of Task2	Completed
----------------------	-----------

A	Yes
B	No
C	No
D	No

09/10/22:

The program still outputted repeated values so to fix this I added a parameter that disables a value from being stored if it's already been stored to a likely input as shown in the code below:

```
if Place[0:-i] == (Room_List[x])[0:-i] and Result1_Input == "":
    Result1_Input = Room_List[x]
    RepeatResult = True
if Place[0:-i] == (Room_List[x])[0:-i] and Result2_Input == "" and RepeatResult == False:
    Result2_Input = Room_List[x]
    RepeatResult = True
if Place[0:-i] == (Room_List[x])[0:-i] and Result3_Input == "" and RepeatResult == False:
    Result3_Input = Room_List[x]
    RepeatResult = True
if Place[0:-i] == (Room_List[x])[0:-i] and Result4_Input == "" and RepeatResult == False:
    Result4_Input = Room_List[x]
    RepeatResult = True
RepeatResult = False
```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>S13</p> <p>SEARCH</p> <p>LIKELY RESULT 1: S12</p> <p>LIKELY RESULT 2: S13</p> <p>LIKELY RESULT 3: S14</p> <p>LIKELY RESULT 4: S15</p>	Fail

DT2	<p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>S1</p> <p>SEARCH</p> <p>LIKELY RESULT 1: S1</p> <p>LIKELY RESULT 2:</p> <p>LIKELY RESULT 3:</p> <p>LIKELY RESULT 4:</p>	Pass
------------	--	-------------

Sub Modules of Task2	Completed
A	Yes
B	No
C	No
D	No

Success Criteria	Achieved partially or fully or again	Explanation
SC8	Partially	Code is able to successfully find single digit rooms
SC9	Partially	Code is able to successfully find single digit rooms

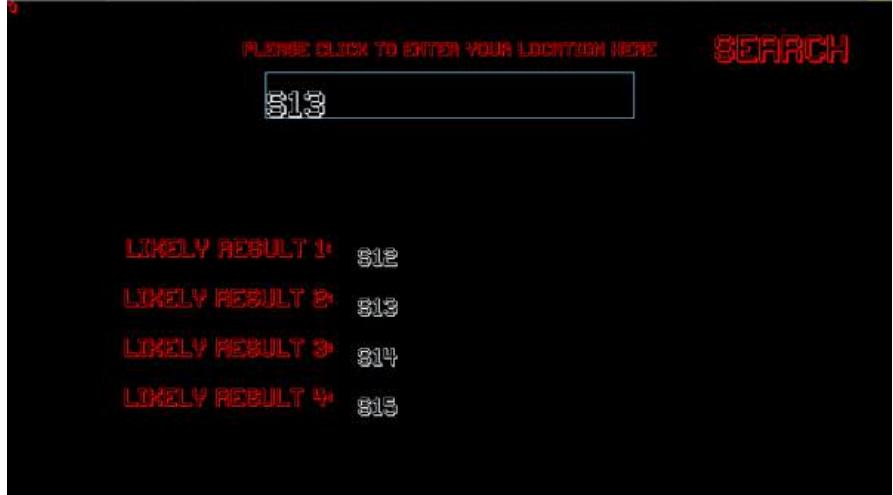
10/05/22:

The code produces the likely results in incorrect order as S12 is above S13 even though it should be the other way around. Tried to fix by increasing number of loops by 1 so that the first loop directly compares the values as shown below:

```

for x in range(len(Room_List)):
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result1_Input == "":
        Result1_Input = Room_List[x]
        RepeatResult = True
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result2_Input == "" and RepeatResult == False:
        Result2_Input = Room_List[x]
        RepeatResult = True
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result3_Input == "" and RepeatResult == False:
        Result3_Input = Room_List[x]
        RepeatResult = True
    if Place[0:-i+1] == (Room_List[x])[0:-i+1] and Result4_Input == "" and RepeatResult == False:
        Result4_Input = Room_List[x]
        RepeatResult = True
RepeatResult = False

```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>LIKELY RESULT 1: s12</p> <p>LIKELY RESULT 2: s13</p> <p>LIKELY RESULT 3: s14</p> <p>LIKELY RESULT 4: s15</p>	Fail
DT2	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>LIKELY RESULT 1: s1</p> <p>LIKELY RESULT 2:</p> <p>LIKELY RESULT 3:</p> <p>LIKELY RESULT 4:</p>	Pass

Sub Modules of Task2	Completed
A	Yes
B	No

C	No
D	No

11/05/22:

Stills produces the code in the incorrect order which I releases is probably due to the loop not directly comparing the inputted value and just assuming that it's incorrect so I made sure that the loop directly compares the inputted value without it being shortened as shown in the code below:

```
def Search(Place):
    global Result1_Input,Result2_Input ,Result3_Input ,Result4_Input, RepeatResult
    for i in range(len(Place)):
        for x in range(len(Room_List)):
            if Place == Room_List[x]:
                Result1_Input = Room_List[x]
                RepeatResult = True
            if Place[0:-i] == (Room_List[x])[0:-i] and Result1_Input == "":
                Result1_Input = Room_List[x]
                RepeatResult = True
            if Place[0:-i] == (Room_List[x])[0:-i] and Result2_Input == "" and RepeatResult == False:
                Result2_Input = Room_List[x]
                RepeatResult = True
            if Place[0:-i] == (Room_List[x])[0:-i] and Result3_Input == "" and RepeatResult == False:
                Result3_Input = Room_List[x]
                RepeatResult = True
            if Place[0:-i] == (Room_List[x])[0:-i] and Result4_Input == "" and RepeatResult == False:
                Result4_Input = Room_List[x]
                RepeatResult = True
    RepeatResult = False
```

Test Performed	Test Evidence	Pass Or Fail
DT1	 <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>LIKELY RESULT 1: S13</p> <p>LIKELY RESULT 2: S10</p> <p>LIKELY RESULT 3: S11</p> <p>LIKELY RESULT 4: S12</p>	Fail

DT2	<p>The screenshot shows a search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE". To the right of the input field is a button labeled "SEARCH". Below the input field, the text "S1" is displayed in a box. Underneath this, there are four lines of text labeled "LIKELY RESULT 1", "LIKELY RESULT 2", "LIKELY RESULT 3", and "LIKELY RESULT 4", each followed by a short description.</p>	Fail
------------	--	-------------

Sub Modules of Task2	Completed
A	Yes
B	Yes
C	No
D	No

12/05/22

I now had to make a function that displays the status of the likely result rooms which I did by checking the stored status of the room in the room text file through the use of the Room List and displayed the corresponding image as shown below:

```
def RoomStatusFinder(Room):
    if RoomStatus_List[int(Room_List.index(Room))] == "Red":
        return RedStatus_Image
    if RoomStatus_List[int(Room_List.index(Room))] == "Yellow":
        return YellowStatus_Image
    if RoomStatus_List[int(Room_List.index(Room))] == "Green":
        return GreenStatus_Image
```

I then implemented the function into the Search function so that the search function is now also capable of finding the status of the possible rooms as show below

```
R1StatusDispaly = RoomStatusFinder(Result1_Input)
R2StatusDispaly = RoomStatusFinder(Result2_Input)
R3StatusDispaly = RoomStatusFinder(Result3_Input)
R4StatusDispaly = RoomStatusFinder(Result4_Input)
```

Now all that was left was to display it on the search screen which can be made possible through the use of pygame's screen.blit function in the search screens' functions as shown

below

```
Screen.blit(R1StatusDisplay, (Center_X,Center_Y))
Screen.blit(R2StatusDisplay, (Center_X,Center_Y +(Height)/10))
Screen.blit(R3StatusDisplay, (Center_X,Center_Y +(Height)/5))
Screen.blit(R4StatusDisplay, (Center_X,Center_Y +(Height)/10 +(Height)/5))
```

Although I also added a new function called ClearSearchData as it became evident to me that there were wasted lines from repetition of clearing the data in the search screens so I replaced them with the single line of ClearSearchData as shown below

```
def ClearSearchData():#resets search data for new room to be searched
    global Result1_Input,Result2_Input ,Result3_Input ,Result4_Input,R1StatusDisplay,R2StatusDisplay,R3StatusDisplay,R4StatusDisplay,Location_Input
    R1StatusDisplay = RedStatus_Image
    R2StatusDisplay = RedStatus_Image
    R3StatusDisplay = RedStatus_Image
    R4StatusDisplay = RedStatus_Image
    Result1_Input = ""
    Result2_Input = ""
    Result3_Input = ""
    Result4_Input = ""
    Location_Input = ""

    Search(Location_Input)
    if Result1_Button.checkForInput(Start_Mouse_Pos):#selects Result 1 as the starting position
        ActualLocation = Result1_Input
        if ActualLocation == "":
            ClearSearchData()#resets search data for new room to be searched
            False_Sc3()
        else:
            ClearSearchData()#resets search data for new room to be searched
            DestinationSearch_Sc()
    if Result2_Button.checkForInput(Start_Mouse_Pos):#selects Result 2 as the starting position
        ActualLocation = Result2_Input
        if ActualLocation == "":
            ClearSearchData()#resets search data for new room to be searched
            False_Sc3()
        else:
            ClearSearchData()#resets search data for new room to be searched
            DestinationSearch_Sc()
    if Result3_Button.checkForInput(Start_Mouse_Pos):#selects Result 3 as the starting position
        ActualLocation = Result3_Input
        if ActualLocation == "":
            ClearSearchData()#resets search data for new room to be searched
            False_Sc3()
        else:
            ClearSearchData()#resets search data for new room to be searched
            DestinationSearch_Sc()
    if Result4_Button.checkForInput(Start_Mouse_Pos):#selects Result 4 as the starting position
        ActualLocation = Result4_Input
        if ActualLocation == "":
            ClearSearchData()#resets search data for new room to be searched
            False_Sc3()
        else:
            ClearSearchData()#resets search data for new room to be searched
            DestinationSearch_Sc()
    if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
        ClearSearchData()#resets search data for new room to be searched
```

Test Performed	Test Evidence	Pass Or Fail

DT1	 A screenshot of the SJWMS search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE" and a button labeled "SEARCH". Below the input field, the text "S13" is entered. Underneath, a list of "LIKELY RESULT"s is displayed: "LIKELY RESULT 1: S13" with a green dot, "LIKELY RESULT 2: S10" with a green dot, "LIKELY RESULT 3: S11" with a green dot, and "LIKELY RESULT 4: S12" with a green dot.	Pass
DT2	 A screenshot of the SJWMS search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE" and a button labeled "SEARCH". Below the input field, the text "S1" is entered. Underneath, a list of "LIKELY RESULT"s is displayed: "LIKELY RESULT 1:" with a red dot, "LIKELY RESULT 2:" with a red dot, "LIKELY RESULT 3:" with a red dot, and "LIKELY RESULT 4:" with a red dot. <pre>pygame 2.1.2 (SDL 2.0.18, Python 3.10.7) Hello from the pygame community. https://www.pygame.org/contribute.html Traceback (most recent call last): File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 2598, in <module> Start_Sc() File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 1047, in Start_Sc LogIn_Sc() File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 1093, in LogIn_Sc Main_Sc() File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 452, in Main_Sc Menu_Sc() File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 524, in Menu_Sc StatusSearch_Sc() File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 964, in StatusSearch_Sc R1statusDisplay = RoomStatusFinder(Result1_Input) File "C:\Users\Tireni\Downloads\SjwmsMaps Finished\Final SJWMS.py", line 2250, in RoomStatusFinder if RoomStatus_List[int(Room_List.index(Room))] == "Red": ValueError: '' is not in list</pre>	Fail

Sub Modules of Task2	Completed
A	Yes
B	Yes
C	No
D	No

Success Criteria	Achieved partially or fully or again	Explanation
SC8	Partially	Code is able to successfully find any room that isn't a single digit as shown in DT1.
SC9	Partially	Code is able to successfully find any room that isn't a single digit as shown in DT1.
SC12	Partially	Code is able to successfully find the status of any room that aren't single digit as shown in DT1
SC16	Again	Code able to accomplish the likely result validation of suggest possible corrections when a room is searched for as shown in DT1.

13/05/22

The error that occurred in DT2 on the 12/05/22 was due to me not taking into consideration the empty results that occur when you search for a singular digit room and thus the code does not make any possible corrections so to fix this I made the RoomStatusFinder take into consideration empty inputs by displaying red

```
def RoomStatusFinder(Room):
    if Room == '':
        return RedStatus_Image
    else:
        if RoomStatus_List[int(Room_List.index(Room))] == "Red":
            return RedStatus_Image
        if RoomStatus_List[int(Room_List.index(Room))] == "Yellow":
            return YellowStatus_Image
        if RoomStatus_List[int(Room_List.index(Room))] == "Green":
            return GreenStatus_Image
```

Test Performed	Test Evidence	Pass Or Fail

DT2	<p>The screenshot shows a search interface with a text input field containing 'S1'. Above the input field is a placeholder text 'PLEASE CLICK TO ENTER YOUR LOCATION HERE'. To the right of the input field is a button labeled 'SEARCH'. Below the input field, the text 'LIKELY RESULT 1: S1' is followed by a green dot. Below it, 'LIKELY RESULT 2:' is followed by a red dot. Below that, 'LIKELY RESULT 3:' is followed by a red dot. At the bottom, 'LIKELY RESULT 4:' is followed by a red dot.</p>	Pass
------------	---	-------------

Sub Modules of Task2	Completed
A	Yes
B	Yes
C	No
D	No

Success Criteria	Achieved partially or fully or again	Explanation
SC8	Fully	Code is able to successfully find any room
SC9	Fully	Code is able to successfully find any room
SC12	Fully	Code is able to successfully find the status of any room
SC16	Again	Code able to suggest possible corrections when a room is searched for.

14/05/22:

I then made it so the result that was clicked was stored as the user's actual location to be used later in the path visualizer to be developed in task 4 and reset all other results to null to reduce the chance of an error.

```
if Locate_Button.checkForInput(Start_Mouse_Pos):#Searches the room list for inputted value and values simular to it
    Search(Location_Input)
if Result1_Button.checkForInput(Start_Mouse_Pos):#selects Result 1 as the starting position
    ActualLocation = Result1_Input
    if ActualLocation == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc3()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        DestinationSearch_Sc()
if Result2_Button.checkForInput(Start_Mouse_Pos):#selects Result 2 as the starting position
    ActualLocation = Result2_Input
    if ActualLocation == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc3()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        DestinationSearch_Sc()
if Result3_Button.checkForInput(Start_Mouse_Pos):#selects Result 3 as the starting position
    ActualLocation = Result3_Input
    if ActualLocation == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc3()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        DestinationSearch_Sc()
if Result4_Button.checkForInput(Start_Mouse_Pos):#selects Result 4 as the starting position
    ActualLocation = Result4_Input
    if ActualLocation == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc3()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        DestinationSearch_Sc()
```

All that was left was the destination search screen which in reality is just the same as the location search screen but instead it stores its results in a different variable being the users actual destination and goes to the path visualiser screen developed in task4 as shown

below:

```

if Locate_Button.checkForInput(Start_Mouse_Pos):#Searches the room list for inputted value and values simular to it
    Search(Destination_Input)
if Result1_Button.checkForInput(Start_Mouse_Pos):#selects Result 1 as the ActualDestination
    ActualDestination = Result1_Input
    if ActualDestination == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc4()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        Path1_Sc()
if Result2_Button.checkForInput(Start_Mouse_Pos):#selects Result 2 as the ActualDestination
    ActualDestination = Result2_Input
    if ActualDestination == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc4()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        Path1_Sc()
if Result3_Button.checkForInput(Start_Mouse_Pos):#selects Result 3 as the ActualDestination
    ActualDestination = Result3_Input
    if ActualDestination == "":
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        False_Sc4()
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        Path1_Sc()
if Result4_Button.checkForInput(Start_Mouse_Pos):#selects Result 4 as the ActualDestination
    ActualDestination = Result4_Input
    if ActualDestination == "":
        False_Sc4()
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
    else:
        Result1_Input = ""
        Result2_Input = ""
        Result3_Input = ""
        Result4_Input = ""
        Path1_Sc()

```

Sub Modules of Task2	Completed
A	Yes
B	Yes
C	Yes
D	Yes

Client Review:

After completing task 2 to the best of my ability I sent an overview email to my client so that I can once again get feedback to make improvements to the product and thus following the iterative development as shown below:

"To Whom This May Concern,

I have now finished the first second task of the development of the application which consists of a singular sub-program made in order to search for rooms within the site and the screens it will be used on.

Upon clicking the search icon in the top right hand corner of the mains screen the screen switches to the 1st search screen designed for you to search for a location, while also displaying the searched room status, and then confirm the location by picking a result which will then swap the screen to the 2nd search screen designed for you to search for a destination and then confirm the it by picking a result as shown in the screenshots provided.

Search Icon:



Location Search Screen:



Destination Search Screen:



I would appreciate if you could suggest any potential additions

Yours Faithfully , Tireni Oluwabunmi"

Replies:

Joseph Baker: "The search icon is rather hard to see since it blends in with the black background could you potentially make it more clear"

Mr Cooker: "Could you change the colour of the search icon so it's easier to see. "

Alvin Roy: "I can't locate the search function could you possibly highlight it in some sort of way"

Review Modifications:

From my client review I have been made aware that:

A)The search icon is not noticeable and hard to see

A: Simply changed the colour of the search icon so it no longer blends in with the black background and now stands out as it directly contrasts it with the icon now being white.



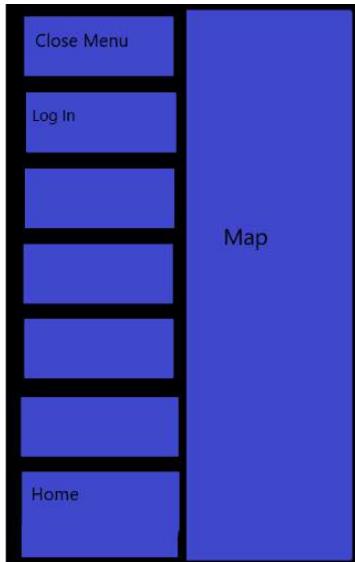
Task 3:

Development:

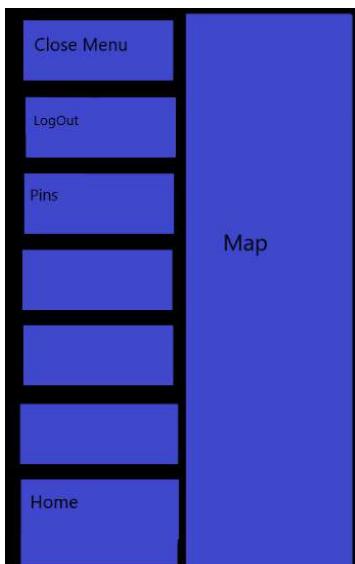
16/05/22:

Similar to the start of the search function development the main menu icon button is already present all I need to do is code it so that upon it being clicked it opens the menu in order to fulfil **SC3**, although the code also needs to open the correct menu by checking if the user is logged in and the level of the accounts authorization to see which type of menu to open as there are 3 types displayed from the screen designs below from design:

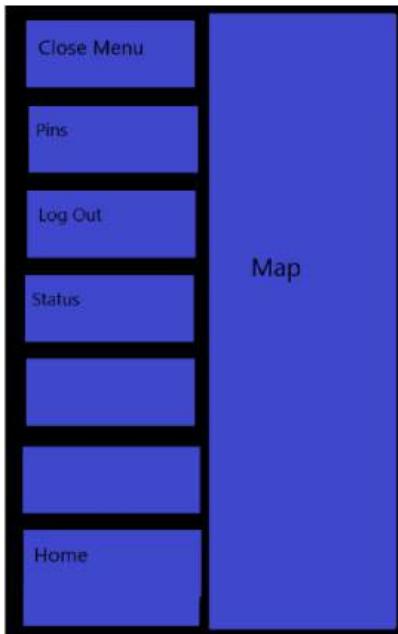
Not logged in menu



Non Admin menu



Admin Menu



This was achieved by simply using if statements to check the nature of the account loaded in order to decide what buttons to be drawn onto the screen

```
def Menu_Sc():
    global Username_Input, Password_Input, Username_Rect, Password_Rect, Createaccount_Button, X_Button, Check_Button, Username_Type, Password_Type, LoggedIn, UserAccessKey, UserAccount, UserPic
    while True:
        if LoggedIn == True:#checks too see if user is logged in
            if str(UserAccessKey) == "ADMIN56":#checks to see if account is an admin
                Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

                for event in pygame.event.get():
                    if event.type == pygame.QUIT:
                        pygame.quit()
                        sys.exit()

                pygame.display.update()
                clock.tick(60)

                Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
                Screen.blit(BackgroundMap_Img, ((Center_X)/4, 0 - (Height)/2))
                Screen.blit(SideMenu_Image, (0,0))
            else:#if account is not admin
                Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

                for event in pygame.event.get():
                    if event.type == pygame.QUIT:
                        pygame.quit()
                        sys.exit()
                    pygame.display.update()
                    clock.tick(60)

                Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
                Screen.blit(BackgroundMap_Img, ((Center_X)/4, 0 - (Height)/2))
                Screen.blit(SideMenu_Image, (0,0))

        elif LoggedIn == False:#if user is not logged in
            Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    pygame.quit()
                    sys.exit()

            pygame.display.update()
            clock.tick(60)

        Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
        Screen.blit(BackgroundMap_Img, ((Center_X)/4, 0 - (Height)/2))
        Screen.blit(SideMenu_Image, (0,0))
```

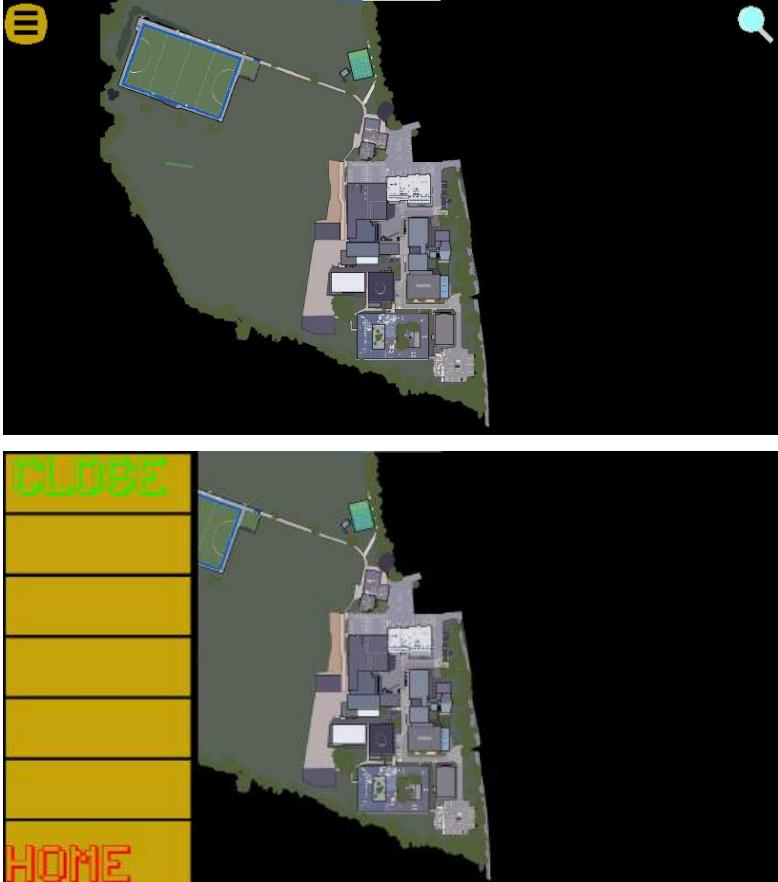
I then decided to code the functions and buttons in the order of how common they were as some functions and buttons are used repeatedly in different menu variations as shown in the screen designs and thus the first buttons I coded were the close menu and home button first as they are used in all variations of the menu.

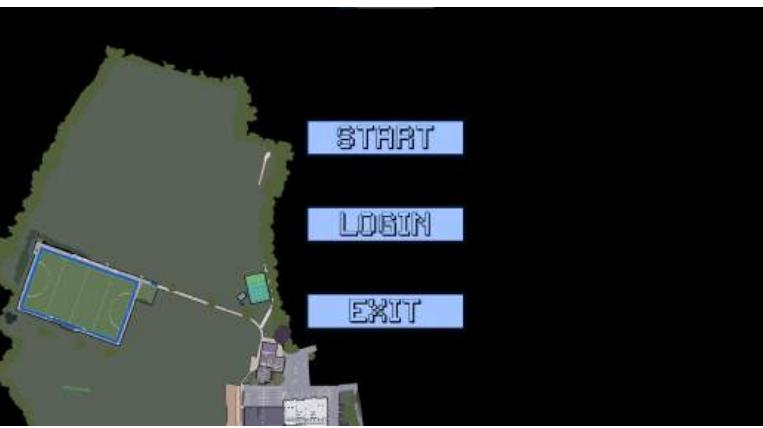
The close menu button simply had to go back to the main screen

```
if Close_Button.checkForInput(Start_Mouse_Pos):#if clicked on goes back to main screen
    Main_Sc()
```

And the home button simply had to go back to the start screen

```
if Home_Button.checkForInput(Start_Mouse_Pos):#if clicked on goes back to start| screen
    Start_Sc()
```

Test Performed	Test Evidence	Pass Or Fail
DT14		Fail

DT17	 	Pass
DT18	 	Pass

Sub Modules of Task3	Completed
A	Yes
B	No
C	No
D	No
E	No
F	No
G	No
H	No

Success Criteria	Achieved partially or fully or again	Explanation
SC3	Again	The close button swaps the screen from the menu screen to the main screen and the home button swaps the screen to the start screen from the menu screen.

17/05/22:

The only other button left for the logged out menu to be complete was the login button which simply referred back to the login screen in similar fashion to the close menu and home button ,in order to further fulfil **SC3** and **SC4**, as shown below

```
if LogIn_Button.checkForInput(Start_Mouse_Pos)::#if clicked on goes to log in screen
LogIn_Sc()
```

With the logged out menu being done I continued my development by progressing into the non admin menu which only requires me to switch the login button with logout button in order to fulfil **SC5** and add form the pin function and button in order to fulfil **SC14** and **SC15**.

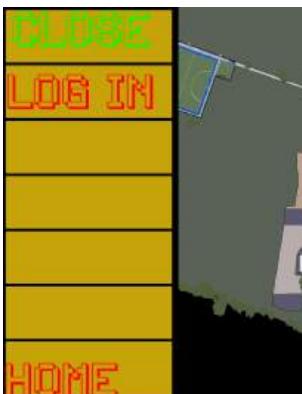
To complete the logout button I simply cleared the signed in account data and made the

logged in variable false upon the button being clicked.

```
if LogOff_Button.checkForInput(Start_Mouse_Pos):  
    UserAccount = ""  
    UserAccessKey = ""  
    UserPins = []  
    LoggedIn = False  
    Start_Sc()
```

Test Performed	Test Evidence	Pass Or Fail
DT14		Pass

DT21	 	Pass
DT15	 	Pass

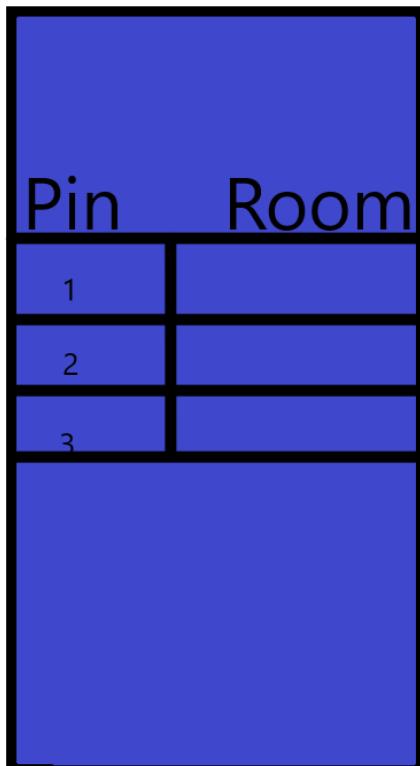
DT22	  	Pass
-------------	--	-------------

Sub Modules of Task3	Completed
A	Yes
B	Yes
C	Yes
D	Yes
E	Yes
F	No
G	No

Success Criteria	Achieved partially or fully or again	Explanation
SC3	Again	<p>The menu button swaps from the main screen to the logged out menu screen if clicked and the user is not logged in.</p> <p>The log out button swaps from the menu screen to the start screen.</p> <p>The login button swaps from the menu screen to the login screen.</p>
SC4	Again	<p>The login button allows the user to log into an account by referring them to the login screen</p>
SC5	Fully	<p>The log out button clears the user's account details leaving them as a non logged in account.</p>

18/05/22:

The logged in menu only required one more function and button to be completed being the pin function so that **SC14** and **SC15** are fulfilled. In order to create the pin function I firstly had to create a screen to display the already saved pins as shown in the screen design designed in design below:



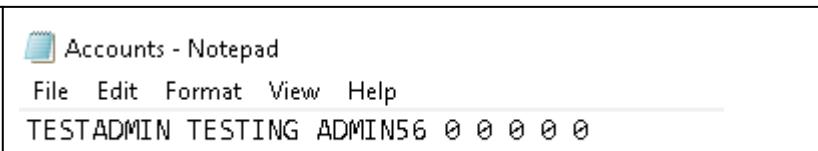
I then made it possible for the pins to be changed by making the pins link to a search screen that reused the location search screen in order for the user to be able to search up another room with the search function with the results buttons now linking back to the pin screen and storing the result picked in that pin.

```
if Pin1Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin1 based on the room searched for
    Pin1Searching = True
    PinSearch_Sc()
if Pin2Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin2 based on the room searched for
    Pin2Searching = True
    PinSearch_Sc()
if Pin3Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin3 based on the room searched for
    Pin3Searching = True
    PinSearch_Sc()
```

I then created buttons for the pin to be displayed upon in the main screen through the use of the button function

```
Pin1_Button = Button(image = Pin1_Image ,pos = (60 ,(Height)/6) , text_input = s)
Pin2_Button = Button(image = Pin2_Image ,pos = (60, 5*(Height)/24) , text_input = s)
Pin3_Button = Button(image = Pin3_Image ,pos = (60,(Height)/4) , text_input = st)
```
def Main_Sc():
 global Menu_Button, Search_Button ,LoggedIn
 while True:
 if LoggedIn == True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 for Button in [Menu_Button, Search_Button, Pin1_Button, Pin2_Button, Pin3_Button]:# goes through each button and sends its info to the screen
```

| Test Performed | Test Evidence | Pass Or Fail |
|----------------|---------------|--------------|
|----------------|---------------|--------------|

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>DT3</b> |  <p>Accounts - Notepad<br/>File Edit Format View Help<br/>TESTADMIN TESTING ADMIN56 0 0 0 0 0 0</p>  <p>PINS ROOMS</p> <p>1<br/>2<br/>3</p> <p>0 TO 0<br/>0 TO 0<br/>0 TO 0</p> <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>S12</p>  <p>LIKELY RESULT 1: S12 •<br/>LIKELY RESULT 2: S10 •<br/>LIKELY RESULT 3: S11 •<br/>LIKELY RESULT 4: S13 •</p> <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>E2</p>  <p>LIKELY RESULT 1: E2 •<br/>LIKELY RESULT 2: •<br/>LIKELY RESULT 3: •<br/>LIKELY RESULT 4: •</p> <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>E2</p> | <b>Fail</b> |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|



|             |  |             |
|-------------|--|-------------|
| <b>DT19</b> |  | <b>Pass</b> |
|-------------|--|-------------|

| <b>Sub Modules of Task3</b> | <b>Completed</b> |
|-----------------------------|------------------|
| A                           | Yes              |
| B                           | Yes              |
| C                           | Yes              |
| D                           | Yes              |
| E                           | Yes              |
| F                           | No               |
| G                           | No               |

| <b>Success Criteria</b> | <b>Achieved partially or fully or again</b> | <b>Explanation</b>                                                               |
|-------------------------|---------------------------------------------|----------------------------------------------------------------------------------|
| SC3                     | Again                                       | Allows user to swap to the status search screen upon a room option being touched |

|      |           |                                                                                                        |
|------|-----------|--------------------------------------------------------------------------------------------------------|
| SC14 | Partially | Allows user to access the search screen in order to change the room that is saved to the selected pin. |
|------|-----------|--------------------------------------------------------------------------------------------------------|

**19/05/22:**

Although the pin function could now change the pins of a logged in user it needed to be capable of storing the new data so I made a new sub function called store in order to store the data that was changed so that **SC14** can be fulfilled properly .

I did so by rewriting the list that was stored and then clearing the Accounts text file and uploading the new Accounts list in order to update it.

```
def Store_Pins(Username):
 global UserPins, Pins_List
 Pins_List.insert(3*int(User_List.index(Username)),UserPins[0])
 Pins_List.insert(3*int(User_List.index(Username))+1,UserPins[1])
 Pins_List.insert(3*int(User_List.index(Username))+2,UserPins[2])
 Pins_List.insert(3*int(User_List.index(Username))+3,UserPins[3])
 Pins_List.insert(3*int(User_List.index(Username))+4,UserPins[4])
 Pins_List.insert(3*int(User_List.index(Username))+5,UserPins[5])
```

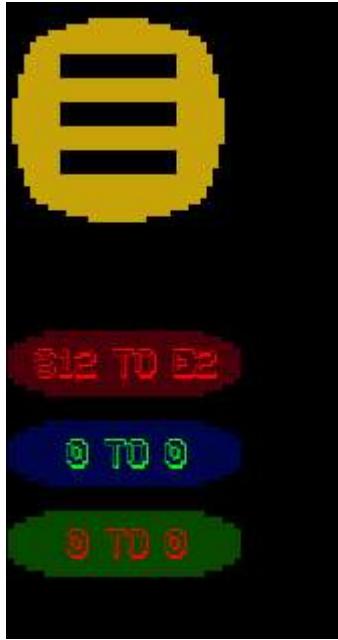
I also made it so the code suggest to the user to fill in a pin if an empty pin is selected

```
 LocationSearch.Sc()
 if Pin1_Button.checkForInput(Start_Mouse_Pos):
 if UserPins[0] == "0" and UserPins[1] == "1":#if the pin is empty it goes to a false screen to correct the user accordingly
 False_Sc5()
 else:#Inputs pin into path finding algorithim
 StartLocation = UserPins[0]
 EndLocation = UserPins[1]
 Path1_Sc()
 if Pin2_Button.checkForInput(Start_Mouse_Pos):
 if UserPins[2] == "0" and UserPins[3] == "1":#if the pin is empty it goes to a false screen to correct the user accordingly
 False_Sc5()
 else:#Inputs pin into path finding algorithim
 StartLocation = UserPins[2]
 EndLocation = UserPins[3]
 Path1_Sc()
 if Pin3_Button.checkForInput(Start_Mouse_Pos):
 if UserPins[4] == "0" and UserPins[5] == "1":#if the pin is empty it goes to a false screen to correct the user accordingly
 False_Sc5()
 else:#Inputs pin into path finding algorithim
 StartLocation = UserPins[4]
 EndLocation = UserPins[5]
 Path1_Sc()
```

| Test Performed | Test Evidence | Pass Or Fail |
|----------------|---------------|--------------|
|----------------|---------------|--------------|

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |      |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| DT3 | <p>Accounts - Notepad</p> <p>File Edit Format View Help</p> <p>TESTADMIN TESTING ADMIN56 0 0 0 0 0</p>  <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>S12</p> <p>LIKELY RESULT 1: S12 •</p> <p>LIKELY RESULT 2: S10 •</p> <p>LIKELY RESULT 3: S11 •</p> <p>LIKELY RESULT 4: S13 •</p>  <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>E2</p> <p>LIKELY RESULT 1: E2 •</p> <p>LIKELY RESULT 2: •</p> <p>LIKELY RESULT 3: •</p> <p>LIKELY RESULT 4: •</p>  | Pass |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|



|      |                                                                                   |      |
|------|-----------------------------------------------------------------------------------|------|
| DT12 |  | Pass |
|------|-----------------------------------------------------------------------------------|------|

| Sub Modules of Task3 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | Yes       |
| D                    | Yes       |
| E                    | Yes       |
| F                    | Yes       |
| G                    | No        |

| <b>Success Criteria</b> | <b>Achieved partially or fully or again</b> | <b>Explanation</b>                                                                                                                                                                      |
|-------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SC14                    | Again                                       | Store Pins allows users pins to now be saved to their account                                                                                                                           |
| SC15                    | Fully                                       | Store Pins allows users pins to now be saved to their account and displays it on the pin screen                                                                                         |
| SC16                    | Fully                                       | As shown in DT12 the program is able to contain the <b>Pin Error Screen Validation</b> as clicking an empty pin will cause the code to suggest to the user to fill the pin with a value |

**20/05/22:**

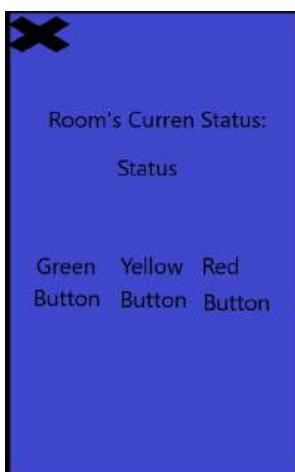
With the pin button now being completed the logged in screen was completed and now all that was left was the status button for the admin menu in order for **SC12** and **SC13** to be fulfilled.

For the status button to be completed I reused the location search screen from task 2 again in order to use the search function to search for a specific room

```
if Status_Button.checkForInput(Start_Mouse_Pos):
 StatusSearch_Sc()
```

but instead now the

with the results now linkin to a new page defined as the status screen which displays the current status of the room and the options to change it as detailed in the screen designs below



I used the button function to display the green , yellow and red buttons in the status screen and change the status to the button's colour upon it being clicked as shown below

```
#Status Edit Screen Buttons:
RedStatus_Button = Button(image = RedStatus_Image ,pos = (Center_X +(Width)/8,Center_Y) , text_input = "", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")

YellowStatus_Button = Button(image = YellowStatus_Image ,pos = (Center_X ,Center_Y) , text_input = "", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")

GreenStatus_Button = Button(image = GreenStatus_Image ,pos = (Center_X -(Width)/8,Center_Y) , text_input = "", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")
"""

Main_Sc()
if RedStatus_Button.checkForInput(Start_Mouse_Pos):#if red status button is pressed then status of the room becomes red
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Red")
 Main_Sc()
if YellowStatus_Button.checkForInput(Start_Mouse_Pos):#if yellow status button is pressed then status of the room becomes yellow
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Yellow")
 Main_Sc()
if GreenStatus_Button.checkForInput(Start_Mouse_Pos):#if green status button is pressed then status of the green becomes red
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Green")
 Main_Sc()
```

I then reused the method of clearing a text file and storing a changed list in it to update from the store accounts function in order to be able store the new status of rooms in the room text file with the storeroom function

```
def Store_Pins(Username):
 global UserPins, Pins_List
 Pins_List.insert(3*int(User_List.index(Username)),UserPins[0])#Stores the start location of the 1st pin
 Pins_List.insert(3*int(User_List.index(Username))+1,UserPins[1])#Stores the end location of the 1st pin
 Pins_List.insert(3*int(User_List.index(Username))+2,UserPins[2])#Stores the start location of the 2nd pin
 Pins_List.insert(3*int(User_List.index(Username))+3,UserPins[3])#Stores the end location of the 2nd pin
 Pins_List.insert(3*int(User_List.index(Username))+4,UserPins[4])#Stores the start location of the 3rd pin
 Pins_List.insert(3*int(User_List.index(Username))+5,UserPins[5])#Stores the end location of the 3rd pin
```

| Test Performed | Test Evidence | Pass Or Fail |
|----------------|---------------|--------------|
|----------------|---------------|--------------|

|      |                                                                                                                                                                                                                                                                   |      |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| DT16 | <br><br> | Pass |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|

|            |                                                                                                                                                                                                                                                                                            |             |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>DT8</b> | S2 68 334 Green S_EX1<br><br><br> | <b>Pass</b> |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|

|             |  |             |
|-------------|--|-------------|
| <b>DT20</b> |  | <b>Pass</b> |
|-------------|--|-------------|

| <b>Sub Modules of Task3</b> | <b>Completed</b> |
|-----------------------------|------------------|
| A                           | Yes              |
| B                           | Yes              |
| C                           | Yes              |
| D                           | Yes              |
| E                           | Yes              |
| F                           | Yes              |
| G                           | Yes              |

| <b>Success Criteria</b> | <b>Achieved partially or fully or again</b> | <b>Explanation</b>                                                             |
|-------------------------|---------------------------------------------|--------------------------------------------------------------------------------|
| SC3                     | Again                                       | In DT20 it shows that clicking the status button will switch the user's screen |

|      |       |                                                                                                                                                                                  |
|------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      |       | from the menu screen to the status search screen and in DT16 it shows that clicking the menu button while logged into an admin account will transfer you to an admin level menu. |
| SC12 | Fully | Within DT8 it's shown how the user can see the status of the room s2 fulfilling the success criteria                                                                             |
| SC13 | Fully | In DT8 it's shown that an admin level account is capable of changing the status of the room s2 fulfilling the success criteria                                                   |

### **Client Review:**

"To Whom This May Concern,

I am contacting you as I have now completed the third task of the development of the application which consists of sub programs made in order to make a functioning menu. There are 3 menus in order to correspond to the 3 levels of authorization an account may have being none if the user isn't logged in



,base level if the user is logged in but does not have an admin key



,admin level if the user's account also has an admin key



The close menu, home and log in link back to the main screen, start screen and log in screen respectively, but the status and pins buttons go to a search screen with the status button allowing an admin to change the status of the searched room as shown below





And the pins function allowing the path from a searched starting point to an end starting point to be stored in a pin as shown below





I would appreciate if you could suggest any potential additions

Yours Faithfully , Tireni Oluwabunmi"

#### Replies:

**Joseph Baker:** "This looks alright to me. I have no suggestions currently."

**Mr Cooker:** " I would suggest adding the level of authorization in one of the unused menu boxes to clearly indicate to the user their level of authorization of their account . "

**Alivn Roy:**"This looks great I have no suggestions to be added"

#### Review Modifications:

From my client review I have been made aware that:

A)It would be optimal if the menu displayed the level of authorization of the user

A: To accomplish this I simply screen blited the word admin in white in the side menu ,in the instance that the user signed into an admin, and screenlited the username of the signed in account ,in the instance that the user signed into a non admin account, through the use of the already made if statement to check the level of authorization as shown below:

```

def Menu_Set():
 global Username_Input, Password_Input, Username_Rect ,Password_Rect , Createaccount_Button ,I_Button ,Check_Button , Username_Type , Password_Type, LoggedIn,UserAccessKey,UserAccount, UserPins
 while True:
 if LoggedIn == True:
 if str(UserAccessKey) == "ADMINKEY":
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 AdminIdentifier = get_font((Height)/11.5).render("Admin",True,(240,248,255))#Admin's font in order to be screenslited to screen so that the program can verify to the User that they've been signed into an Admin account

 else:Normal account
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 SignedIdentifier = get_font((Height)/11.5).render(UserAccount,True,(240,248,255))#User's Username in font in order to be screenslited to screen so that the program can verify to them that they've been signed in

```



(Username = Test)



## Task 4:

### Development:

21/05/22:

Task 4's main purpose is to implement the use of the A\* path finder in order to accomplish **SC10 SC8** and **SC9** which I think can be best achieved through the use of CocoXiao1998's path visualizer function as although the intended purpose was to click on two positions on a blank map and then draw the path between them I found that by replacing the single parameter of Image with Colorued\_Img and Blank\_Img I could use the Blank\_Img in order to create the path with the find\_path\_alg function ,as the code was only compatible with grayscale images,

```
A* alg
def find_path_alg(Blank_Img,Colorued_Img, width, height, starting_position, ending_position, path_color):
 # creating a list to hold the f, g, and h-values and the parent coords for each node
 # 5 for f, g, h values and parent x, y values
 node_details = np.full((width, height, 5), -1)

 # create a 2D-list that marks all nodes that have been evaluated by 1, 0 if not
 closed_list = np.zeros((width, height))

 # creating an open list that will contains nodes with calculated f costs
 # each index will contain: [f, [x, y]]
 open_list = []

 # putting the starting position in open list with its f as 0
 open_list.append([0, [starting_position[0], starting_position[1]]])

 while len(open_list) != 0:
 # set current equal to node with the lowest f-value
 if len(open_list) == 1:
 current = open_list[0]
 else:
 current = open_list[0]
 for node in open_list:
 node_x = node[1][0]
 node_y = node[1][1]
 cur_x = current[1][0]
 cur_y = current[1][1]

 if node[0] < current[0]:
 current = node
 # if f-values are equal
 elif node[0] == current[0]:
 # setting current equal to node if the h-value is lower
 node_h = node_details[node_x][node_y][2]
 current_h = node_details[cur_x][cur_y][2]
 if node_h < current_h:
 current = node

 # current node coords vars
 cur_x = current[1][0]
 cur_y = current[1][1]

 # removing current node from open list
 open_list.remove(current)
```

```

removing current node from open list
open_list.remove(current)

marking node as 1 in closed list
closed_list[cur_x][cur_y] = 1

checking if current node is the destination
if is_destination(Blank_Img, cur_x, cur_y, ending_position):
 print_path(Colorued_Img, starting_position, ending_position, path_color, node_details)
 return

getting the neighbor nodes
else:
 neighbor_list = [(cur_x - 1, cur_y), (cur_x + 1, cur_y), (cur_x, cur_y + 1), (cur_x, cur_y - 1),
 (cur_x - 1, cur_y + 1), (cur_x - 1, cur_y - 1), (cur_x + 1, cur_y + 1), (cur_x + 1, cur_y - 1)]
 direction_counter = 0

 for neighbor in neighbor_list:
 # neighbor node x and y vars
 n_x = neighbor[0]
 n_y = neighbor[1]

 # check that neighbor exists, is traversable and is marked as 0 in closed list
 if (exists(Blank_Img, n_x, n_y, width, height) and
 is_traversable(Blank_Img, n_x, n_y) and
 closed_list[n_x][n_y] == 0):
 # checking if current node is starting position in order to get right value g
 if current[1] == starting_position:
 if direction_counter <= 3:
 g = 10
 else:
 g = 14
 else:
 if direction_counter <= 3:
 g = node_details[cur_x][cur_y][1] + 10
 else:
 g = node_details[cur_x][cur_y][1] + 14

 # getting other values
 h = heuristics(img, n_x, n_y, ending_position)
 f = g + h

 # checking if neighbor needs to be updated
 # checking if f-cost to neighbor is shorter or if neighbor not in open list
 update = True
 in_open = False
 index = -1
 count = 0
 for node in open_list:
 node_f = node[0]
 node_x = node[1][0]
 node_y = node[1][1]
 if n_x == node_x and n_y == node_y:
 in_open = True
 index = count
 if f >= node_f:
 update = False
 count += 1

 update = False
 count += 1

 # update node_details and updating open list
 if update == True:
 # storing details in node details
 node_details[n_x][n_y][0] = f
 node_details[n_x][n_y][1] = g
 node_details[n_x][n_y][2] = h
 node_details[n_x][n_y][3] = cur_x
 node_details[n_x][n_y][4] = cur_y

 # if neighbor is not in open, then append neighbor, else just update f-value
 if in_open == False:
 open_list.append(([f, [n_x, n_y]]))
 else:
 open_list[index][0] = f

 direction_counter += 1

```

and then use the draw\_path function to display the path on the Colorued\_Img

```

def print_path(Colorued_Img, starting_position, ending_position, path_color, node_details):
 endpoint_color = (255, 0, 0)
 start_x = starting_position[0]
 start_y = starting_position[1]
 end_x = ending_position[0]
 end_y = ending_position[1]
 parent_x = node_details[end_x][end_y][3]
 parent_y = node_details[end_x][end_y][4]
 Colorued_Img[parent_y][parent_x] = path_color#draws path onto coloured image

 # will loop until the parent x and y are the starting position
 while True:
 if parent_x == start_x and parent_y == start_y:
 Colorued_Img[start_y][start_x] = endpoint_color
 Colorued_Img[end_y][end_x] = endpoint_color
 return

 # getting the parent coords and then setting the coords to the path color
 temp_x = parent_x
 temp_y = parent_y
 parent_x = node_details[temp_x][temp_y][3]
 parent_y = node_details[temp_x][temp_y][4]
 Colorued_Img[parent_y][parent_x] = path_color

```

enabling me to be able to create a path to display to the user.

## 23/05/22

With the path finder now being re-coded I now just had to create functions in order to implement data into the function efficiently.

To do so first I needed to actually load the data required for such functions to work as although the Rooms text file had the coords for the rooms themselves they did not have the building that they were inside so in order to identify building that they were in so that a path from building to building and from the room to the building's exit could be made I stored the exits of the buildings in separate variables

```

#Outside Nodes
ScienceBlock_Nodes = ["S_EX1", "S_EX2"]
LanguageBlock_Nodes = ["LB_ST1", "LB_ST2"]
EnglishAndMusicBlock_Nodes = ["EB_EX1", "MB_EX1", "MB_EX2",]
MathsBlockDownstairs_Nodes = ["M_E1", "M_EX1"]
MathsBlockUpstairs_Nodes = ["M_ST1", "M_ST2"]
ComputingBlock_Nodes = ["C_ST1", "C_ST2"]
GeographyBlock_Nodes = ["G_ST1", "G_ST2"]
DTBlockDownstairs_Nodes = ["DT_E1", "DT_EX1"]
DTBlockUpstairs_Nodes = ["DT_ST1", "DT_ST2"]
SFC_EX1_Nodes = ["S_EX1", "S_EX2", "LB_ST1", "EB_EX1", "M_E1", "M_EX1", "M_ST1", "M_ST2", "C_ST1", "C_ST2", "G_ST1", "G_ST2", "DT_ST1", "DT_ST2"]
SFC_EX2_Nodes = ["DT_EX1", "MB_EX1", "MB_EX2"]
SFC_EX3_Nodes = ["LB_ST2"]
SixthFormCenter_Nodes = [SFC_EX1_Nodes, SFC_EX2_Nodes, SFC_EX3_Nodes]
Stair_Nodes = ["LB_ST1", "LB_ST2", "M_ST1", "M_ST2", "DT_ST1", "DT_ST2", "G_ST1", "G_ST2", "C_ST1", "C_ST2"]

```

as the Rooms Text File did store the exit that the room was closeted to allowing the building of the room to now be identified later, although I realised that I had to separate the rooms near stairs as they did not immediately go out the building and would also need to add a new stage, F, to the checklist for the breakdown of task 4 which would ensure "Unique paths" have been individually coded.

I then loaded the grayscale, coloured maps as jpeg in order to be compatible with CocoXiao1998's path visualizer function as it uses cvs2 making sure to have 2 coloured maps as 1 would be changing and the other would be needed in order to change the changing map back to its original state.

```

-----Data-----##
#Grayscale Backgrounds:
MainMap_G = cv2.imread("MainMapBlank.jpg")
ComputingBlock_G = cv2.imread("ComputingBlockBlank.jpg")
GeographyBlock_G = cv2.imread("GeographyBlockBlank.jpg")
ScienceBlock_G = cv2.imread("ScienceBlockBlank.jpg")
MathsBlockDownstairs_G = cv2.imread("MathsDownstairsBlank.jpg")
MathsBlockUpstairs_G = cv2.imread("MathsUpstairsBlank.jpg")
DTBlockDownstairs_G = cv2.imread("DtBlockDownstairsBlank.jpg")
DTBlockUpstairs_G = cv2.imread("DtBlockUpstairsBlank.jpg")
EnglishBlock_G = cv2.imread("EnglishBlockBlank.jpg")
LanguageBlock_G = cv2.imread("LanguageBlockBlank.jpg")
SixthFormCenter_G = cv2.imread("SixthFormCenterBlank.jpg")
SixthFormStairsA_G = cv2.imread("SixthFormStairsABlank.jpg")
SixthFormStairsB_G = cv2.imread("SixthFormStairsBBlank.jpg")
LanguageStairsFromA_G = cv2.imread("LanguageStairsFromABlank.jpg")
LanguageStairsFromB_G = cv2.imread("LanguageStairsFromBBlank.jpg")
DTStairsFrom_G = cv2.imread("DTStairsFromBlank.jpg")
MathsStairsFrom_G = cv2.imread("MathStairsBlank.jpg")
LanguageStairsToA_G = cv2.imread("LanguageStairsToABlank.jpg")
LanguageStairsToB_G = cv2.imread("LanguageStairsToBBlank.jpg")
DTStairsToA_G = cv2.imread("DTBlockToABlank.jpg")
DTStairsToB_G = cv2.imread("DTBlockToBBlank.jpg")
MathsStairsTo_G = cv2.imread("MathStairsBlank.jpg")
GeographyStairsTo_G = cv2.imread("GeographyStairsToBlank.jpg")
GeographyStairsFrom_G = cv2.imread("GeographyStairsFromBlank.jpg")
MainBlockFromA_G = cv2.imread("MainBlockFromABlank.jpg")
MainBlockFromB_G = cv2.imread("MainBlockFromBBlank.jpg")
MainBlockToA_G = cv2.imread("MainBlockToABlank.jpg")
MainBlockToB_G = cv2.imread("MainBlockToBBlank.jpg")

#PermaBackgrounds:
MainMap_P = cv2.imread("MainMapColour.jpg")
ComputingBlock_P = cv2.imread("ComputingBlockColor.jpg")
GeographyBlock_G = cv2.imread("GeographyBlockColor.jpg")
ScienceBlock_P = cv2.imread("ScienceBlockColour.jpg")
MathsBlockDownstairs_P = cv2.imread("MathsDownstairsColor.jpg")
MathsBlockUpstairs_P = cv2.imread("MathsUpstairsColor.jpg")
DTBlockDownstairs_P = cv2.imread("DtBlockDownstairsColor.jpg")
DTBlockUpstairs_P = cv2.imread("DtBlockUpstairsColor.jpg")
EnglishBlock_P = cv2.imread("EnglishBlockColor.jpg")
LanguageBlock_P = cv2.imread("LanguageBlockColour.jpg")
SixthFormCenter_P = cv2.imread("SixthFormCenterColor.jpg")
SixthFormStairsA_P = cv2.imread("SixthFormStairsAColor.jpg")
SixthFormStairsB_P = cv2.imread("SixthFormStairsBColor.jpg")
LanguageStairsFromA_P = cv2.imread("LanguageStairsFromAColor.jpg")
LanguageStairsFromB_P = cv2.imread("LanguageStairsFromBColor.jpg")
DTStairsFrom_P = cv2.imread("DTStairsFromColor.jpg")
MathsStairsFrom_P = cv2.imread("MathStairsColor.jpg")
LanguageStairsToA_P = cv2.imread("LanguageStairsToAColor.jpg")
LanguageStairsToB_P = cv2.imread("LanguageStairsToBColor.jpg")
DTStairsToA_P = cv2.imread("DTBlockToAColor.jpg")
DTStairsToB_P = cv2.imread("DTBlockToBColor.jpg")
MathsStairsTo_P = cv2.imread("MathStairsColor.jpg")
GeographyStairsTo_P = cv2.imread("GeographyStairsToColor.jpg")
GeographyStairsFrom_P = cv2.imread("GeographyStairsFromColor.jpg")
MainBlockFromA_P = cv2.imread("MainBlockFromAColor.jpg")
MainBlockFromB_P = cv2.imread("MainBlockFromBColor.jpg")
MainBlockToA_P = cv2.imread("MainBlockToAColor.jpg")

```

```
#ChaningBackgrounds:
MainMap_C = cv2.imread("MainMapChainging.jpg")
ComputingBlock_C = cv2.imread("ComputingBlockChainging.jpg")
GeographyBlock_C = cv2.imread("GeographyBlockChainging.jpg")
ScienceBlock_C = cv2.imread("ScienceBlockChainging.jpg")
MathsBlockDownstairs_C = cv2.imread("MathsDownstairsChainging.jpg")
MathsBlockUpstairs_C = cv2.imread("MathsUpstairsChainging.jpg")
DTBlockDownstairs_C = cv2.imread("DtBlockDownstairsChainging.jpg")
DTBlockUpstairs_C = cv2.imread("DtBlockUpstairsChainging.jpg")
EnglishBlock_C = cv2.imread("EnglishBlockChainging.jpg")
LanguageBlock_C = cv2.imread("LanguageBlockChainging.jpg")
SixthFormCenter_C = cv2.imread("SixthFormCenterChainging.jpg")
SixthFormStairsA_C = cv2.imread("SixthFormStairsAChainging.jpg")
SixthFormStairsB_C = cv2.imread("SixthFormStairsBChainging.jpg")
LanguageStiarsFromA_C = cv2.imread("LanguageStiarsFromAChainging.jpg")
LanguageStiarsFromB_C = cv2.imread("LanguageStiarsFromBChainging.jpg")
DTStiarsFrom_C = cv2.imread("DTStiarsFromChainging.jpg")
MathsStairsFrom_C = cv2.imread("MathStairsChainging.jpg")
LanguageStiarsToA_C = cv2.imread("LanguageStiarsToAChainging.jpg")
LanguageStiarsToB_C = cv2.imread("LanguageStiarsToBChainging.jpg")
DTStiarsToA_C = cv2.imread("DTBlockToAChainging.jpg")
DTStiarsToB_C = cv2.imread("DTBlockToBChainging.jpg")
MathsStairsTo_C = cv2.imread("MathStairsChainging.jpg")
GeographyStairsTo_C = cv2.imread("GeographyStairsToChainging.jpg")
GeographyStairsFrom_C = cv2.imread("GeographyStiarsFromChainging.jpg")
MainBlockFromA_C = cv2.imread("MainBlockFromAChainging.jpg")
MainBlockFromB_C = cv2.imread("MainBlockFromBChainging.jpg")
MainBlockToA_C = cv2.imread("MainBlockToAChainging.jpg")
MainBlockToB_C = cv2.imread("MainBlockToBChainging.jpg")
```

I now needed to a function to identify the map to be used so I create the IdentifyMap function which took a parameter of an exit and checks to see if it's within any of the buildings exists list and if so returns the map data as shown below

```
def IdentityMap(Exit):
 for i in range(len(ScienceBlock_Nodes)):
 if ScienceBlock_Nodes[i] == str(Exit):
 return [ScienceBlock_G, ScienceBlock_C, ScienceBlock_P, ScienceBlock_Img, "ScienceBlockChainging.jpg"]

 for i in range(len(LanguageBlock_Nodes)):
 if LanguageBlock_Nodes[i] == str(Exit):
 return [LanguageBlock_G, LanguageBlock_C, LanguageBlock_P, LanguageBlock_Img, "LanguageBlockChainging.jpg"]

 for i in range(len(EnglishAndMusicBlock_Nodes)):
 if EnglishAndMusicBlock_Nodes[i] == str(Exit):
 return [EnglishBlock_G, EnglishBlock_C, EnglishBlock_P, EnglishBlock_Img, "English BlockChainging.jpg"]

 for i in range(len(MathsBlockDownstairs_Nodes)):
 if MathsBlockDownstairs_Nodes[i] == str(Exit):
 return [MathsBlockDownstairs_G, MathsBlockDownstairs_C, MathsBlockDownstairs_P, MathsBlockDownstairs_Img, "MathsDownstairsChainging.jpg"]

 for i in range(len(MathsBlockUpstairs_Nodes)):
 if MathsBlockUpstairs_Nodes[i] == str(Exit):
 return [MathsBlockUpstairs_G, MathsBlockUpstairs_C, MathsBlockUpstairs_P, MathsBlockUpstairs_Img, "MathsUpstairsChainging.jpg"]

 for i in range(len(ComputingBlock_Nodes)):
 if ComputingBlock_Nodes[i] == str(Exit):
 return [ComputingBlock_G, ComputingBlock_C, ComputingBlock_P, ComputingBlock_Img, "ComputingBlockChainging.jpg"]

 for i in range(len(GeographyBlock_Nodes)):
 if GeographyBlock_Nodes[i] == str(Exit):
 return [GeographyBlock_G, GeographyBlock_C, GeographyBlock_P, GeographyBlock_Img, "GeographyBlockChainging.jpg"]

 for i in range(len(DTBlockDownstairs_Nodes)):
 if DTBlockDownstairs_Nodes[i] == str(Exit):
 return [DTBlockDownstairs_G, DTBlockDownstairs_C, DTBlockDownstairs_P, DTBlockDownstairs_Img, "DtBlockDownstairsChainging.jpg"]

 for i in range(len(DTBlockUpstairs_Nodes)):
 if DTBlockUpstairs_Nodes[i] == str(Exit):
 return [DTBlockUpstairs_G, DTBlockUpstairs_C, DTBlockUpstairs_P, DTBlockUpstairs_Img, "DtBlockUpstairsChainging.jpg"]
```

|                             |                  |
|-----------------------------|------------------|
| <b>Sub Modules of Task4</b> | <b>Completed</b> |
|-----------------------------|------------------|

|   |     |
|---|-----|
| A | Yes |
| B | No  |
| C | No  |
| D | No  |
| E | No  |
| F | No  |

**25/05/22:**

Although the data for the map is now loaded the coordinates of the nodes on the map have not so I stored the coordinates within the text file MainNodes and used the same method of loading and storing data of the Account and Room text files as shown below

```
NodeName_List = []
NodeX_List = []
NodeY_List = []
with open("MainNodes.txt") as f:
 for line in f:#for each line in the file assign the node data to the dictioanry accordingly
 (NodeName, Node_X , Node_Y) = line.split()
 NodeName_List.append(NodeName)
 NodeX_List.append(Node_X)
 NodeY_List.append(Node_Y)
```

I then needed to be able to get retrieve the co ords for the room's exit based on the room itself so I created the IdentifyNode function which identified the coords for the rooms exit by comparing the value of the room's exit stored against the node name list and having it return the X and Y cords for the node if the exit name is same in order to fulfil **SC8** and **SC9** as shown below

```
def IdentifyNode(Room):
 for i in range(len(NodeName_List)):#Looks for room the exit of the room from the node text file and returns the exit's x and y cords
 if str(RoomExit_List[Room_List.index(Room)]) == str(NodeName_List[i]):
 return[NodeX_List[i],NodeY_List[i]]

 for i in range(len(Stair_Nodes)):
 if str(RoomExit_List[Room_List.index(Room)]) == str(Stair_Nodes[i]):#If room's exit is a stair nodes then code returns the stair nodes's x and y cords
 return[int(NodeX_List[NodeName_List.index(Stair_Nodes[i])]),int(NodeY_List[NodeName_List.index(Stair_Nodes[i])])]
```

| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | No        |
| D                    | No        |
| E                    | No        |
| F                    | No        |

**26/05/22:**

With the map and nodes now being identifiable through the use Map and Node Identifier functions I can finally start the implementation of CocoXiao1998's path visualizer function in order to accomplish **SC10**.

As shown by the path finder algorithm in design I shall start with the inside path functions as the path starts from the inside of the building where the room is.

To create the inside path function I first needed to see if the user was going to the room or if they were going from the room so I added two parameters with the first being for the room itself and the other being a boolean value used to decide if the function is needed to depict a path going to the room as shown below

```
| def CreatePath_Inside(Location, GoingTo) :
```

I then split these possibilities with the use of an if statement and used the IdentifyMap function in order to find the data of the map of the building that the room was in and loaded it into the find\_path\_alg function parameters in order to create the path. Afterwards I then rewrote the path onto the coloured variant and screenblited the coloured map making sure to then reset the map back to its original shape by writing the permanent map onto the colour map with as shown below

```
def CreatePath_Inside(Location, GoingTo):
 global pos1, pos2
 pos1 = [int((RoomX_List[Room_List.index(Location)])), int((RoomY_List[Room_List.index(Location)])))] #rooms cords
 pos2 = globalis[[RoomExit_List[Room_List.index(Location)]]] #exit cords
 if GoingTo == True:#if path needed to go from room to exit
 find_path_alg([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][0], [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1], int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1][1].shape[1])), int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1][1].shape[0])), cv2.imwrite([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1]), [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][2])#restores changing map back to it's original state
 cv2.imwrite([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][4], [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1])#creates path onto changing map
 Screen.fill("black")
 Screen.blit(pygame.image.load([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1]), (((Center_X)+((IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1].shape[0]))/2), (((Center_Y)-int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1].shape[1])))/2))
 elif GoingTo == False:#if path needed to go from exit to room
 find_path_alg([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][0], [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1], int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1][1].shape[1])), int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1][1].shape[0])), cv2.imwrite([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1]), [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][2])#restores changing map back to it's original state
 cv2.imwrite([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][4], [IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1])#creates path onto changing map
 Screen.fill("black")
 Screen.blit(pygame.image.load([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1]), (((Center_X)+((IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1].shape[0]))/2), (((Center_Y)-int(([IdentifyMap(RoomExit_List[Room_List.index(str(Location))])][1].shape[1])))/2))
```

| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | Yes       |
| D                    | No        |
| E                    | No        |
| F                    | No        |

**28/05/22:**

Now that the inside path function is created I now had to make the outside path function which shows the path from building to building. As shown in the algorithim from design below in order to do so the function must take start and end rooms as parameters

`def CreatePath_Outside(StartLocation, EndLocation):` with the identifyNode function being able to convert the rooms to the cords of the building they are from and substitute them into the find\_path\_alg function in order to create the path. The path is then drawn onto the chaining main map and screen blit as shown below

```
def CreatePath_Outside(StartLocation,EndLocation):
 global pos1, pos2,NodeInfo
 cv2.imwrite("MainMapChainging.jpg", MainMap_P)#reverts chainging map to it's original state
 pos1 = [int(IdentifyNode(StartLocation)[0]),int(IdentifyNode(StartLocation)[1])]# cords of the starting room's building
 pos2 = [int(IdentifyNode(EndLocation)[0]),int(IdentifyNode(EndLocation)[1])]#cords of the end room's building
 find_path_alg(MainMap_G, MainMap_C ,int((MainMap_C.shape)[1]),int((MainMap_C.shape)[0]), pos1 ,pos2 , path_color)#creates path from cords
 cv2.imwrite("MainMapChainging.jpg", MainMap_C)#path drawn onto the chainging map
 Screen.fill("black")
 Screen.blit(MainMap_Img,((Center_X)/4, 0 - (Height)/2))#screens blits map with path on it
```

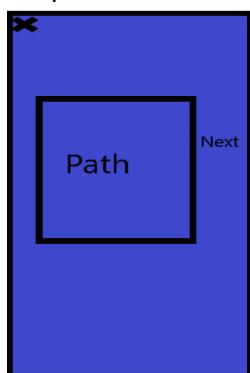
| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | Yes       |
| D                    | Yes       |
| E                    | No        |
| F                    | No        |

### **29/05/22:**

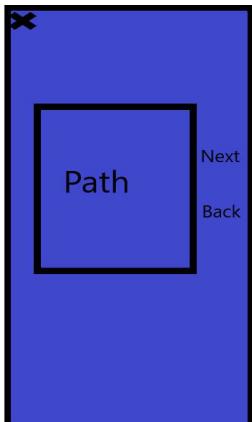
With the inside and outside path function now being completed all that was left was to dictate the order that they were used in. As shown in the algorithm from design

I shall make use of 3 path screens ,which will all be linked through the use of next and back buttons, as shown in the screen designs from design below

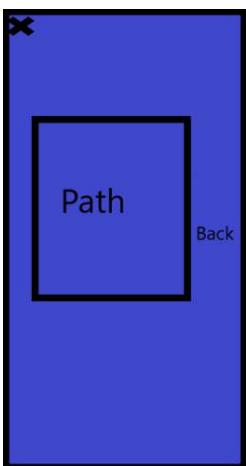
1st path:



2nd path:



3rd path:



I firstly created the next and back buttons through the use of the button class as they were necessary in order to switch the screens and further fulfil **SC3**

```
#Path Display Screen Buttons:
NextPath_Button = Button(image = None ,pos = (Center_X -(Width)/8,Center_Y) , text_input = "Next", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")
LastPath_Button = Button(image = None ,pos = (Center_X -(Width)/8,(Center_Y)/2) , text_input = "Back", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")
```

Afterwards I coded the screens by simply using the inside path function and outside path function within the correct context and linked them as shown below!

```
def Path1_Sc():
 global NextPath_Button ,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 CreatePath_Inside(ActualLocation,True)#Screen blits path from room to exit onto the screen
 for Button in [NextPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#goes back to main screen if user exits screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):#goes to 2nd path screen if user upon button being clicked
 Path2_Sc()

 pygame.display.update()
 clock.tick(60)
```

```

def Path2_Sc():
 global NextPath_Button , LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")

 CreatePath_Outside(ActualLocation,ActualDestination)
 for Button in [NextPath_Button,LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#goes back to main screen if user exits screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):#goes to 3rd path screen upon button being clicked
 Path3_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#goes back to 2nd path screenupon button being clicked
 Path1_Sc()

 pygame.display.update()
 clock.tick(60)

def Path3_Sc():
 global LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")

 CreatePath_Inside(ActualDestination,False)

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#goes back to main screen if user exits screen
 Main_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#goes back to 2nd path screenupon button being clicked
 Path2_Sc()

 pygame.display.update()
 clock.tick(60)

```

| Test Performed | Test Evidence                                                                                          | Pass Or Fail |
|----------------|--------------------------------------------------------------------------------------------------------|--------------|
| DT4            | <pre> Location_P =  IdentifyMap[Location])[2] TypeError: 'NoneType' object is not subscriptable </pre> | Fail         |

| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |

|   |     |
|---|-----|
| C | Yes |
| D | Yes |
| E | No  |
| F | No  |

**30/05/22:**

To fix the error that occurred in dt3 instead of having IdentifyMap return the values I stored the variable's contents into a list and did the same for IdentifyNode as it may run into the same errors as shown below

```

def IdentifyMap(Exit):
 global MapInfo
 for i in range(len(ScienceBlock_Nodes)):
 if ScienceBlock_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(ScienceBlock_G)
 MapInfo.append(ScienceBlock_C)
 MapInfo.append(ScienceBlock_P)
 MapInfo.append("ScienceBlockChainging.jpg")

 for i in range(len(LanguageBlock_Nodes)):
 if LanguageBlock_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(LanguageBlock_G)
 MapInfo.append(LanguageBlock_C)
 MapInfo.append(LanguageBlock_P)
 MapInfo.append("LanguageBlockChainging.jpg")

 for i in range(len(EnglishAndMusicBlock_Nodes)):
 if EnglishAndMusicBlock_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(EnglishBlock_G)
 MapInfo.append(EnglishBlock_C)
 MapInfo.append(EnglishBlock_P)
 MapInfo.append("EnglishBlockChainging.jpg")

 for i in range(len(MathsBlockDownstairs_Nodes)):
 if MathsBlockDownstairs_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(MathsBlockDownstairs_G)
 MapInfo.append(MathsBlockDownstairs_C)
 MapInfo.append(MathsBlockDownstairs_P)
 MapInfo.append("MathsDownstairsChainging.jpg")

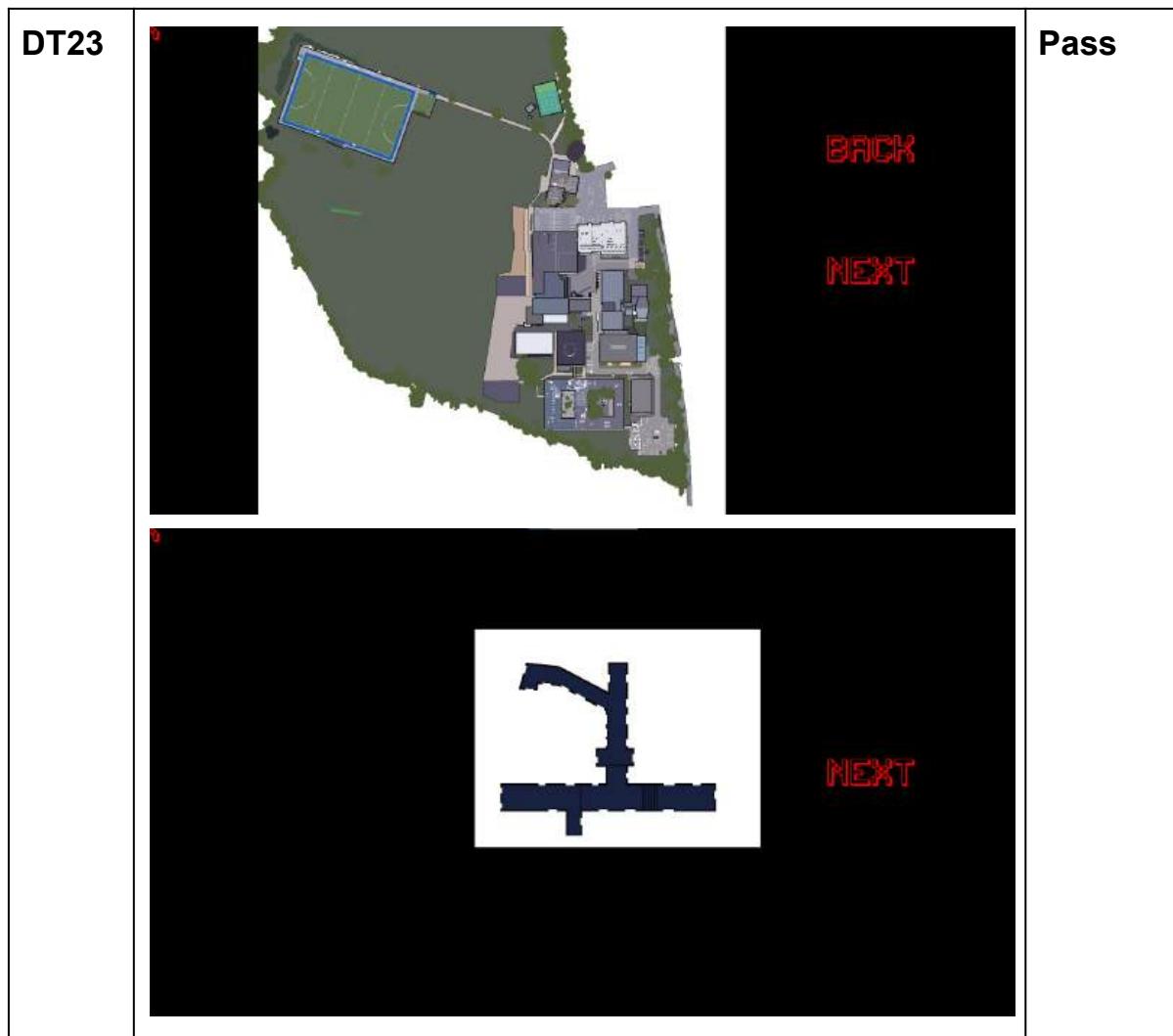
 for i in range(len(MathsBlockUpstairs_Nodes)):
 if MathsBlockUpstairs_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(MathsBlockUpstairs_G)
 MapInfo.append(MathsBlockUpstairs_C)
 MapInfo.append(MathsBlockUpstairs_P)
 MapInfo.append("MathsUpstairsChainging.jpg")

 for i in range(len(ComputingBlock_Nodes)):
 if ComputingBlock_Nodes[i] == str(Exit):
 MapInfo.clear() #get rid of previous data
 MapInfo.append(ComputingBlock_G)
 MapInfo.append(ComputingBlock_C)
 MapInfo.append(ComputingBlock_P)
 MapInfo.append("ComputingBlockChainging.jpg")

def IdentifyNode(Room):
 global NodeInfo
 NodeInfo.clear() #get rid of previous data
 for i in range(len(NodeName_List)):#Looks for room the exit of the room from the node text file and adds the exit's x and y cords to the current node info list
 if str(RoomExit_List[Room_List.index(Room)]) == str(NodeName_List[i]):
 NodeInfo.append(NodeX_List[i])
 NodeInfo.append(NodeY_List[i])

 for i in range(len(Stair_Nodes)):#If room's exit is a stair nodes then code adds the stair nodes's x and y cords to the current node info list
 if str(RoomExit_List[Room_List.index(Room)]) == str(Stair_Nodes[i]):
 NodeInfo.append(int(NodeX_List[NodeName_List.index(Stair_Nodes[i])]))
 NodeInfo.append(int(NodeY_List[NodeName_List.index(Stair_Nodes[i])]))
```

| Test Performed | Test Evidence                                                                                                                                                               | Pass Or Fail             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| DT4            |                                                                                           | Fail                     |
| DT22           | <br> | Pass<br><br>BACK<br>NEXT |



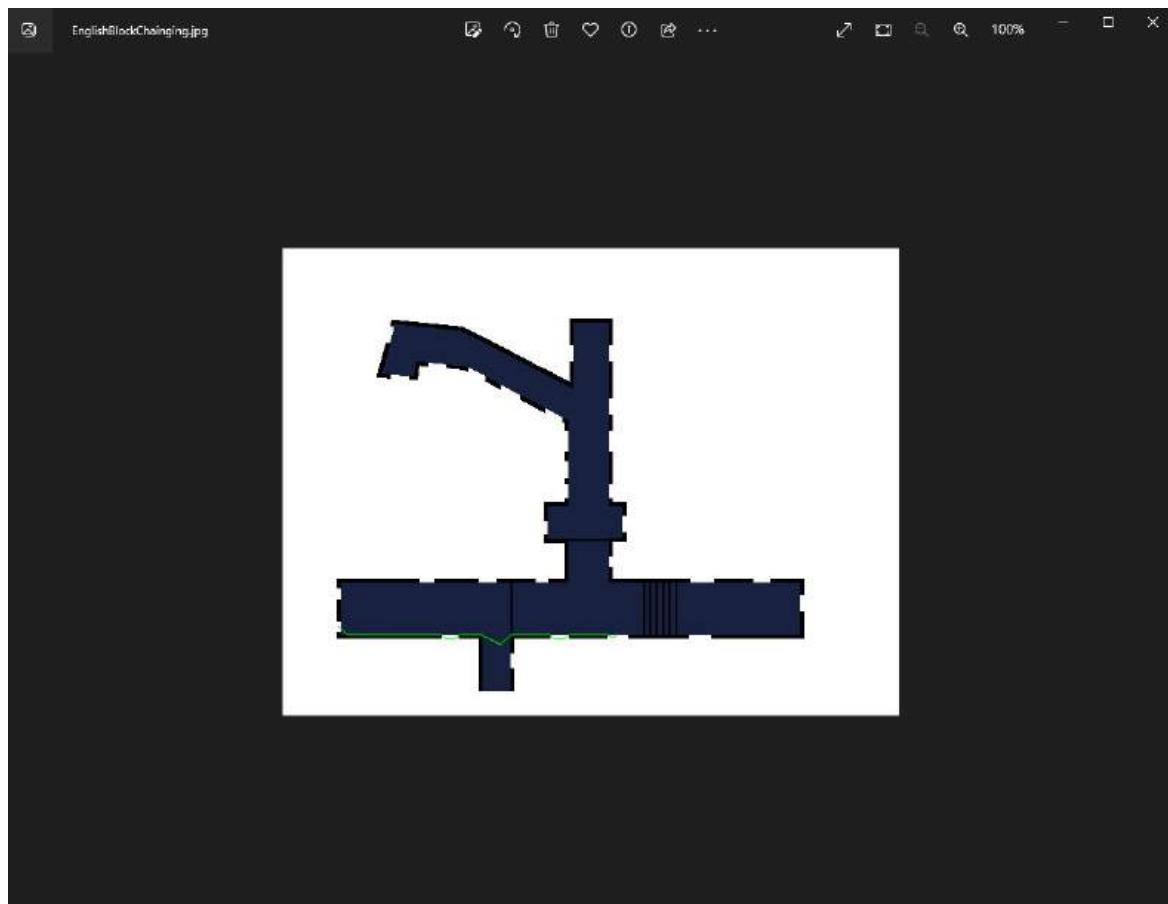
| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | Yes       |
| D                    | Yes       |
| E                    | No        |
| F                    | No        |

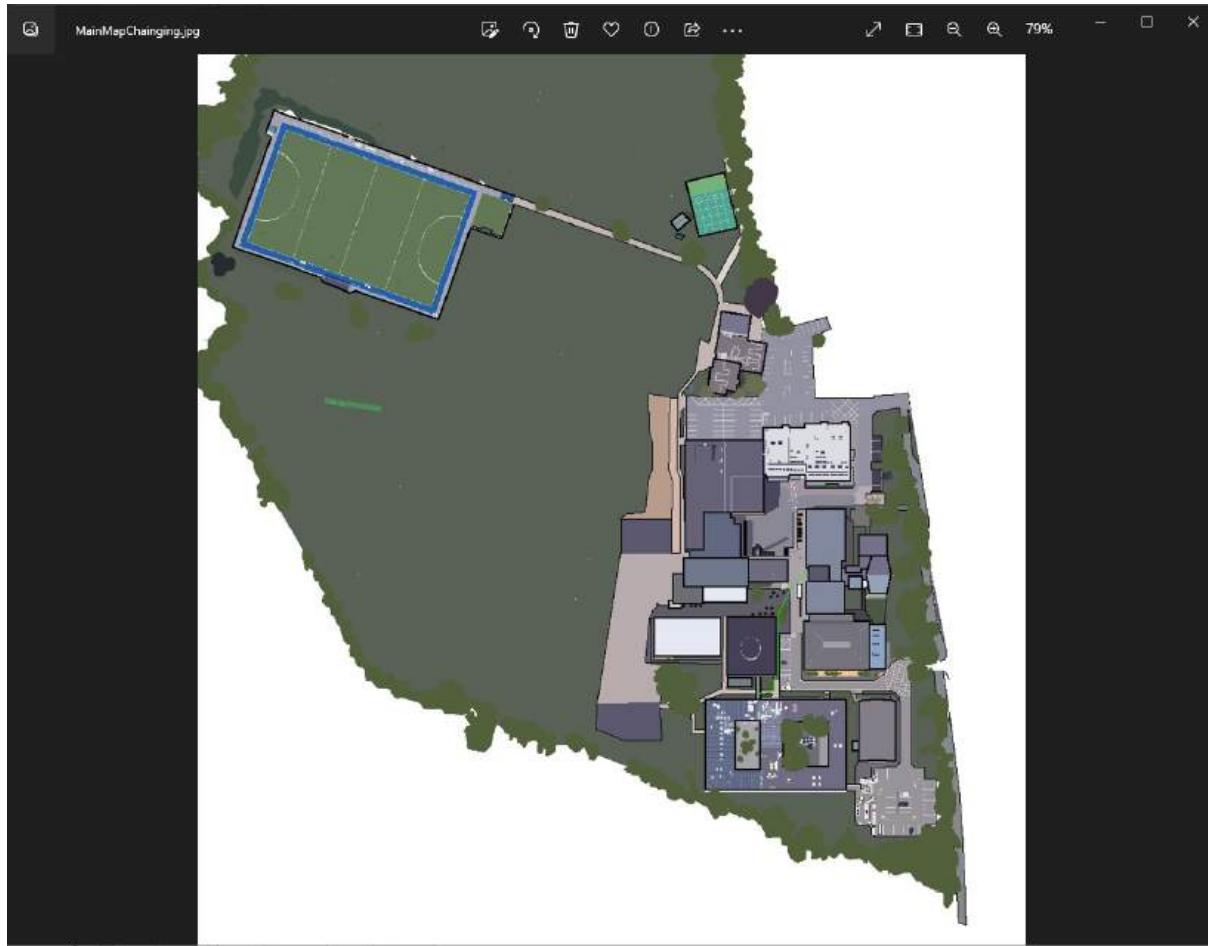
| Success Criteria | Achieved partially or fully or again | Explanation |
|------------------|--------------------------------------|-------------|
|                  |                                      |             |

|     |       |                                                                       |
|-----|-------|-----------------------------------------------------------------------|
| SC3 | Again | The next button and back button are capable are swapping path screens |
|-----|-------|-----------------------------------------------------------------------|

**31/05/22:**

Although no error was produced I was able to tell from the result of DT4 that the problem lied within the screen blit as I checked the actual file of the chaining map and it had changed





what seemed to be the problem was the fact that the code stored the unchanged version of it as an image at the start and thus did not screen blot the version with the path on the map so to fix this I loaded the new version of the image to be screen blit by inputting the location through pygame as a part of the screen blot as shown below

```

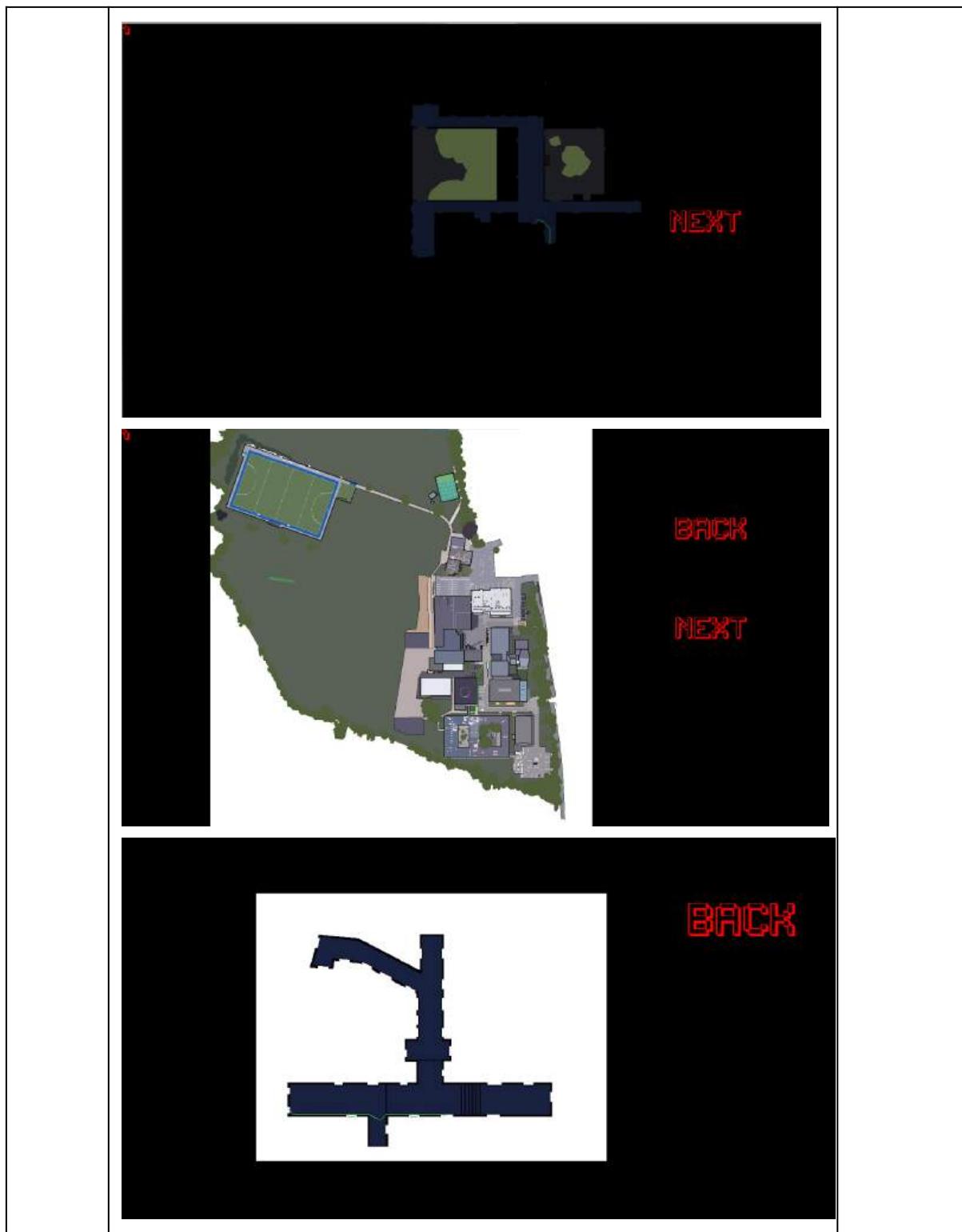
def CreatePath_Inside(Location,GoingTo):
 global pos1, pos2, MapInfo
 IdentifyMap(RoomExit_List[Room_List.index(str(Location))])#Identifies the incide map needed
 cv2.imwrite(MapInfo[3],MapInfo[2])#restores chainging map back to it's original state
 pos1 = [int(RoomList[Room_List.index(Location)]),int(RoomY_List[Room_List.index(Location)])]#cords cords
 pos2 = [global](RoomExit_List[Room_List.index(Location)])#exit cords
 if GoingTo == True:
 find_path_alg(MapInfo[0],MapInfo[1],int((MapInfo[1]).shape[1]),int((MapInfo[1]).shape[0]),pos1,pos2, path_color)
 cv2.imwrite(MapInfo[3],MapInfo[1])#draws path onto chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load(MapInfo[3]),((Center_X)+(int((MapInfo[1]).shape[0])/2),(Center_Y)-(int((MapInfo[1]).shape[1])/2)))#screenblits map with path onto the screen
 elif GoingTo == False:
 find_path_alg(MapInfo[0],MapInfo[1],int((MapInfo[1]).shape[1]),int((MapInfo[1]).shape[0]),pos2,pos1, path_color)
 cv2.imwrite(MapInfo[3],MapInfo[1])#draws path onto chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load(MapInfo[3]),((Center_X)+(int((MapInfo[1]).shape[0])/2),(Center_Y)-(int((MapInfo[1]).shape[1])/2)))#screenblits map with path onto the screen

def CreatePath_Outside(StartLocation,EndLocation):
 global pos1, pos2,NodeInfo
 cv2.imwrite("MainMapChainging.jpg", MainMap_P)#reverts chainging map to it's original state
 IdentifyNode(StartLocation)
 pos1 = [int(NodeInfo[0]),int(NodeInfo[1])]# cords of the starting room's building
 IdentifyNode(EndLocation)
 pos2 = [int(NodeInfo[0]),int(NodeInfo[1])]#cords of the end room's building
 find_path_alg(MainMap_G, MainMap_C,int((MainMap_C.shape)[1]),int((MainMap_C.shape)[0]), pos1 ,pos2 , path_color)#creates path from cords
 cv2.imwrite("MainMapChainging.jpg", MainMap_C)#path drawn onto the chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainMapChainging.jpg"),((Center_X)/4, 0 - (Height)/2))#screens blits map with path on it

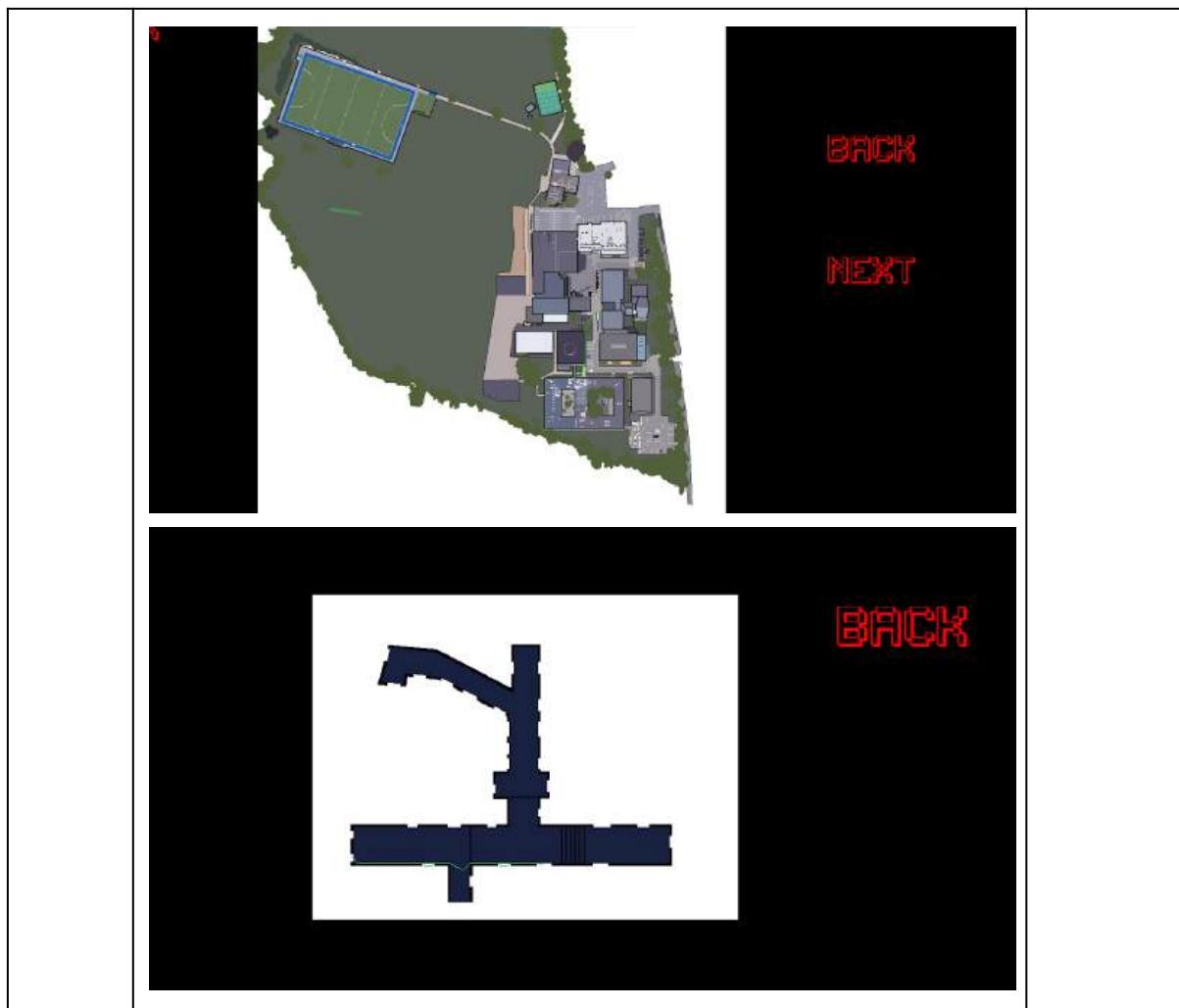
```

| Test Perfo | Test Evidence | Pass Or Fail |
|------------|---------------|--------------|
|------------|---------------|--------------|

| rmed |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| DT4  | <br><br> <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE <b>SEARCH</b></p> <p>S12</p> <p>LIKELY RESULT 1: S12 ●</p> <p>LIKELY RESULT 2: S10 ●</p> <p>LIKELY RESULT 3: S11 ●</p> <p>LIKELY RESULT 4: S13 ●</p><br> <p>PLEASE CLICK TO ENTER YOUR DESTINATION HERE <b>SEARCH</b></p> <p>E2</p> <p>LIKELY RESULT 1: E2 ●</p> <p>LIKELY RESULT 2: ●</p> <p>LIKELY RESULT 3: ●</p> <p>LIKELY RESULT 4: ●</p> | Pass |
|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |      |



|     |                                                                                                                                                                          |      |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| DT5 | <br> | Pass |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|



| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |
| C                    | Yes       |
| D                    | Yes       |
| E                    | Yes       |
| F                    | No        |

| Success Criteria | Achieved partially or fully or again | Explanation              |
|------------------|--------------------------------------|--------------------------|
| SC3              | Again                                | The next button and back |

|      |       |                                                                                                                |
|------|-------|----------------------------------------------------------------------------------------------------------------|
|      |       | button are capable are swapping path screens                                                                   |
| SC8  | Again | The path finder displayed in DT4 clearly displays that the code is capable of finding the inputted location    |
| SC9  | Again | The path finder displayed in DT4 clearly displays that the code is capable of finding the inputted destination |
| SC10 | Fully | The screenshot from DT4 shows that the code is able to create a clearly highlighted route                      |

### **01/06/22 - 02/06/22:**

As stated on the 23/05/22 as some paths require the user to transverse through stairs I have to specifically code those unique paths from rooms near stairs.

To do this I used the stairs list I made on the 23/05/22 containing the room exits of the stairs in order to identify them as shown below

```
if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if str(RoomExit_List[Room_List.index(actualLocation)]) in Stair_Nodes == True:#Goes to the unique path creator for stair nodes in the instance the path has a stairwell
 Path_5_Set()
```

This the linked from the normal paths to the unique path screens where each path from a room with a stair node to the stairs and the path from the stairs to the building exit is pre coded as shown below:

```
def Path_5_Set():#path from stairwell entrance to stairwell exit
 global NewPath_Button,LectPath_Button,C_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 evl.inwrite("LanguageStairFloodChaining.jpg", LanguageStairFront_C)#draws path onto the chaining version of the map
 Screen.blit(pygame.image.load("LanguageStairFrontChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("LanguageStairFrontChaining.jpg", LanguageStairFront_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it

 if RoomExit_List[Room_List.index(actualLocation)] == "#B_STI" or RoomExit_List[Room_List.index(actualLocation)] == "#B_STI":#check if the room is update in the language block from the room exit node
 find_path_sig(LanguageStairFront_C, LanguageStairFront_C,inc((LanguageStairFront_C.shape[1]),inc((LanguageStairFront_C.shape[0]),int(LBStairsFrom_Entrance),int(LBStairsFront_Exit)), path_color))#creates path
 evl.inwrite("LanguageStairFloodChaining.jpg", LanguageStairFront_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("LanguageStairChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("LanguageStairChaining.jpg", LanguageStairFront_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it

 if RoomExit_List[Room_List.index(actualLocation)] == "#H_STI" or RoomExit_List[Room_List.index(actualLocation)] == "#H_STI":#check if the room is update in the path block from the room exit node
 find_path_sig(BathStairsFrom_C, BathStairsFrom_C,inc((BathStairsFrom_C.shape[1]),inc((BathStairsFrom_C.shape[0]),int(HBStairsFrom_Entrance),int(HBStairsFront_Exit)), path_color))#creates path
 evl.inwrite("BathStairsChaining.jpg", BathStairsFrom_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("BathStairsChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("BathStairsChaining.jpg", BathStairsFrom_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it

 if RoomExit_List[Room_List.index(actualLocation)] == "#T_STI" or RoomExit_List[Room_List.index(actualLocation)] == "#T_STI":#check if the room is update in the dt block from the room exit node
 find_path_sig(DTStairsFrom_C, DTStairsFrom_C,inc((DTStairsFrom_C.shape[1]),inc((DTStairsFrom_C.shape[0]),int(DTStairsFrom_Entrance),int(DTStairsFront_Exit)), path_color))#creates path
 evl.inwrite("DTStairsChaining.jpg", DTStairsFrom_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("DTStairsChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("DTStairsChaining.jpg", DTStairsFrom_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it

 if RoomExit_List[Room_List.index(actualLocation)] == "#C_STI" or RoomExit_List[Room_List.index(actualLocation)] == "#C_STI":#check if the room is update in the main block from the room exit node
 find_path_sig(MainBlockFrom_C, MainBlockFrom_C,inc((MainBlockFrom_C.shape[1]),inc((MainBlockFrom_C.shape[0]),ComputingBlockStairFrom_Entrance,ComputingBlockStairFrom_Exit), path_color))#creates path
 evl.inwrite("MainBlockFromChaining.jpg", MainBlockFrom_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("MainBlockFromChaining.jpg", MainBlockFrom_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it

 if RoomExit_List[Room_List.index(actualLocation)] == "#G_STI" or RoomExit_List[Room_List.index(actualLocation)] == "#G_STI":#check if the room is update in the main block from the room exit node
 find_path_sig(GeographyStairFrom_C, GeographyStairFrom_C,inc((GeographyStairFrom_C.shape[1]),inc((GeographyStairFrom_C.shape[0]),(GeographyBlockStairFrom_Entrance),(GeographyBlockStairFrom_Exit)), path_color))#creates path
 evl.inwrite("GeographyStairFromChaining.jpg", GeographyStairFrom_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("GeographyStairFromChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 evl.inwrite("GeographyStairFromChaining.jpg", GeographyStairFrom_C,Path_5_Set(),Path_5_Set())#rewrites map to original version without the path on it
```

```

def Path1_3D_Sc():#path from stairwell exit to building's exit
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Screen.fill("black")
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 if RoomExit_List[Room_List.index(ActualLocation)] == "1B_ST1" or RoomExit_List[Room_List.index(ActualLocation)] == "1B_ST2":#checks if the room is upstairs in the language block from the rooms exit node
 find_path_smg(LanguageStairsFromB_C ,LanguageStairsFromB_C.shape[1],int((LanguageStairsFromB_G.shape[1]), int((LanguageStairsFromB_G.shape[0])),int(DTSstairsFromB_Extrance),int(DTSstairsFromB_Exit)), path_color)#creates path
 cv2.imwrite("LanguageStairsFromBChaining.jpg", LanguageStairsFromB_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("LanguageStairsFromBChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 cv2.imwrite("LanguageStairsFromBChaining.jpg", LanguageStairsFromB_F)#reverts map to original version without the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "1T_ST1" or RoomExit_List[Room_List.index(ActualLocation)] == "1T_ST2":#checks if the room is upstairs in the dt block from the rooms exit node
 find_path_smg(DTSstairsFromB_C ,DTSstairsFromB_C.shape[1],int(DTSstairsFromB_G.shape[0]),int(DTSstairsFromB_Extrance),int(DTSstairsFromB_Exit), path_color)#creates path
 cv2.imwrite("DTSstairsFromBChaining.jpg", DTSstairsFromB_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("DTSstairsFromBChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 cv2.imwrite("DTSstairsFromBChaining.jpg", DTSstairsFromB_F)#reverts map to original version without the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "C_ST1" or RoomExit_List[Room_List.index(ActualLocation)] == "C_ST2":#checks if the room is upstairs in the main block from the rooms exit node
 find_path_smg(MainBlockFromC_B ,MainBlockFromC_C.shape[1],int((MainBlockFromC_G.shape[1]), int((MainBlockFromC_G.shape[0])),int(GeographyBlockStairFromC_Extrance),int(GeographyBlockStairFromC_Exit)), path_color)#creates path
 cv2.imwrite("MainBlockFromCChaining.jpg", MainBlockFromC_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromCChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 cv2.imwrite("MainBlockFromCChaining.jpg", MainBlockFromC_F)#reverts map to original version without the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "G_ST1" or RoomExit_List[Room_List.index(ActualLocation)] == "G_ST2":#checks if the room is upstairs in the main block from the rooms exit node
 find_path_smg(MainBlockFromG_B ,MainBlockFromG_C.shape[1],int((MainBlockFromG_G.shape[1]), int((MainBlockFromG_G.shape[0])),int(ComputingBlockStairFromG_Extrance),int(ComputingBlockStairFromG_Exit)), path_color)#creates path
 cv2.imwrite("MainBlockFromGChaining.jpg", MainBlockFromG_C)#draws path onto the chaining version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromGChaining.jpg"),(Center_X,Center_Y))#displays map with the path on it
 cv2.imwrite("MainBlockFromGChaining.jpg", MainBlockFromG_F)#reverts map to original version without the path on it

```

Afterwards I then involved added if statements to the nodeinfo function in order to get the correct node information of the nodes within the stair nodes list as shown below

```

def IdentifyNode(Room):
 global NodeInfo
 NodeInfo.clear()#get rid of previous data
 for i in range(len(NodeName_List)):#loops for room the exit of the room from the node text file and adds the existing x and y cords to the current node info list
 if str(RoomExit_List[Room_List.index(Room)]) == str(NodeName_List[i]):
 NodeInfo.append(NodeF_List[NodeName_List.index(Room)])#appends the x cord to node information list
 NodeInfo.append(NodeT_List[NodeName_List.index(Room)])#appends the y cord to node information list

 if (RoomExit_List[Room_List.index(Room)]) == "1B_ST1" or (RoomExit_List[Room_List.index(Room)]) == "1B_ST2" or (RoomExit_List[Room_List.index(Room)]) == "1C_ST1" or (RoomExit_List[Room_List.index(Room)]) == "1C_ST2":#checks to see if the room is upstairs in the language block
 NodeInfo.append(int(NodeF_List[NodeName_List.index("1B_EX1")]))#appends the x cord to node information list
 NodeInfo.append(int(NodeT_List[NodeName_List.index("1B_EX1")]))#appends the y cord to node information list

 if (RoomExit_List[Room_List.index(Room)]) == "1B_ST1" or (RoomExit_List[Room_List.index(Room)]) == "1B_ST2":#checks to see if the room is upstairs in the language block
 NodeInfo.append(int(NodeF_List[NodeName_List.index("1B_EX1")]))#appends the x cord to node information list
 NodeInfo.append(int(NodeT_List[NodeName_List.index("1B_EX1")]))#appends the y cord to node information list

 if (RoomExit_List[Room_List.index(Room)]) == "1T_ST1" or (RoomExit_List[Room_List.index(Room)]) == "1T_ST2":#checks to see if the room is upstairs in the dt block
 NodeInfo.append(int(NodeF_List[NodeName_List.index("1T_EX1")]))#appends the x cord to node information list
 NodeInfo.append(int(NodeT_List[NodeName_List.index("1T_EX1")]))#appends the y cord to node information list

 if (RoomExit_List[Room_List.index(Room)]) == "C_ST1" or (RoomExit_List[Room_List.index(Room)]) == "C_ST2":#checks to see if the room is upstairs in the main block
 NodeInfo.append(int(NodeF_List[NodeName_List.index("C_EX1")]))#appends the x cord to node information list
 NodeInfo.append(int(NodeT_List[NodeName_List.index("C_EX1")]))#appends the y cord to node information list

```

| Test Performed | Test Evidence | Pass Or Fail |
|----------------|---------------|--------------|
|                |               |              |

DT6



Pass

PLEASE CLICK TO ENTER YOUR LOCATION HERE

SEARCH

16

LIKELY RESULT 1: 16



LIKELY RESULT 2:



LIKELY RESULT 3:



LIKELY RESULT 4:



PLEASE CLICK TO ENTER YOUR DESTINATION HERE

SEARCH

DT2

LIKELY RESULT 1: DT2



LIKELY RESULT 2: DT1

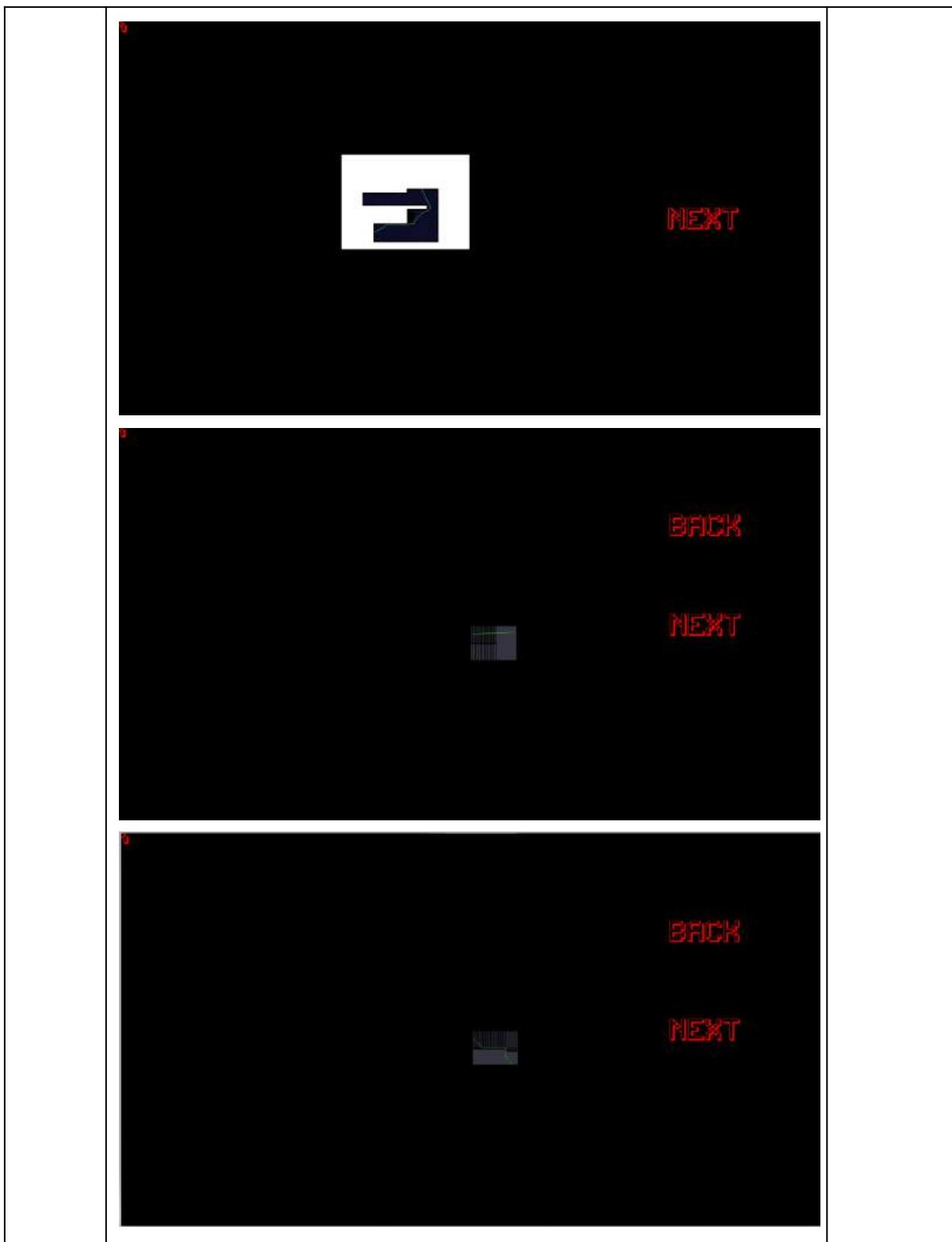


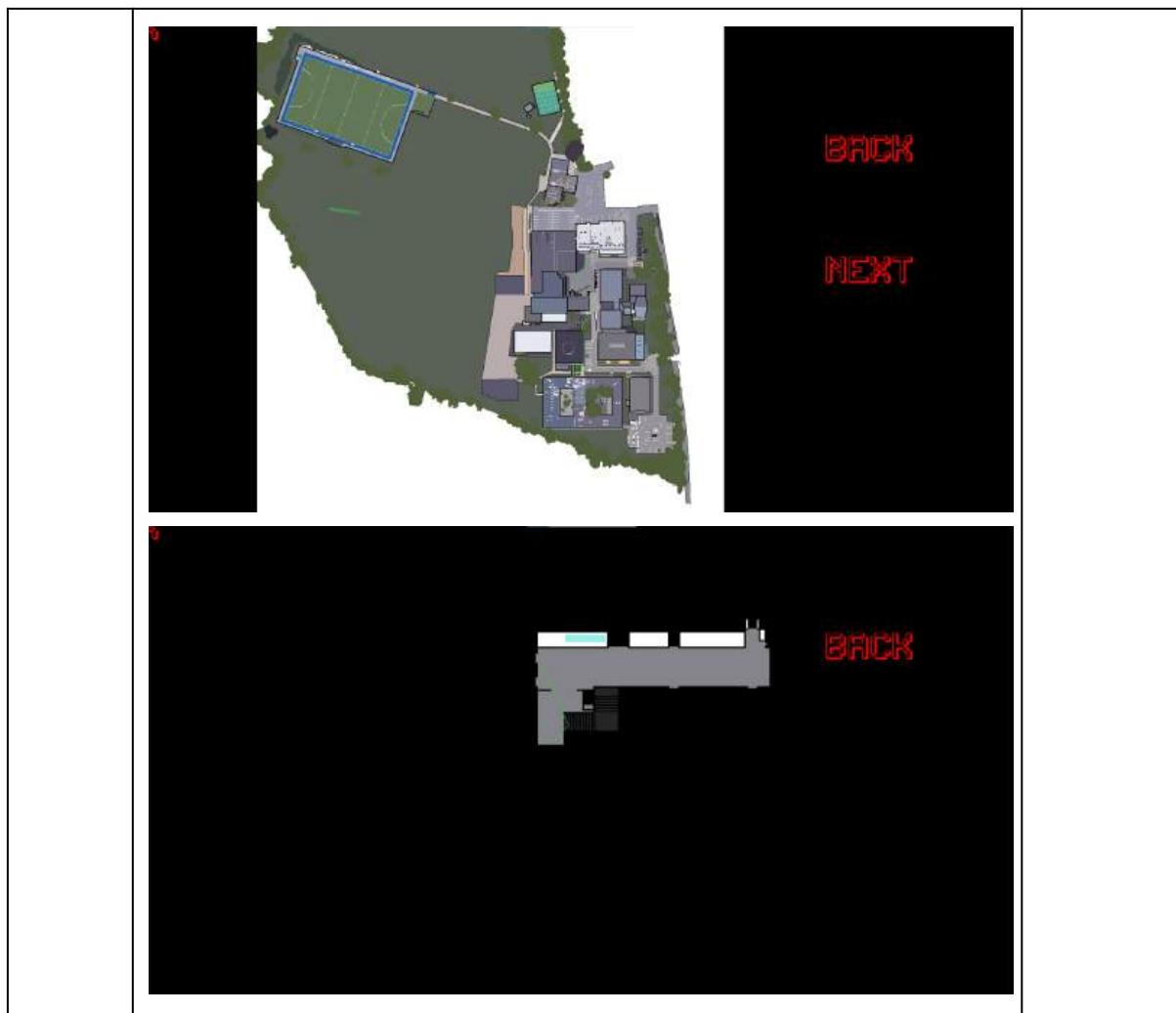
LIKELY RESULT 3: DT3

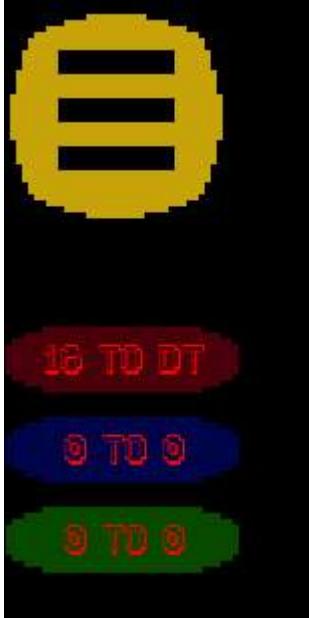
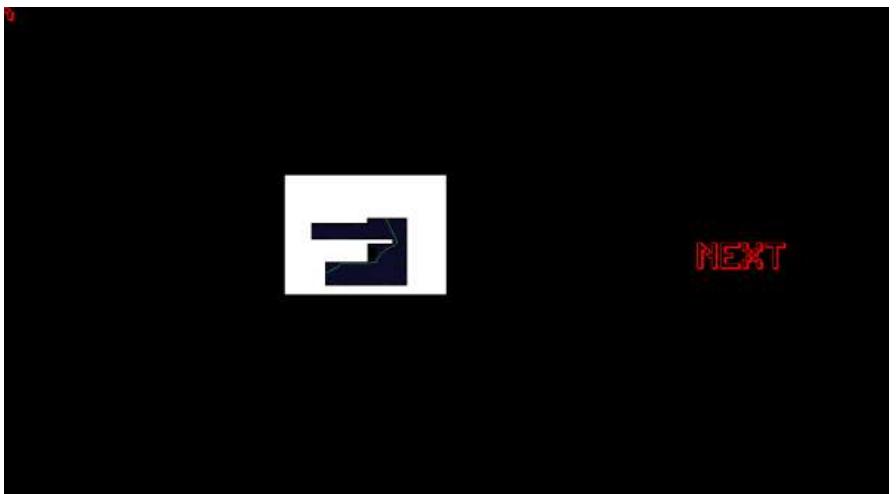
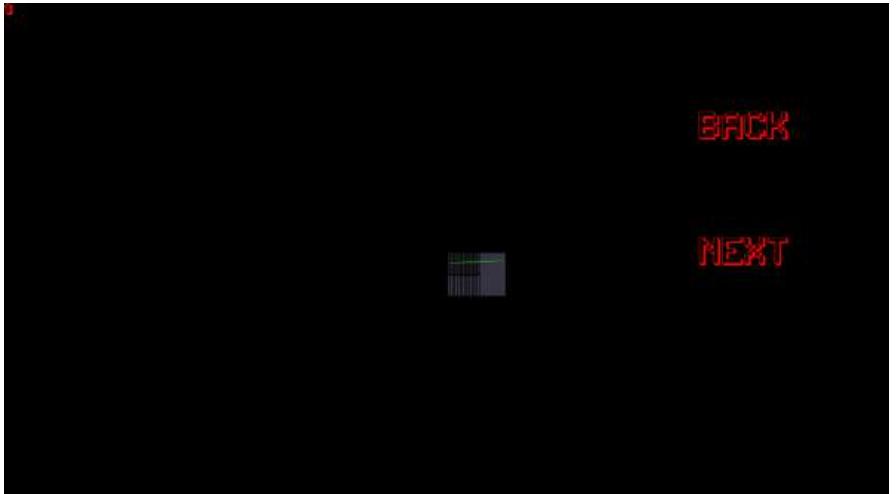


LIKELY RESULT 4: DT4







|     |                                                                                                                                                                                                                                                                  |      |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| DT7 | <br><br> | Pass |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|



| Sub Modules of Task4 | Completed |
|----------------------|-----------|
| A                    | Yes       |
| B                    | Yes       |

|   |     |
|---|-----|
| C | Yes |
| D | Yes |
| E | Yes |
| F | Yes |

| <b>Success Criteria</b> | <b>Achieved partially or fully or again</b> | <b>Explanation</b>                                                                                                                   |
|-------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| SC8                     | Again                                       | As shown in dt6 the code is able to locate the inputted destination of DT2.                                                          |
| SC9                     | Again                                       | As shown in dt6 the code is able to locate the inputted location of room 16.                                                         |
| SC10                    | Again                                       | As shown in dt6 and dt7 the code is capable of producing and displaying a highlighted route from the start location to end location. |

## **Client Review:**

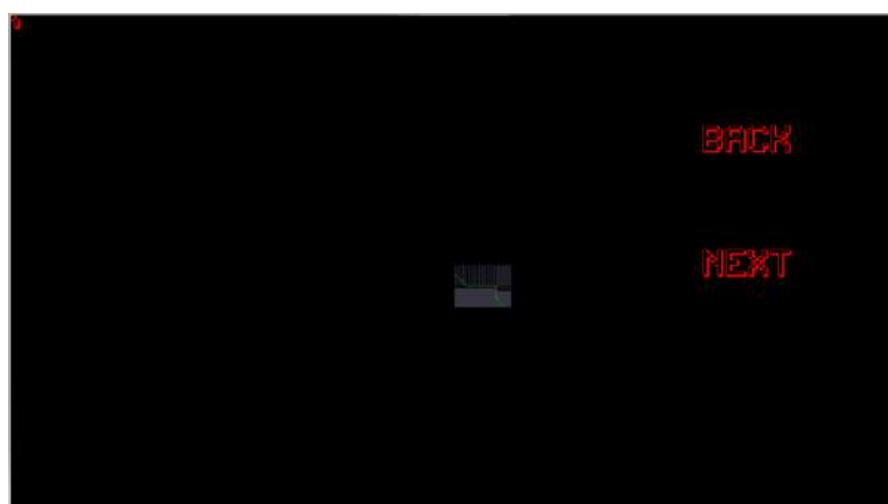
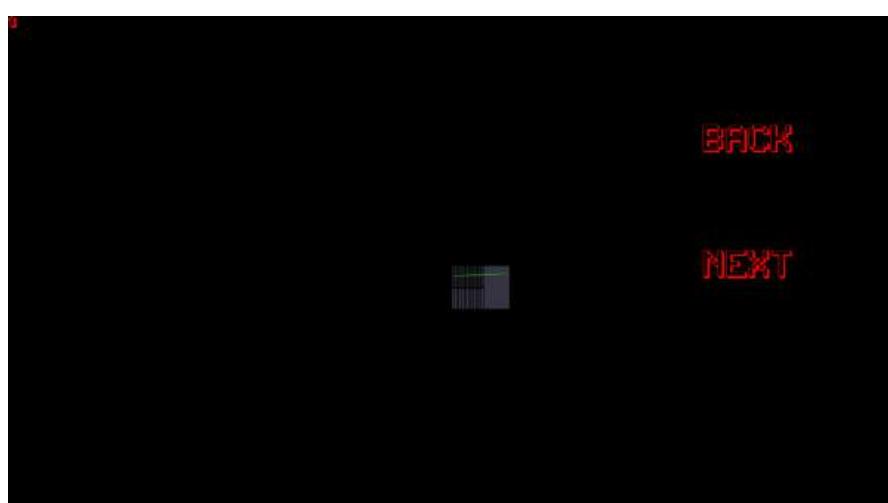
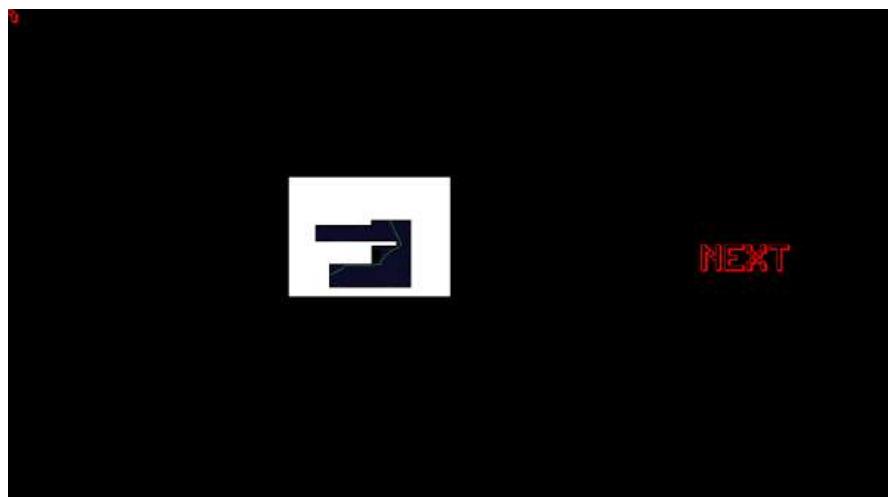
After completing the last task 4 I sent an email to my client so that I can get feedback to make improvements to the product and thus following the iterative development as shown below:

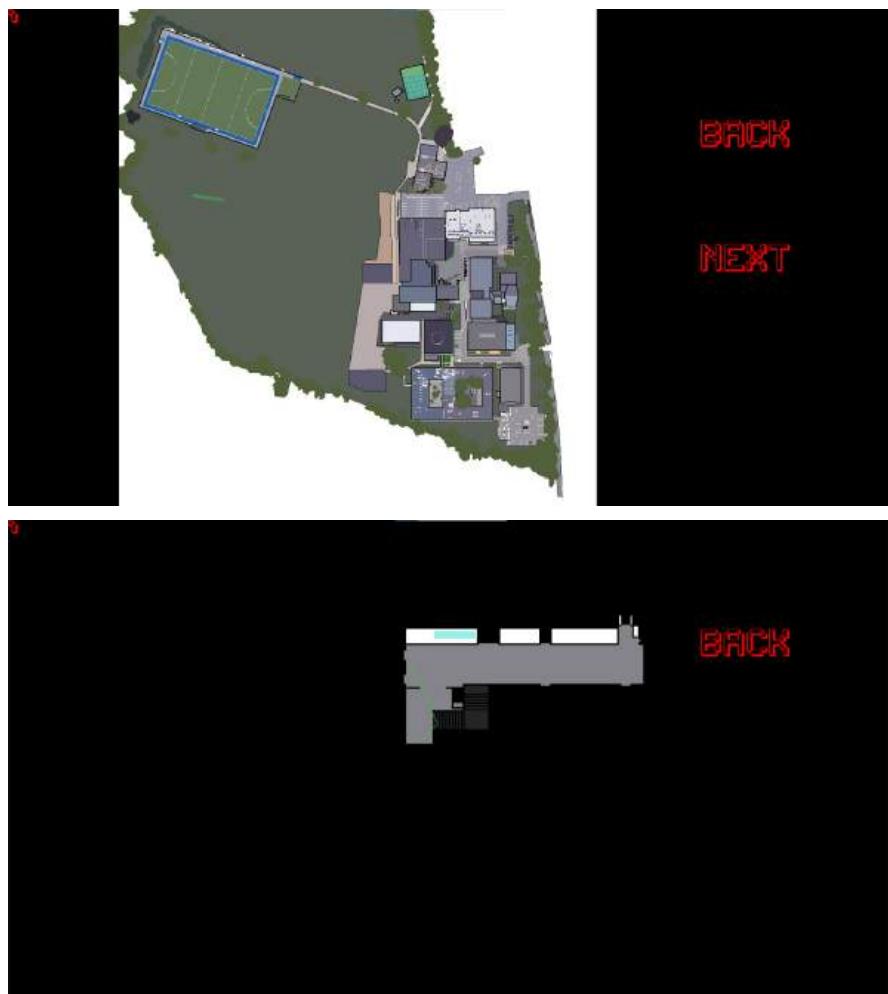
"To Whom This May Concern,

I am contacting you as I have now completed the last stage of the development which entailed the coding of the path finder. With the path finder now complete you can fully utilize both the search function and pin function as shown below.

Search:







Pins:





I would appreciate if you could suggest any potential additions

Yours Faithfully , Tireni Oluwabunmi"

**Joseph Baker:** "I have no complaints when it comes too this section."

**Alvin Roy:** "Would it be possible to change the colour of the path as it's somewhat difficult to see even when full screen"

**Mr Cooker:** "I feel like green is slightly hard too see on the main map so how about changing the path colour to red."

## **Review Modifications:**

From my client review I have been made aware that:

A) It would be optimal if the path colour was red instead of green

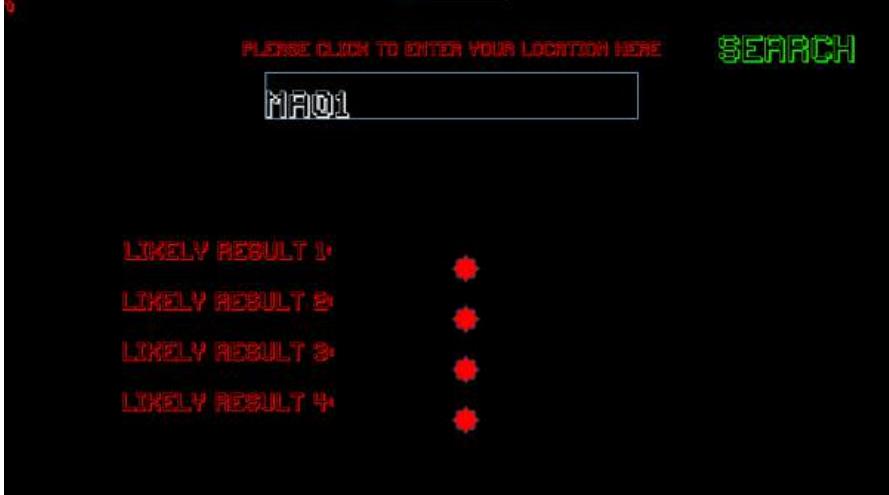
A:

To accomplish this all I had to do was change the rgb value of the path colour from green to

```
#PathFinder Vairables
red path_color = (0, 0, 255) #colour of the path
```



## **Evaluation:**

| <b><u>Beta Testing:</u></b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                     |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <b>Test Number</b>          | <b>Evidence</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Pass or Fail</b> |
| PT1                         |  <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>MA01</p> <p>LIKELY RESULT 1: MA01</p> <p>LIKELY RESULT 2: MA02</p> <p>LIKELY RESULT 3: MA03</p> <p>LIKELY RESULT 4: MA04</p><br> <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <p>SEARCH</p> <p>MA01</p> <p>LIKELY RESULT 1: MA01</p> <p>LIKELY RESULT 2: MA02</p> <p>LIKELY RESULT 3: MA03</p> <p>LIKELY RESULT 4: MA04</p> | Pass                |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PT2 | <p>A screenshot of a web-based search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE" and a "SEARCH" button. Below the input field, the text "E1" is entered. Underneath, the heading "LIKELY RESULT 1:" is followed by four small red circular icons. The heading "LIKELY RESULT 2:" is followed by three small red circular icons. The heading "LIKELY RESULT 3:" is followed by two small red circular icons. The heading "LIKELY RESULT 4:" is followed by one small red circular icon.</p> | Pass |
|     | <p>A screenshot of a web-based search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE" and a "SEARCH" button. Below the input field, the text "E1" is entered. Underneath, the heading "LIKELY RESULT 1:" is followed by four small red circular icons. The heading "LIKELY RESULT 2:" is followed by three small red circular icons. The heading "LIKELY RESULT 3:" is followed by two small red circular icons. The heading "LIKELY RESULT 4:" is followed by one small red circular icon.</p> |      |
| PT3 | <p>A screenshot of a map titled "PINS ROOMS". The map shows a green polygonal area with several buildings and icons. One building has a blue outline and is labeled "1". Another building is labeled "2". A third building is labeled "3". To the right of the map, there is a list of room counts:<ul style="list-style-type: none"><li>1: 0 TO 0</li><li>2: 0 TO 0</li><li>3: 0 TO 0</li></ul></p>                                                                                                                                                               | Pass |

9

PLEASE CLICK TO ENTER YOUR LOCATION HERE

SEARCH

MR01

LIKELY RESULT 1: MR01 ●

LIKELY RESULT 2: MR02 ●

LIKELY RESULT 3: MR03 ●

LIKELY RESULT 4: MR04 ●

9

PLEASE CLICK TO ENTER YOUR DESTINATION HERE

SEARCH

P2

LIKELY RESULT 1: P2 ●

LIKELY RESULT 2: ●

LIKELY RESULT 3: ●

LIKELY RESULT 4: ●

9

**PINS ROOMS**

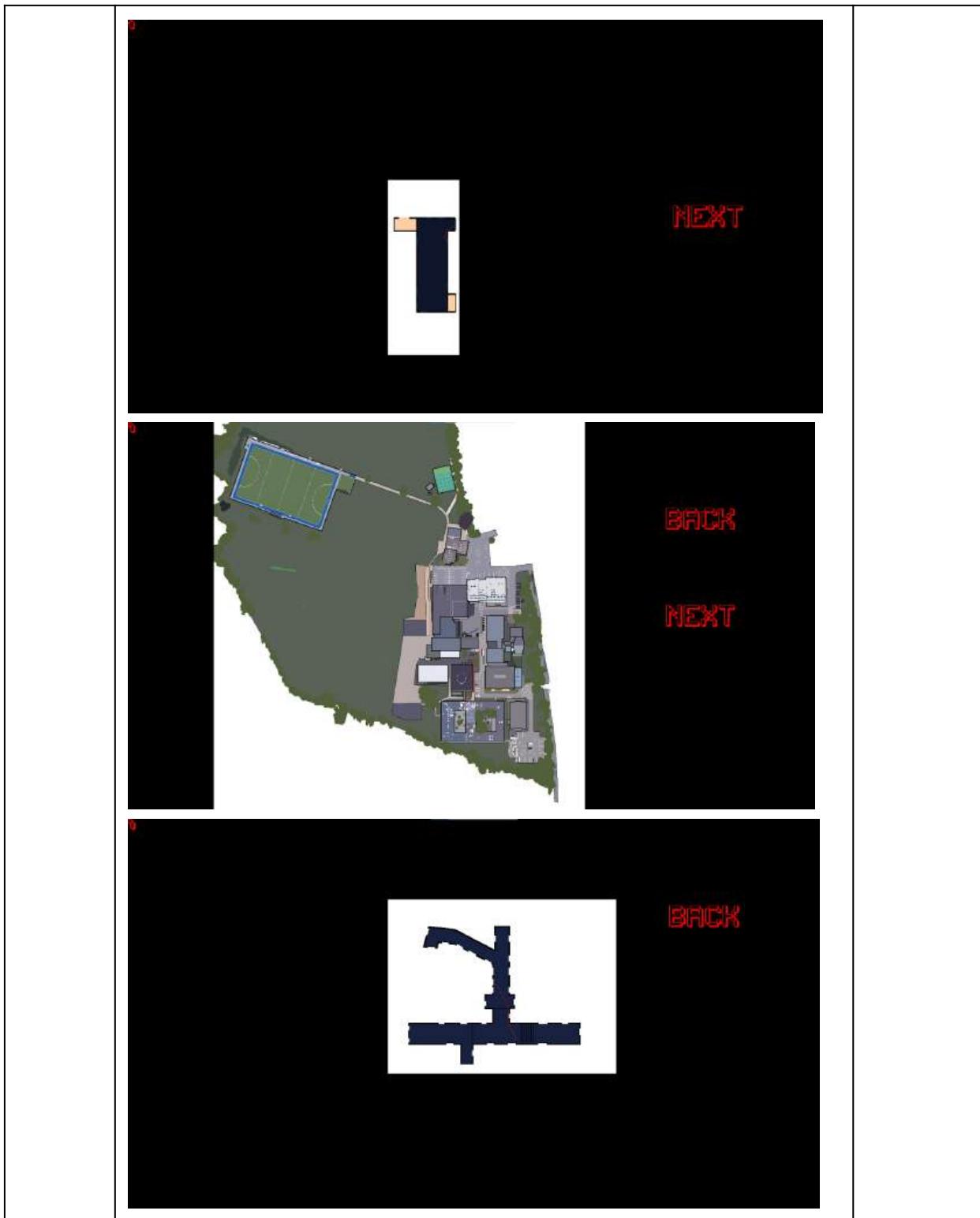
1 MR01 TO P2

2 0 TO 0

3 0 TO 0



|     |                                                                                                                                                                             |      |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|     |                                                                                            |      |
| PT4 | <br> | Pass |

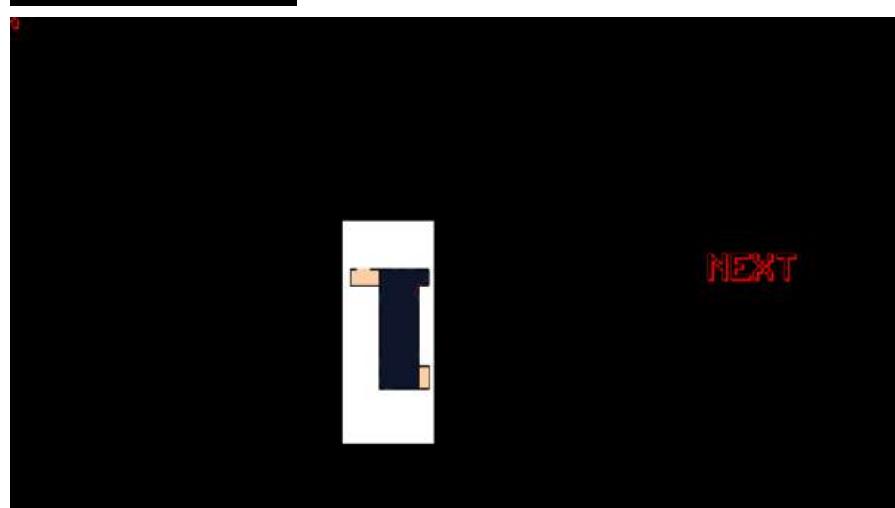


PT5



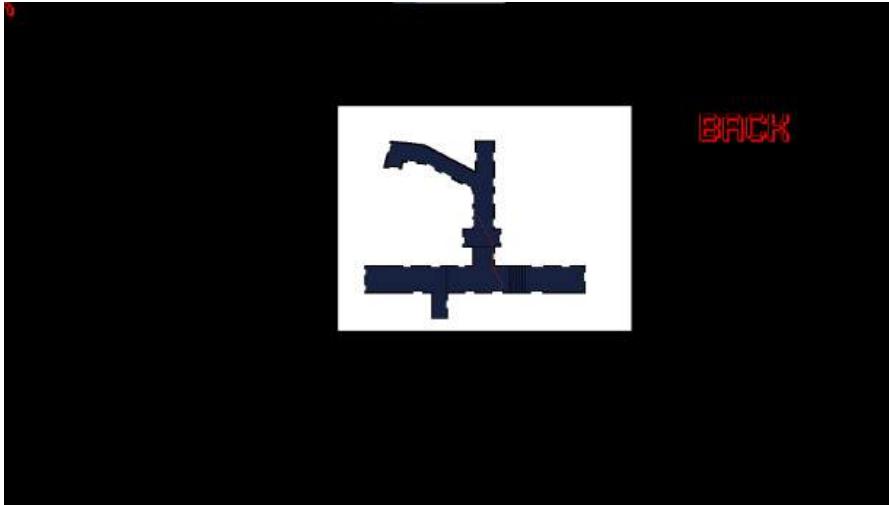
Pass

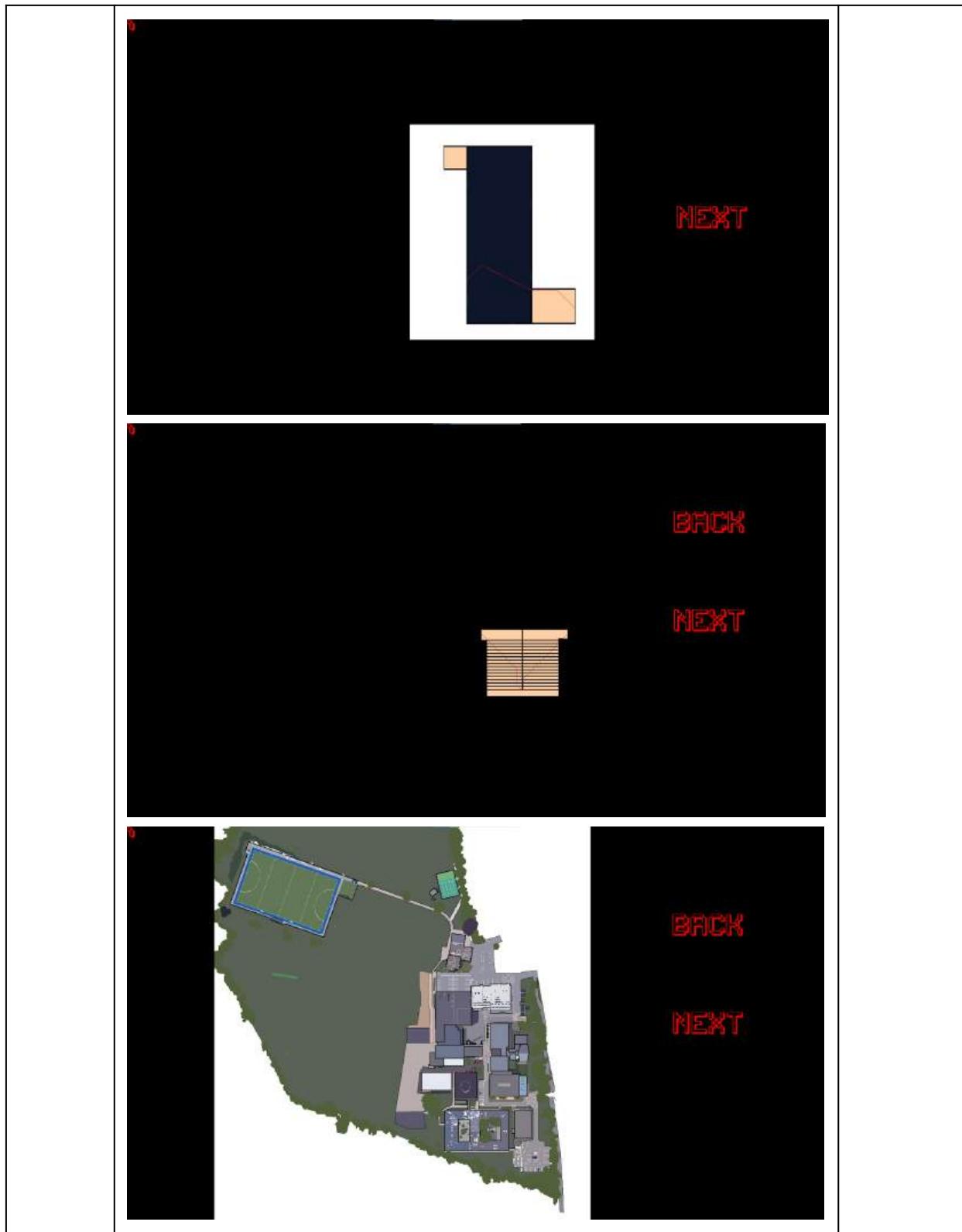
NEXT

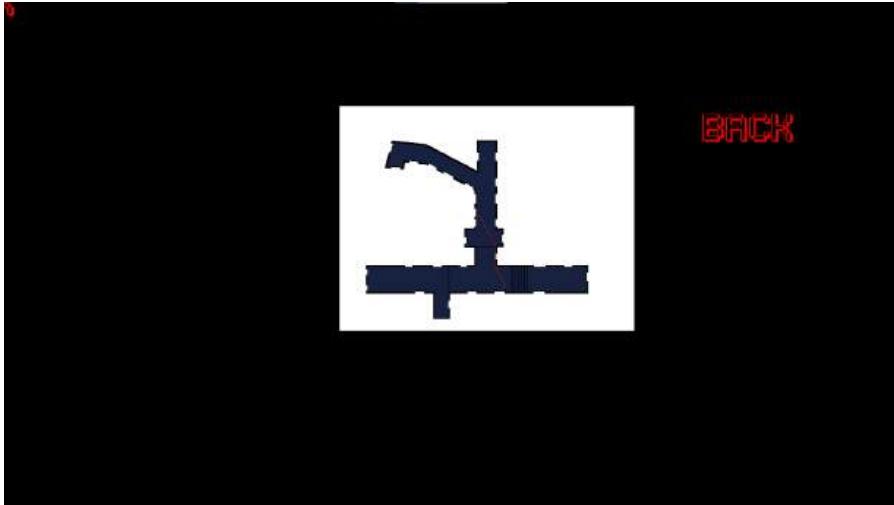
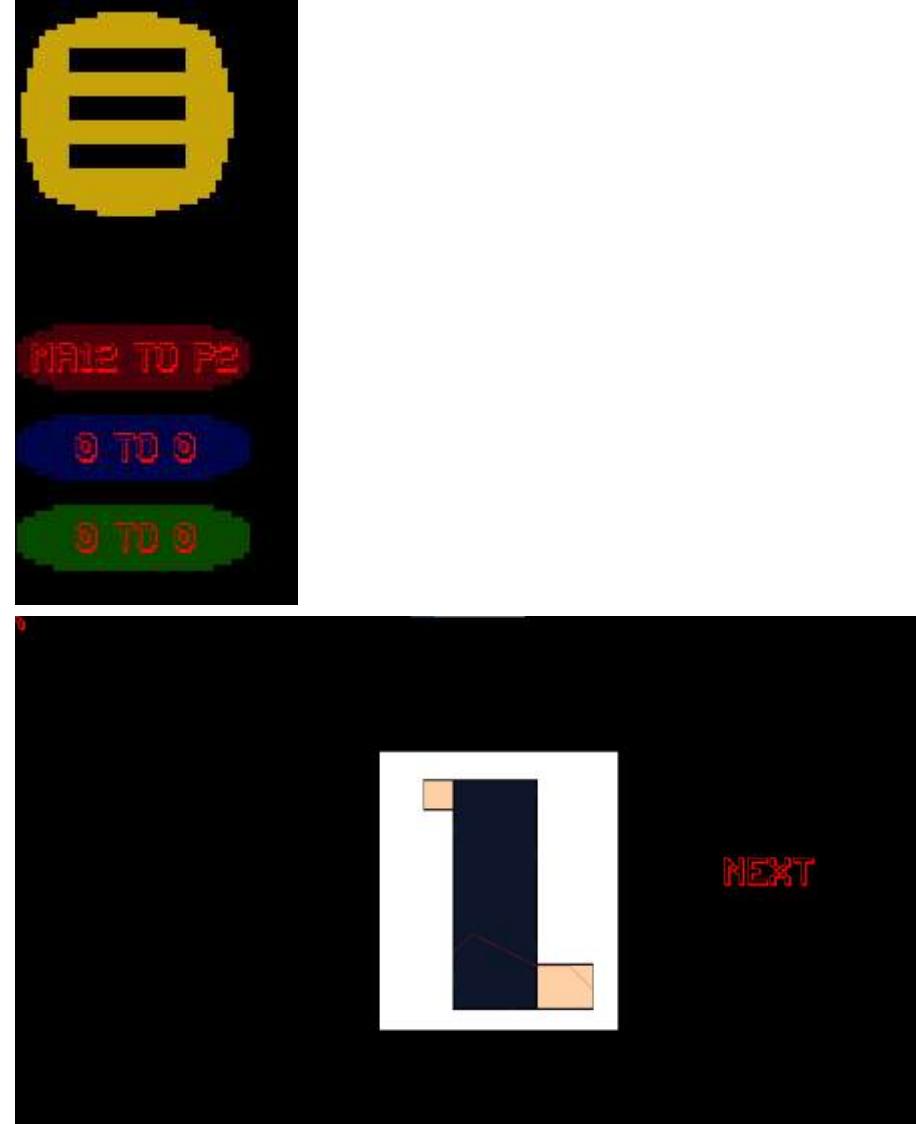


BACK

NEXT

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                         |      |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|     |                                                                                                                                                                                                                                                                                                                                                       |      |
| PT6 | <p>PLEASE CLICK TO ENTER YOUR LOCATION HERE</p> <input type="text" value="MA12"/> <p>SEARCH</p> <p>LIKELY RESULT 1: MA12 ●</p> <p>LIKELY RESULT 2: MA11 ●</p> <p>LIKELY RESULT 3: MA13 ●</p> <p>LIKELY RESULT 4: MA14 ●</p> <p>PLEASE CLICK TO ENTER YOUR DESTINATION HERE</p> <input type="text" value="P2"/> <p>SEARCH</p> <p>LIKELY RESULT 1: P2 ●</p> <p>LIKELY RESULT 2: ●</p> <p>LIKELY RESULT 3: ●</p> <p>LIKELY RESULT 4: ●</p> | Pass |

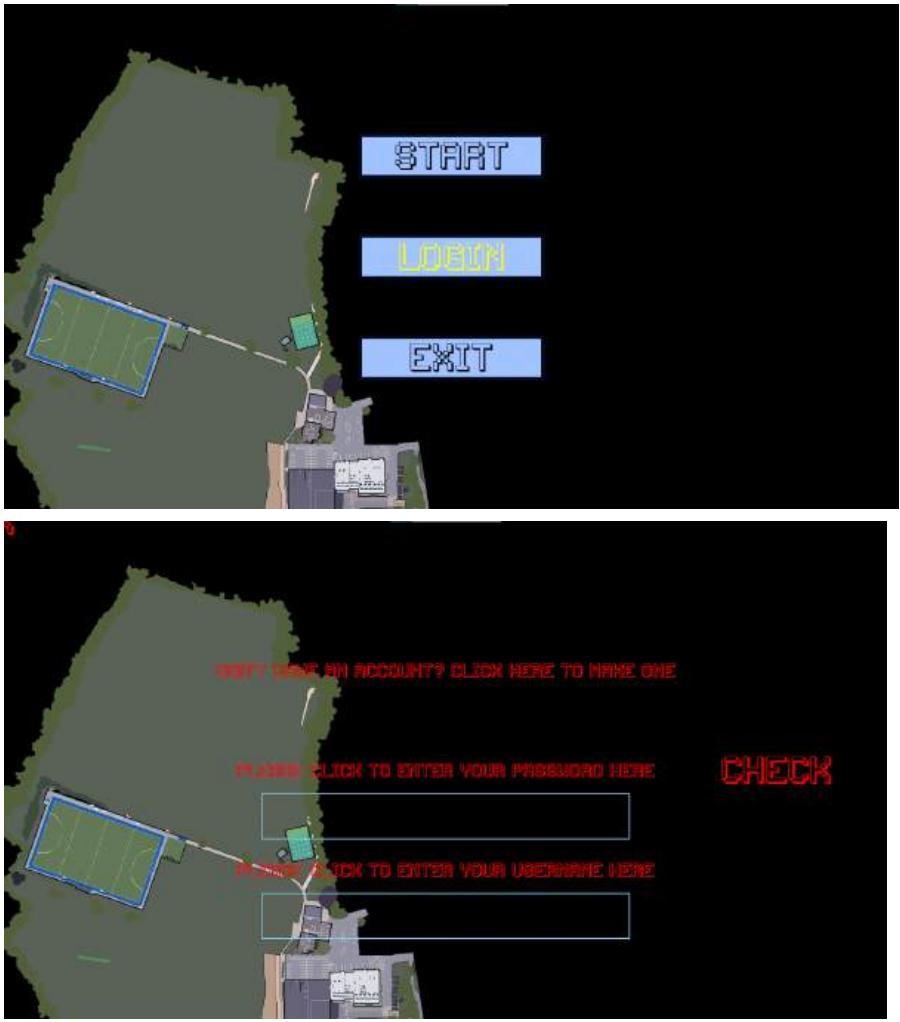


|     |                                                                                     |      |
|-----|-------------------------------------------------------------------------------------|------|
|     |   |      |
| PT7 |  | Pass |



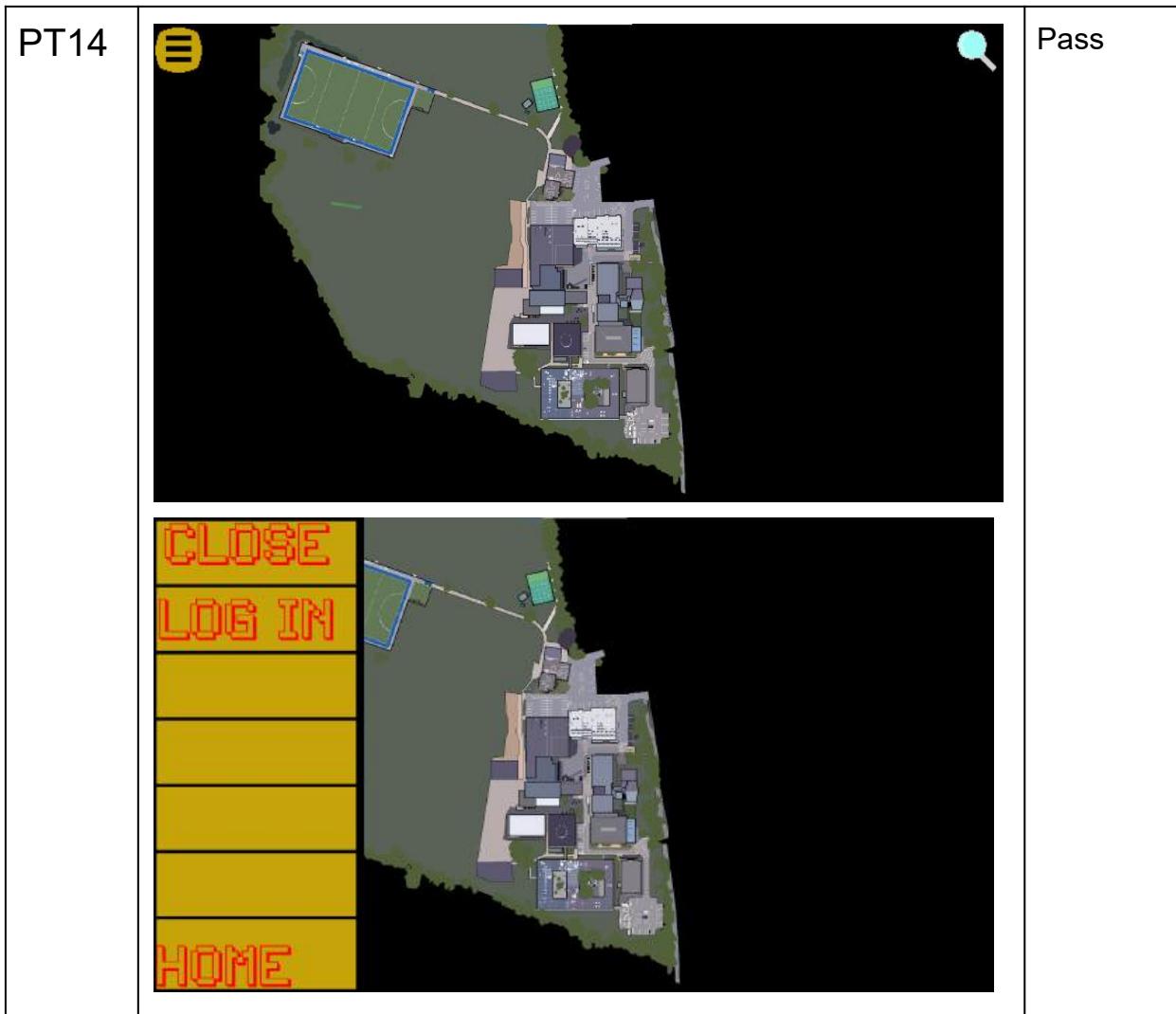
|     |                                                                                     |      |
|-----|-------------------------------------------------------------------------------------|------|
| PT8 |  | Pass |
|-----|-------------------------------------------------------------------------------------|------|

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                         |      |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|     | <p>A screenshot of a search interface. At the top, there is a text input field containing "MA12" and a button labeled "SEARCH". Below the search bar, the text "PLEASE CLICK TO ENTER YOUR LOCATION HERE" is displayed. A list titled "LIKELY RESULT" shows four entries: "LIKELY RESULT 1: MA12" with a red dot, "LIKELY RESULT 2: MA11" with a green dot, "LIKELY RESULT 3: MA13" with a green dot, and "LIKELY RESULT 4: MA14" with a green dot.</p> |      |
| PT9 | <p>Two screenshots of a map interface for PT9. The top screenshot shows a simplified map with a green area, a blue rectangular building, and three blue buttons labeled "START", "LOGIN", and "EXIT". The bottom screenshot shows a more detailed map with various buildings, roads, and icons like a magnifying glass and a menu icon.</p>                                                                                                             | Pass |

|      |                                                                                     |      |
|------|-------------------------------------------------------------------------------------|------|
| PT10 |   | Pass |
| PT11 |  | Pass |

|      |                                                                                   |      |
|------|-----------------------------------------------------------------------------------|------|
| PT12 |  | Pass |
|------|-----------------------------------------------------------------------------------|------|

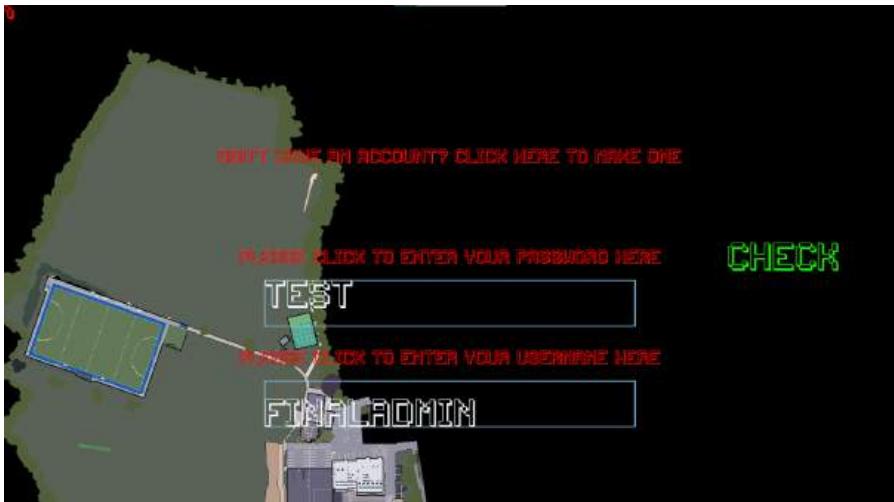
|      |                                                                                     |      |
|------|-------------------------------------------------------------------------------------|------|
| PT13 |  | Pass |
|------|-------------------------------------------------------------------------------------|------|



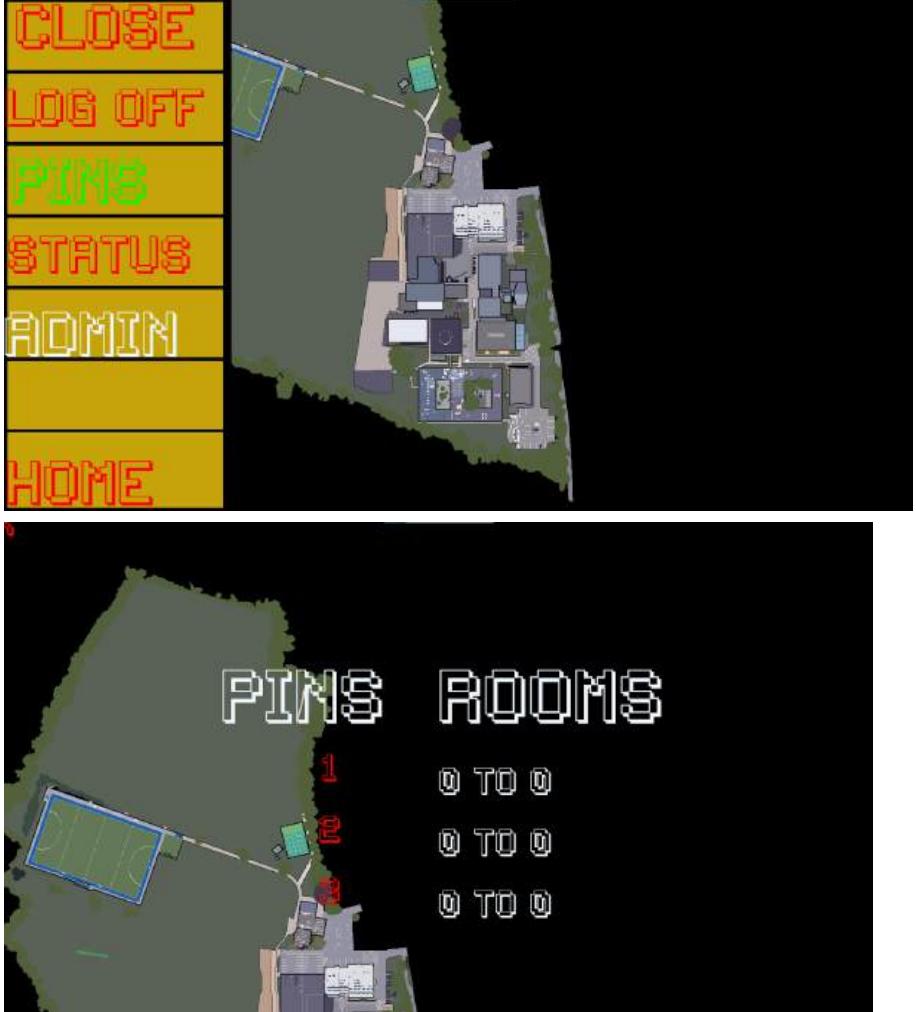
PT15

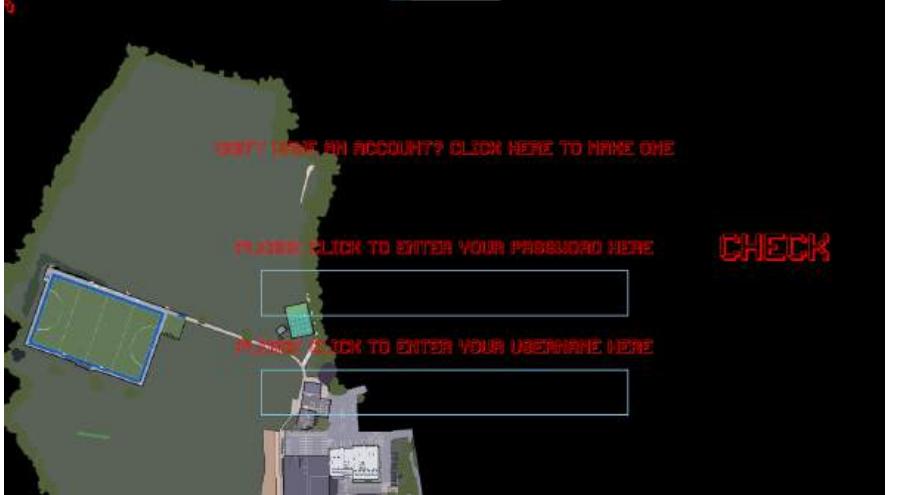


Pass

|      |                                                                                                                                                                           |      |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PT16 | <br> | Pass |
| PT17 |                                                                                       | Pass |

|      |                                                                                                                                                                                                                                                  |      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|      |                                                                                                                                                                |      |
| PT18 | <p>CLOSE<br/>LOG OFF<br/>PINS<br/>STATUS<br/>ADMIN<br/><br/>HOME</p> <br> | Pass |

|      |                                                                                      |      |
|------|--------------------------------------------------------------------------------------|------|
| PT19 |   | Pass |
| PT20 |  | Pass |

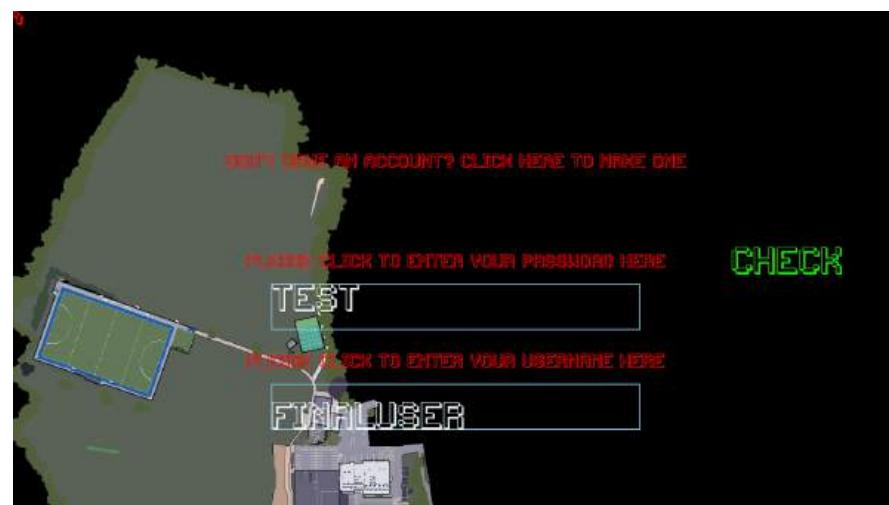
|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|      |  <p>A screenshot of a web-based search interface. At the top, there is a text input field with the placeholder "PLEASE CLICK TO ENTER YOUR LOCATION HERE". To the right of the input field is a blue "SEARCH" button. Below the input field, the text "LIKELY RESULT 1" is followed by four small red circular icons. Below each icon is a line of text: "LIKELY RESULT 2", "LIKELY RESULT 3", and "LIKELY RESULT 4".</p>                       |      |
| PT21 |  <p>A screenshot of a map-based application. On the left side, there is a vertical navigation bar with yellow buttons labeled "CLOSE", "LOG IN", and "HOME". The main area shows a detailed map of a campus or facility with various buildings and green spaces. A blue dashed rectangle highlights a specific area on the map.</p>                                                                                                            | Pass |
|      |  <p>A screenshot of a login interface. At the top, it asks "DON'T HAVE AN ACCOUNT? CLICK HERE TO MAKE ONE". Below this, there are two text input fields: one for "PLEASE CLICK TO ENTER YOUR PASSWORD HERE" and another for "PLEASE CLICK TO ENTER YOUR USERNAME HERE". To the right of the password field is a blue "CHECK" button. The background features a map of a campus with a blue dashed rectangle highlighting a specific area.</p> |      |

|      |                                                                                                                                                                                                                                                                                                                                                                                   |      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PT22 | <p>A screenshot of a map interface. On the left, there is a vertical menu bar with yellow buttons containing red text. The buttons are labeled: CLOSE, LOG OFF, PINS, STATUS, ADMIN, and HOME. To the right of the menu is a map showing a green area with a blue boundary, a blue building, and a grey building complex. A white line with arrows indicates a path or route.</p> | Pass |
|      | <p>A screenshot of a map interface. In the center, there is a black rectangular panel with three blue buttons labeled: START, LOGIN, and EXIT. The background shows a map with a green area, a blue building, and a grey building complex. A white line with arrows indicates a path or route.</p>                                                                                |      |
|      | <p>A screenshot of a map interface. On the left, there is a yellow circular icon with three horizontal lines. On the right, there is a blue magnifying glass icon. The background shows a map with a green area, a blue building, and a grey building complex. A white line with arrows indicates a path or route.</p>                                                            |      |

|      |                                                                                                                                                                         |               |      |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------|
| PT23 | <br> | BRICK<br>NEXT | Pass |
| PT24 |                                                                                      | BRICK         | Pass |

|      |                                                                                                                                                                          |      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|      |       |      |
| PT25 |   | Pass |

PT26



Pass



PT27



Pass

|      |                                                                                     |      |
|------|-------------------------------------------------------------------------------------|------|
|      | <p>THE PASSWORD OR USERNAME IS INCORRECT PLEASE TRY AGAIN</p>                       |      |
| PT28 |  | Pass |

|      |                                                                                                                                                                           |      |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PT29 | <br> | Pass |
| PT30 |                                                                                       | Pass |

|      |                                                                                                                                                                                                                                                                      |      |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|      | <p>The AccessKey entered is incorrect. Note that an access key is not necessary to make an account.</p>                                                                                                                                                              |      |
| PT31 | <p>General Sharing Security Previous Versions Customize</p> <p> SjwmsMaps</p> <hr/> <p>Type: File folder</p> <p>Location: [REDACTED]</p> <p>Size: 1.88 MB (1,974,884 bytes)</p> <p>Size on disk: 2.11 MB (2,220,032 bytes)</p> <p>Contains: 115 Files, 0 Folders</p> | Pass |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| PT32 |  Rooms - Notepad<br>File Edit Format View Help<br>S1 68 450 Green S_EX1<br>S2 68 334 Green S_EX1<br>S4 66 107 Green S_EX1<br>S5 111 61 Green S_EX1<br>S6 179 95 Green S_EX1<br>S7 274 93 Green S_EX2<br>S8 353 92 Green S_EX2<br>S9 356 158 Green S_EX2<br>S10 517 335 Green S_EX2<br>S11 356 256 Green S_EX2<br>S12 402 384 Green S_EX2<br>S13 306 356 Green S_EX1<br>S14 144 335 Green S_EX1<br>S15 577 123 Green S_EX2<br>S16 667 576 Green S_EX2<br>604 441 67 Green LB_ST2<br>605 505 89 Green LB_ST2<br>606 458 187 Green LB_ST2<br>607 459 281 Green LB_ST2<br>608 475 347 Green LB_ST2<br>609 448 360 Green LB_ST2<br>610 345 292 Green LB_ST1<br>611 257 292 Green LB_ST1<br>612 174 292 Green LB_ST1<br>613 161 262 Green LB_ST1<br>614 232 262 Green LB_ST1<br>615 428 141 Green LB_ST2<br>E1 171 400 Green EB_EX1<br>E2 58 390 Green EB_EX1<br>E3 59 356 Green EB_EX1<br>E4 149 343 Green EB_EX1<br>E5 212 343 Green EB_EX1<br>E6 429 400 Green EB_EX1<br>E7 533 336 Green EB_EX1<br>E8 491 343 Green EB_EX1<br>E9 430 343 Green EB_EX1<br>E10 267 343 Green EB_EX1<br>P1 291 252 Green EB_EX1<br>P2 292 229 Green EB_EX1<br>P3 291 200 Green EB_EX1<br>P4 235 149 Green EB_EX1<br>P5 200 130 Green EB_EX1<br>P6 159 121 Green EB_EX1<br>P7 122 133 Green EB_EX1<br>P9 337 197 Green EB_EX1<br>MA01 162 166 Green M_E1<br>MA02 162 228 Green M_EX1<br>MA03 82 165 Green M_E1<br>MA04 82 302 Green M_EX1<br>MA05 82 343 Green M_EX1<br>MA11 158 480 Green M_ST1<br>MA12 158 422 Green M_ST1<br>MA13 158 164 Green M_ST2<br>...<br>MA11 158 480 Green M_ST1<br>MA12 158 422 Green M_ST1<br>MA13 158 164 Green M_ST2<br>MA14 331 85 Green M_ST2<br>MA15 330 216 Green M_ST2<br>MA16 330 419 Green M_ST1<br>11 59 123 Green C_ST2<br>12 157 107 Green C_ST2<br>13 261 108 Green C_ST2<br>14 261 224 Green C_ST2<br>15 111 236 Green C_ST2<br>16 89 211 Green C_ST2<br>21 31 137 Green G_ST2<br>22 121 120 Green G_ST2<br>23 233 123 Green G_ST2<br>24 232 232 Green G_ST2<br>25 81 250 Green G_ST2<br>26 61 225 Green G_ST2<br>DT1 206 178 Green DT_E1<br>DT2 206 124 Green DT_E1<br>DT3 510 189 Green DT_EX1<br>DT4 492 145 Green DT_ST2<br>DT5 491 303 Green DT_ST2<br>SEN 510 103 Green DT_EX1<br>PrepRoom 684 59 Green DT_EX1<br>TechRoom 713 93 Green DT_EX1<br>DTLab 684 188 Green DT_EX1<br>A1 103 257 Green DT_ST2<br>A2 100 190 Green DT_ST2<br>MA13 158 164 Green M_ST2 | Pass |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|

|  |  |
|--|--|
|  |  |
|--|--|

## **Usability features:**

After completing the beta testing I sent a copy of the application to my stakeholders: Alvin Roy ,Joseph Baker and Mr Cooker , so that they themselves can test the product and I could see how usable the product is by getting them to fill in the questionnaire below afterwards.

| <b><u>Questionnaire:</u></b> |                                                                                                             |                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------|-------------------------|
| <b>Number</b>                | <b>Question</b>                                                                                             | <b>Success Criteria</b> |
| <b>1</b>                     | Did you find navigating the application difficult?                                                          | SC1,SC2,SC3             |
| <b>2</b>                     | Was the route displayed clear to follow?                                                                    | SC3,SC8,SC9,SC10        |
| <b>3</b>                     | Was the route displayed efficient?                                                                          | SC8,SC9,SC10            |
| <b>4</b>                     | Were you able to successfully store ,retrieve and use saved data after the application had been closed?     | SC8,SC9,SC10,SC14,SC15  |
| <b>5</b>                     | If you ran into an error did an error screen pop up and if so was it able to help you correct your mistake? | SC16                    |

| <b><u>Questionnaire Feedback:</u></b> |               |                                                                                                                                                                                                                                                                                              |
|---------------------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Number</b>                         | <b>Person</b> | <b>Answer</b>                                                                                                                                                                                                                                                                                |
| <b>1</b>                              | Alvin Roy     | I found navigating the application rather easy due to buttons being placed at sensible positions and not blending in with the background.                                                                                                                                                    |
|                                       | Joseph Baker  | Although navigating the application the majority of the time was fine when it came to going between the search screens, sometimes it was confusing as I was sometimes unsure whether I was entering the start location or the destination as there's very little change between the screens. |
|                                       | Mr Cooker     | The size of the buttons and names made it obvious how the application was supposed to be navigated.                                                                                                                                                                                          |
| <b>2</b>                              | Alvin Roy     | The displayed route was clear in most cases when on the inside but sometimes on the outside map of the site the route would be difficult to see as the map of the site is significantly bigger in contrast to the other maps.                                                                |

|   |              |                                                                                                                                                                                                                                                                                                                                                         |
|---|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Joseph Baker | The colour red made the route very clear as it contrasted the background.                                                                                                                                                                                                                                                                               |
|   | Mr Cooker    | I would say that the route was clear for the most part however it was noticeably difficult too see on the main map due to the size difference between the path and the map.                                                                                                                                                                             |
| 3 | Alvin Roy    | In most cases within buildings the route was efficient however when travelling from buildings to buildings sometimes the path would choose a further exit rather than a closer one making the journey longer than it needed to be.                                                                                                                      |
|   | Joseph Baker | I found the route to be efficient except sometimes for buildings with multiple entrances as the application would choose the further entrance rather than the closer one.                                                                                                                                                                               |
|   | Mr Cooker    | The path displayed was efficient as it even considered the one way system around parts of the school however sometimes for areas with multiple entrances the further entrance would be chosen.                                                                                                                                                          |
| 4 | Alvin Roy    | I was able to successfully use the pins in order to store a route and use it.                                                                                                                                                                                                                                                                           |
|   | Joseph Baker | I found no difficulties in storing and retrieving and using my pins.                                                                                                                                                                                                                                                                                    |
|   | Mr Cooker    | At first I was confused when trying to store the pin as I believed that you needed to click the pin in order to store a value but the error screen made it clear that was not the case ,however the rest of the process with the pins went alright.                                                                                                     |
| 5 | Alvin Roy    | I didn't run into any errors so I am unable to answer this.                                                                                                                                                                                                                                                                                             |
|   | Joseph Baker | I ran into an error when entering my details when creating an account because as the application pointed out I entered an incorrect access key when one wasn't needed in the first place to create my account.                                                                                                                                          |
|   | Mr Cooker    | I ran into an error at first due to the fact that I clicked upon an empty pin as I thought that I had to click on the pin in order to store a value but after seeing the error screen it was made evident that I was wrong, but although the error screen corrected me in the fact that I needed to store the pin first it did not tell me exactly how. |

### Possible Improvements:

| Improvement                                 | Explanation                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add faded background text to search screens | Within the first question of the questionnaire I was made aware by one of my clients that it was difficult to differentiate between the search screens so to make it easier I could add a faded background text in order for the user to be able to identify the search screen they're on without obstructing anything else. |
| Increases size of the path on the main map  | Within the second question of the questionnaire it was identified that the size                                                                                                                                                                                                                                              |

|                                                                                   |                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                   | of the main map sometimes made it difficult to see the path as it was small in comparison so I could simply increase the size of the path when on the main map although this might slightly reduce its efficiency.                                                                                                           |
| Change code so it goes to the closest entrance of the destination room's building | From question three of the questionnaire I was made aware that the choice to have each entrance to a building fixed reduced the efficiency of the path as there were more optimal entrances so instead I should have made the code go through the closest entrance in order to increase the efficiency of the path more.     |
| Edit pin error screen to direct user to where to store pins                       | One of my clients made me aware of the possibility of a user misunderstanding how to store pins through their response in the fourth question of the questionnaire so in order to make it clearer to the user I could edit the error screen in order to also direct the user to the correct location of where to store pins. |

## Evaluation:

| <b><u>Success Criteria CheckList</u></b> |                                                                                                                                                                                                                                                      |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Success Criteria</b>                  | <b>How well program matches the requirement</b>                                                                                                                                                                                                      |
| SC1                                      | The program matches this success criteria perfectly as shown through <b>PT9</b> <b>PT10</b> and <b>PT11</b> as they display a starting screen with the necessary buttons and functions present and working in order for the application to function. |
| SC2                                      | As shown in <b>PT10</b> the program has the capabilities to allow the user to exit and thus matches this success criteria perfectly.                                                                                                                 |
| SC3                                      | As shown through the majority of the PTs the program is capable of swapping between screens and thus matches this success criteria perfectly.                                                                                                        |
| SC4                                      | As shown in <b>PT26</b> and <b>PT27</b> the program has the capabilities to allow the                                                                                                                                                                |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | user to log in to accounts of any level and thus matches this success criteria perfectly.                                                                                                                                                                                                                                                                                                                                                                                   |
| SC5  | As shown in <b>PT22</b> the program has the capabilities to allow the user to log off to accounts and thus matches this success criteria perfectly.                                                                                                                                                                                                                                                                                                                         |
| SC6  | The program matches this success criteria perfectly as it's capable of storing and creating authorised accounts as shown in <b>PT29</b>                                                                                                                                                                                                                                                                                                                                     |
| SC7  | The program matches this success criteria perfectly as it's capable of storing and creating accounts as shown in <b>PT28</b> and <b>PT29</b>                                                                                                                                                                                                                                                                                                                                |
| SC8  | The program matches this success criteria well as it can locate the user inputted destination as shown in <b>PT4,PT5,PT6</b> and <b>PT7</b> , but it could be more precise if the map used had a higher resolution.                                                                                                                                                                                                                                                         |
| SC9  | The program matches this success criteria well for the same reasons as stated in <b>SC8</b> .                                                                                                                                                                                                                                                                                                                                                                               |
| SC10 | The program matches this success criteria decently as although the choice to make the path red helps it stand out by contrasting the map ,as shown in <b>PT4,PT5,PT6</b> and <b>PT7</b> , the questionnaire responses reveal that while on the main map the choice in colour is mitigated as the path is significantly smaller in comparison to the size of the map.                                                                                                        |
| SC11 | The program matches this success criteria decently as although it contains all rooms on the site ,as shown in <b>PT32</b> , it doesn't contain general areas such as the main hall or the sixth formcentre which contains other rooms.                                                                                                                                                                                                                                      |
| SC12 | The program matches this success criteria decently as although the program can show if a room is available ,slightly busy or not available at all through the use of the colours green yellow and red as shown in <b>PT8</b> , it does not give a detailed report of the status such as a reason why it may be unavailable.                                                                                                                                                 |
| SC13 | The program matches this success criteria perfectly as it's able to successfully display the status of a room as shown in <b>PT8</b>                                                                                                                                                                                                                                                                                                                                        |
| SC14 | The program matches this success criteria perfectly as it's able to successfully change the status of a room as shown in <b>PT8</b>                                                                                                                                                                                                                                                                                                                                         |
| SC15 | The program matches this success criteria perfectly as it's able to successfully load and display the pins saved by the user as shown in <b>PT3</b>                                                                                                                                                                                                                                                                                                                         |
| SC16 | The program matches this success criteria decently as although I was able to include messages to attempt to direct the user in to the method to fix their error in the form of error screens ,as shown in <b>PT27</b> and <b>PT30</b> , within the fifth question of the questionnaire it was made evident that I wasn't able to anticipate all errors that the user could possibly make and thus not be able to give a broad enough description on how to fix their error. |
| SC17 | The program matches this success criteria perfectly as the program takes 1.88 MB as shown in <b>PT31</b>                                                                                                                                                                                                                                                                                                                                                                    |
| SC18 | The program matches this success criteria not too well as although the files the program is stored in can be compressed into an application through an                                                                                                                                                                                                                                                                                                                      |

|  |                                                                                                                   |
|--|-------------------------------------------------------------------------------------------------------------------|
|  | emulator to be used on a mobile device I haven't directly made it into an application for my clients to download. |
|--|-------------------------------------------------------------------------------------------------------------------|

| <b><u>Additional Features:</u></b>                                     |                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Feature</b>                                                         | <b>Usefulness</b>                                                                                                                                                                                                                                                                                                                                    | <b>Approach to code</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Add a help button in the menu explaining clearly how to use the system | One of my clients made me aware of the possibility of a user misunderstanding how to store pins through their response in the fourth question of the questionnaire so in order to make it clearer to the user I could make a help button in order to explain clearly to the user what each button is supposed to be used for improving <b>SC16</b> . | I'd create a help button through the use of the button class and then add it within the menu screen function in order for the button to be displayed in the menu. I would then use an if statement so that upon the help button being clicked it would run a new function called the Help Screen function which simply screenblit text detailing the functions each button does.                                                                                            |
| Add option to see path on black and white layer                        | The general consensus gained from the second question of the questionnaire was that the size of the main map sometimes made it difficult to see the path as it was small in comparison so to improve it I could add option to grayscale the main map in order to make it more clear where the path is on the map improving <b>SC10</b> .             | I would first have to edit the code so that the path is also drawn onto a grayscale version of the map as well as the coloured version and then I'd use the button class inorder to have an image of a coloured map on top of a blank map a button ,so that the user could infer the function of the button. I'd then add it with the path functions and have an if statement so that upon the button being clicked it would screen blit the blank map with the path on it. |
| Add a zoom in and out function                                         | The general consensus gained from the second question of the questionnaire was that the size of the main map sometimes made it difficult to see the path as it was small in comparison so to fix                                                                                                                                                     | I'd use the button class in order to make two buttons with one button being a magnifying glass with a minus sign and the other being a magnifying glass with the plus sign. I'd add both of the buttons to the                                                                                                                                                                                                                                                              |

|  |                                                                                                   |                                                                                                                                                                                                                                 |
|--|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | it I could add a zoom in and out function to mitigate the size difference improving <b>SC10</b> . | main screen and the path screens and use an if statement so that upon the magnifying glass with the plus sign being clicked it would zoom in and upon the magnifying glass with the minus sign being clicked it would zoom out. |
|--|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## **Maintenance:**

The program is ordered in a logical sequence in order to make maintaining and editing it easier with the pygame setup coming first so that it's easy to edit if the code needs to be updated in order to meet the new python requirements and the data, which is indented in order to clearly refer to what they're used for. Following this are the global variables, so that they can be quickly changed as details of the site change over time, with the variables being named appropriately in order for their use to be easily identified as displayed below:

| Variable           | Code Notation   | Description                                                                                                             |
|--------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| Grayscale Img      | ImgName_G       | Stores the black and white version of a map in order to be used when creating a path with the path creation function    |
| Coloured Img       | ImgName_C       | Stores the coloured version of a map in order to be used when drawing the path using the path creation function         |
| Permanent Img      | ImgName_P       | Stores a permanent coloured version of a map in order to be used to revert the coloured Img back to its original state. |
| Backgrounds        | ImgName_Img     | Stores backgrounds to be used with the screens.                                                                         |
| External Gui Image | ImgName_Image   | Stores gui images to be used within the screens                                                                         |
| TextFile           | DataStored_List | Stores the data on a text file within a list so that it can be retrieved                                                |

|                           |                   |                                                                                                                                                                       |
|---------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Building's Rooms          | Building_Nodes    | Stores the rooms within a building so that the path creation function can create a path within and outside the building                                               |
| Building Stairwell Coords | Building_ST       | Stores the coords of the entrance and exit of the stairwell in order to be used in the instance a room is not on the ground floor of a building when creating a path. |
| Building Exit Coords      | Building_Exit     | Stores the coords of the exit of the building in order to be used when creating a path inside the building                                                            |
| Building Entrance Coords  | Building_Entrance | Stores the coords of the entrance of the building in order to be used when creating a path inside the building                                                        |
| Button Display            | ButtonName_Button | Stores the information needed to create and display a button.                                                                                                         |

The data is then followed by screen functions which are named appropriately in order to make it clear what function refers to what screen as shown below:

| ScreenName | Description                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Start_Sc   | The first screen that is displayed when the program opens containing the start ,login and exit functions                            |
| LogIn_Sc   | Screen which enables the user to log into an account                                                                                |
| SignUp_Sc  | Screen which enables the user to sign out of an account                                                                             |
| Main_Sc    | Screen which links to the main functions of the program being the search,pin and menu functions.                                    |
| Menu_Sc    | Screen which displays a side menu over the main sc in order to link to other screens containing sub program algorithms to edit data |
| Status_Sc  | Screen that enables the status of a room to                                                                                         |

|                      |                                                                                                                                   |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|                      | be changed by the user                                                                                                            |
| Pin_Sc               | Screen that enables the user to edit the paths stored within theirs pins                                                          |
| PinSearch_Sc         | Screen that enables the user to search for the start room of the path they want to store                                          |
| PinSearch2_Sc        | Screen that enables the user to search for the destination room of the path they want to store                                    |
| LocationSearch_Sc    | Screen that enables the start room of the user's desired path to be searched for                                                  |
| DestinationSearch_Sc | Screen that enables the destination room of the user's desired path to be searched for                                            |
| StatusSearch_Sc      | Screen that enables the a room to be searched for in order for the user to find it's status                                       |
| Path1_Sc             | Screen that displays the first section of the path within the building of the starting room                                       |
| Path1_5_Sc           | Screen that displays the path the user must take if the start room is within a building with stairs                               |
| Path1_5B_Sc          | Screen that displays the path the user must take if the start room is within a building with stairs                               |
| Path1_5C_Sc          | Screen that displays the path the user must take if the start room is within a building with stairs and on the second floor       |
| Path2_Sc             | Screen that displays the path in between the buildings of the start room and destination room                                     |
| Path2_5_Sc           | Screen that displays the path the user must take if the destination room is within a building with stairs                         |
| Path2_5B_Sc          | Screen that displays the path the user must take if the destination room is within a building with stairs                         |
| Path2_5C_Sc          | Screen that displays the path the user must take if the destination room is within a building with stairs and on the second floor |
| Path3_Sc             | Screen that displays the final section of the                                                                                     |

|           |                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
|           | path within the building of the destination room                                                                         |
| False_Sc  | Screen that tells the user if the user entered the wrong login details when trying to sign in                            |
| False2_Sc | Screen that tells the user if the access key entered when making an account is incorrect                                 |
| False3_Sc | Screen that corrects the user if the username or password entered when making an account does not match the requirements |
| False4_Sc | Screen that tells the user if the inputted location is not found within the text files                                   |
| False5_Sc | Screen that tells the user if an empty result is picked when searching for rooms                                         |
| False6_Sc | Screen that tells the user if an empty pin is selected                                                                   |
| Error_Sc  | Screen that tells the user if an unknown error occurs in the system                                                      |

and contains indented comments within in order to make it clear what procedure does what to the screen in case the screen needs to be edited and what function links to a sub program algorithm or screen. Afterwards the screen functions are followed by the sub program algorithms which are also indented in order to explain clearly what each step of the function does to the data so that it can be edited with ease.

| <b><u>Limitations:</u></b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Limitation</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <b>Possible solution</b>                                                                                                                                                                                                                                                                                                        |
| The biggest limitation of the program's ability to solve the problem of guiding users across the site while familiarising them with the site as mentioned before is the program's inability to receive real time data as this prevents the program from displaying a truly accurate path . For example the program is unable to locate the exact current location of the user and instead asks for the user to enter a starting room which prevents the program from displaying a truly accurate path as the path created is not from the user's actual position. | With more development time it would be possible for the code to integrate the use of an online server to store ,retrieve and update data in real time and thus also make other possible functions liable such as a gps due to less data being stored on the users device and thus leaving room for other functions to be added. |
| As displayed in the questionnaire and mentioned in the success criteria checklist the program is unable to anticipate every possible user error and thus its ability to correct inputs is limited.                                                                                                                                                                                                                                                                                                                                                                | A combination of regular testing and updates for error screens for possible user errors could increase the chance of the program being able to successfully correct the user's input.                                                                                                                                           |

## Final Code

```

import pygame, sys # Pygame is the Library we're using to display the maps upon and sys lets us access system-specific parameters and functions
import math, numpy as np # To do calculations needed for A* Path finding algorithm
import cv2 ,csv #To get the matrix needed for A* Algorithm to be used
import time
from queue import PriorityQueue
import json#To be able to read text files

-----Pygame Setup-----##
pygame.init() #initializes pygame
info = pygame.display.Info() #gets info about user's screen
Width = info.current_w #screen's width
Height = info.current_h# screen's height
Center_X = Width/2#X cords for the center of the screen
Center_Y = Height/2#Y cords for the center of the screen
Screen = pygame.display.set_mode((Width,Height)) #app now fit's to user's screen
clock = pygame.time.Clock()

-----Data-----##
#Grayscale Backgrounds:
MainMap_G = cv2.imread("MainMapBlank.jpg")
ComputingBlock_G = cv2.imread("ComputingBlockBlank.jpg")
GeographyBlock_G = cv2.imread("GeographyBlockBlank.jpg")
ScienceBlock_G = cv2.imread("ScienceBlockBlank.jpg")
MathsBlockDownstairs_G = cv2.imread("MathsDownstairsBlank.jpg")
MathsBlockUpstairs_G = cv2.imread("MathsUpstairsBlank.jpg")
DTBlockDownstairs_G = cv2.imread("DtBlockDownstairsBlank.jpg")
DTBlockUpstairs_G = cv2.imread("DtBlockUpstairsBlank.jpg")
EnglishBlock_G = cv2.imread("EnglishBlockBlank.jpg")
LanguageBlock_G = cv2.imread("LanguageBlockBlank.jpg")
SixthFormCenter_G = cv2.imread("SixthFormCenterBlank.jpg")
SixthFormStairsA_G = cv2.imread("SixthFormStairsABlank.jpg")
SixthFormStairsB_G = cv2.imread("SixthFormStairsBBlank.jpg")
LanguageStairsFromA_G = cv2.imread("LanguageStairsFromABlank.jpg")
LanguageStairsFromB_G = cv2.imread("LanguageStairsFromBBlank.jpg")
DTStairsFrom_G = cv2.imread("DTStairsFromBlank.jpg")
MathsStairsFrom_G = cv2.imread("MathStairsBlank.jpg")
LanguageStairsToA_G = cv2.imread("LanguageStairsToABlank.jpg")
LanguageStairsToB_G = cv2.imread("LanguageStairsToBBlank.jpg")
DTStairsToA_G = cv2.imread("DTBlockToABlank.jpg")
DTStairsToB_G = cv2.imread("DTBlockToBBlank.jpg")
MathsStairsTo_G = cv2.imread("MathStairsBlank.jpg")
GeographyStairsTo_G = cv2.imread("GeographyStairsToBlank.jpg")
GeographyStairsFrom_G = cv2.imread("GeographyStairsFromBlank.jpg")
MainBlockFromA_G = cv2.imread("MainBlockFromABlank.jpg")
MainBlockFromB_G = cv2.imread("MainBlockFromBBlank.jpg")
MainBlockToA_G = cv2.imread("MainBlockToABlank.jpg")
MainBlockToB_G = cv2.imread("MainBlockToBBlank.jpg")

#PermaBackgrounds:
MainMap_P = cv2.imread("MainMapColour.jpg")
ComputingBlock_P = cv2.imread("ComputingBlockColor.jpg")
GeographyBlock_G = cv2.imread("GeographyBlockColor.jpg")
ScienceBlock_P = cv2.imread("ScienceBlockColour.jpg")
MathsBlockDownstairs_P = cv2.imread("MathsDownstairsColor.jpg")
MathsBlockUpstairs_P = cv2.imread("MathsUpstairsColor.jpg")
DTBlockDownstairs_P = cv2.imread("DtBlockDownstairsColor.jpg")
DTBlockUpstairs_P = cv2.imread("DtBlockUpstairsColor.jpg")
EnglishBlock_P = cv2.imread("EnglishBlockColor.jpg")
LanguageBlock_P = cv2.imread("LanguageBlockColour.jpg")
SixthFormCenter_P = cv2.imread("SixthFormCenterColor.jpg")
SixthFormStairsA_P = cv2.imread("SixthFormStairsAColor.jpg")
SixthFormStairsB_P = cv2.imread("SixthFormStairsBColor.jpg")
LanguageStairsFromA_P = cv2.imread("LanguageStairsFromAColor.jpg")
LanguageStairsFromB_P = cv2.imread("LanguageStairsFromBColor.jpg")
DTStairsFrom_P = cv2.imread("DTStairsFromColor.jpg")
MathsStairsFrom_P = cv2.imread("MathStairsColor.jpg")
LanguageStairsToA_P = cv2.imread("LanguageStairsToAColor.jpg")
LanguageStairsToB_P = cv2.imread("LanguageStairsToBColor.jpg")
DTStairsToA_P = cv2.imread("DTBlockToAColor.jpg")
DTStairsToB_P = cv2.imread("DTBlockToBColor.jpg")
MathsStairsTo_P = cv2.imread("MathStairsColor.jpg")
GeographyStairsTo_P = cv2.imread("GeographyStairsToColor.jpg")
GeographyStairsFrom_P = cv2.imread("GeographyStairsFromColor.jpg")
MainBlockFromA_P = cv2.imread("MainBlockFromAColor.jpg")
MainBlockFromB_P = cv2.imread("MainBlockFromBColor.jpg")
MainBlockToA_P = cv2.imread("MainBlockToAColor.jpg")
MainBlockToB_P = cv2.imread("MainBlockToBColor.jpg")

```

```

#ChangingBackgrounds:
MainMap_C = cv2.imread("MainMapChainging.jpg")
ComputingBlock_C = cv2.imread("ComputingBlockChainging.jpg")
GeographyBlock_C = cv2.imread("GeographyBlockChainging.jpg")
ScienceBlock_C = cv2.imread("ScienceBlockChainging.jpg")
MathsBlockDownstairs_C = cv2.imread("MathsDownstairsChainging.jpg")
MathsBlockUpstairs_C = cv2.imread("MathsUpstairsChainging.jpg")
DTBlockDownstairs_C = cv2.imread("DtBlockDownstairsChainging.jpg")
DTBlockUpstairs_C = cv2.imread("DtBlockUpstairsChainging.jpg")
EnglishBlock_C = cv2.imread("EnglishBlockChainging.jpg")
LanguageBlock_C = cv2.imread("LanguageBlockChainging.jpg")
SixthFormCenter_C = cv2.imread("SixthFormCenterChainging.jpg")
SixthFormStairsA_C = cv2.imread("SixthFormStairsAChainging.jpg")
SixthFormStairsB_C = cv2.imread("SixthFormStairsBChainging.jpg")
LanguageStairsFromA_C = cv2.imread("LanguageStairsFromAChainging.jpg")
LanguageStairsFromB_C = cv2.imread("LanguageStairsFromBChainging.jpg")
DTStairsFrom_C = cv2.imread("DTStairsFromChainging.jpg")
MathsStairsFrom_C = cv2.imread("MathStairsChainging.jpg")
LanguageStairsToA_C = cv2.imread("LanguageStairsToAChainging.jpg")
LanguageStairsToB_C = cv2.imread("LanguageStairsToBChainging.jpg")
DTStairsToA_C = cv2.imread("DTBlockToAChainging.jpg")
DTStairsToB_C = cv2.imread("DTBlockToBChainging.jpg")
MathsStairsTo_C = cv2.imread("MathStairsChainging.jpg")
GeographyStairsTo_C = cv2.imread("GeographyStairsToChainging.jpg")
GeographyStairsFrom_C = cv2.imread("GeographyStairsFromChainging.jpg")
MainBlockFromA_C = cv2.imread("MainBlockFromAChainging.jpg")
MainBlockFromB_C = cv2.imread("MainBlockFromBChainging.jpg")
MainBlockToA_C = cv2.imread("MainBlockToAChainging.jpg")
MainBlockToB_C = cv2.imread("MainBlockToBChainging.jpg")

#Backgrounds for pygame:
BackgroundMap_Img = pygame.image.load("MainBackground.png").convert()
MainMap_Img = pygame.image.load("MainMapChainging.jpg").convert()
Loading_Img = pygame.image.load("Loading.png").convert()
ComputingBlock_Img = pygame.image.load("ComputingBlockChainging.jpg").convert()
GeographyBlock_Img = pygame.image.load("GeographyBlockChainging.jpg").convert()
ScienceBlock_Img = pygame.image.load("ScienceBlockChainging.jpg").convert()
MathsBlockDownstairs_Img = pygame.image.load("MathsDownstairsChainging.jpg").convert()
MathsBlockUpstairs_Img = pygame.image.load("MathsUpstairsChainging.jpg").convert()
DTBlockDownstairs_Img = pygame.image.load("DtBlockDownstairsChainging.jpg").convert()
DTBlockUpstairs_Img = pygame.image.load("DtBlockUpstairsChainging.jpg").convert()
EnglishBlock_Img = pygame.image.load("EnglishBlockChainging.jpg").convert()
LanguageBlock_Img = pygame.image.load("LanguageBlockChainging.jpg").convert()
SixthFormCenter_Img = pygame.image.load("SixthFormCenterChainging.jpg").convert()
SixthFormStairsA_Img = pygame.image.load("SixthFormStairsAChainging.jpg").convert()
SixthFormStairsB_Img = pygame.image.load("SixthFormStairsBChainging.jpg").convert()
LanguageStairsFromA_Img = pygame.image.load("LanguageStairsFromAChainging.jpg").convert()
LanguageStairsFromB_Img = pygame.image.load("LanguageStairsFromBChainging.jpg").convert()
DTStairsFrom_Img = pygame.image.load("DTStairsFromChainging.jpg").convert()
MathsStairsFrom_Img = pygame.image.load("MathStairsChainging.jpg").convert()
LanguageStairsToA_Img = pygame.image.load("LanguageStairsToAChainging.jpg").convert()
LanguageStairsToB_Img = pygame.image.load("LanguageStairsToBChainging.jpg").convert()
DTStairsToA_Img = pygame.image.load("DTBlockToAChainging.jpg").convert()
DTStairsToB_Img = pygame.image.load("DTBlockToBChainging.jpg").convert()
MathsStairsTo_Img = pygame.image.load("MathStairsChainging.jpg").convert()
GeographyStairsTo_Img = pygame.image.load("GeographyStairsToChainging.jpg")
GeographyStairsFrom_Img = pygame.image.load("GeographyStairsFromChainging.jpg")
MainBlockFromA_Img = pygame.image.load("MainBlockFromAChainging.jpg")
MainBlockFromB_Img = pygame.image.load("MainBlockFromBChainging.jpg")
MainBlockToA_Img = pygame.image.load("MainBlockToAChainging.jpg")
MainBlockToB_Img = pygame.image.load("MainBlockToBChainging.jpg")

#GUI:
FilledRectangle = pygame.image.load("FilledRectangle.png")
Rectangle = pygame.image.load("Rectangle.png")
Pin1_Image = pygame.image.load("Pin1.png").convert()
Pin2_Image = pygame.image.load("Pin2.png").convert()
Pin3_Image = pygame.image.load("Pin3.png").convert()
Menu_Image = pygame.image.load("MenuB.png").convert()
Search_Image = pygame.image.load("SearchB.png").convert()
SideMenu_Image = pygame.image.load("SideMenu.png")
SideMenu_Image = pygame.transform.scale(SideMenu_Image, (int(Width/4), Height))
RedStatus_Image = pygame.image.load("RedLight.png")
GreenStatus_Image = pygame.image.load("GreenLight.png")
YellowStatus_Image = pygame.image.load("YellowLight.png")

```

```

#TextFiles:
User_List = []
Password_List = []
AccessKey_List = []
NumberOfPins_List = []
Pins_List = []
with open("Accounts.txt") as f:
 for line in f:#for each line in the file assign the account data to the dictionary accordingly
 (username,password,key,pin1_start,pin1_end, pin2_start,pin2_end,pin3_start, pin3_end) = line.split()
 User_List.append(username)
 Password_List.append(password)
 AccessKey_List.append(key)
 Pins_List.append(pin1_start)
 Pins_List.append(pin1_end)
 Pins_List.append(pin2_start)
 Pins_List.append(pin2_end)
 Pins_List.append(pin3_start)
 Pins_List.append(pin3_end)

Room_List = []
RoomX_List = []
RoomY_List = []
RoomStatus_List = []
RoomExit_List = []
with open("Rooms.txt") as f:
 for line in f:#for each line in the file assign the room data to the dictionary accordingly
 (RoomName, Room_X , Room_Y , RoomStatus ,Exit) = line.split()
 Room_List.append(RoomName)
 RoomX_List.append(Room_X)
 RoomY_List.append(Room_Y)
 RoomStatus_List.append(RoomStatus)
 RoomExit_List.append(Exit)

NodeName_List = []
NodeX_List = []
NodeY_List = []
with open("MainNodes.txt") as f:
 for line in f:#for each line in the file assign the node data to the dictionary accordingly
 (NodeName, Node_X , Node_Y) = line.split()
 NodeName_List.append(NodeName)
 NodeX_List.append(Node_X)
 NodeY_List.append(Node_Y)

-----Global Variables-----##
#User's Details:
UserAccount = ""
UserAccessKey = ""
UserPins = ["","","","","","",""]
#Log In Details:
Username_Type = False
Password_Type = False
LoggedIn = False
Username_Check = False
Password_Check= False
LoggedIn= False
AdminLoggedIn= False
AccessKey_Type = False
Username_Input = ""
Password_Input = ""
AccessKey_Input = ""
#Search Details:
Location_Type = False
Destination_Type = False
Pin1Searching = False
Pin2Searching = False
Pin3Searching = False
SearchedStatus = ""
RoomStatus = ""
StatusDisplay = ""
R1StatusDispaly = RedStatus_Image
R2StatusDispaly = RedStatus_Image
R3StatusDispaly = RedStatus_Image
R4StatusDispaly = RedStatus_Image
RepeatResult = False
Location_Input = ""
Result1_Input = ""
Result2_Input = ""
Result3_Input = ""
Result4_Input = ""
ActualLocation = ""
Destination_Input = ""
ActualDestination = ""

```

```

#PathFinder Variables
path_color = (0, 0, 255) #colour of the path
flag = 0
MapInfo = []
NodeInfo = []
pos1 = []
pos2 = []
#Outside Nodes
ScienceBlock_Nodes = ["S_EX1", "S_EX2"]
LanguageBlock_Nodes = ["LB_ST1", "LB_ST2"]
EnglishAndMusicBlock_Nodes = ["EB_EX1", "MB_EX1", "MB_EX2",]
MathsBlockDownstairs_Nodes = ["M_E1", "M_EX1"]
MathsBlockUpstairs_Nodes = ["M_ST1", "M_ST2"]
ComputingBlock_Nodes = ["C_ST1", "C_ST2"]
GeographyBlock_Nodes = ["G_ST1", "G_ST2"]
DTBlockDownstairs_Nodes = ["DT_E1", "DT_EX1"]
DTBlockUpstairs_Nodes = ["DT_ST1", "DT_ST2"]
SFC_EX1_Nodes =
["S_EX1", "S_EX2", "LB_ST1", "EB_EX1", "M_E1", "M_EX1", "M_ST1", "M_ST2", "C_ST1", "C_ST2", "G_ST1", "G_ST2", "DT_ST1", "DT_ST2"]
SFC_EX2_Nodes = ["DT_EX1", "MB_EX1", "MB_EX2"]
SFC_EX3_Nodes = ["LB_ST2"]
SixthFormCenter_Nodes = [SFC_EX1_Nodes, SFC_EX2_Nodes, SFC_EX3_Nodes]
Stair_Nodes = ["LB_ST1", "LB_ST2", "M_ST1", "M_ST2", "DT_ST1", "DT_ST2", "G_ST1", "G_ST2", "C_ST1", "C_ST2"]
#Stair Nodes
LBStairsFromA_Entrance = [26, 7]
LBStairsFromA_Exit = [63, 22]
LBStairsFromB_Entrance = [63, 100]
LBStairsFromB_Exit = [27, 104]
LBStairsToA_Entrance = [168, 153]
LBStairsToA_Exit = [139, 143]
LBStairsToB_Entrance = [171, 186]
LBStairsToB_Exit = [173, 117]

MathsStairs_Entrance = [14, 38]
MathsStairs_Exit = [249, 38]

DTStairsFromA_Entrance = [76, 184]
DTStairsFromA_Exit = [33, 165]
DTStairsToA_Entrance = [167, 92]
DTStairsToA_Exit = [36, 218]
DTStairsToB_Entrance = [168, 109]
DTStairsToB_Exit = [79, 88]

ComputingBlockStairsTo_Entrance = [2, 24]
ComputingBlockStairsTo_Exit = [124, 21]
ComputingBlockStairsFrom_Entrance = [int(4), int(19)]
ComputingBlockStairsFrom_Exit = [int(126), int(25)]

GeographyBlockStairsToA_Entrance = [2, 24]
GeographyBlockStairsToA_Exit = [2, 77]
GeographyBlockStairsFromA_Entrance = [2, 24]
GeographyBlockStairsFromA_Exit = [124, 21]
GeographyBlockStairsToB_Entrance = [4, 76]
GeographyBlockStairsToB_Exit = [126, 76]
GeographyBlockStairsFromB_Entrance = [126, 75]
GeographyBlockStairsFromB_Exit = [126, 25]

MainBlockStairsFromB_Entrance = [4, 24]
MainBlockStairsFromB_Exit = [112, 4]
MainBlockStairsToA_Entrance = [123, 90]
MainBlockStairsToA_Exit = [123, 27]
#Inside Nodes
S_EX1 = [96, 476]
S_EX2 = [447, 400]
LB_ST1 = [71, 389]
LB_ST2 = [300, 96]
EB_EX1 = [344, 397]
MB_EX1 = [274, 287]
MB_EX2 = [333, 108]
M_E1 = [183, 121]
M_EX1 = [184, 336]
M_ST1 = [450, 499]
M_ST2 = [97, 91]
C_ST1 = [218, 238]
C_ST2 = [218, 95]
G_ST1 = [185, 251]
G_ST2 = [185, 108]
DT_E1 = [236, 317]
DT_STE = [720, 123]
DT_ST1 = [135, 304]
DT_ST2 = [526, 261]
SFC_EX1 = [324, 81]
SFC_EX2 = [34, 198]
SFC_EX3 = [351, 375]
SixthFormCenter_NodePosition = [SFC_EX1, SFC_EX2, SFC_EX3]

```

```

#General Buttons:
_X_Button = Button(image = None, pos =(0,0) ,text_input = "X", font = get_font((Height)/20), base_color = "Red",
hovering_color = "Green")
#Main Menu Buttons:
START_Button = Button(image = FilledRectangle ,pos = (Center_X ,Center_Y -(Height)/5) ,text_input = "START",
font = get_font((Height)/20), base_color = "#000317", hovering_color = "Yellow")

LOGIN_Button = Button(image = FilledRectangle, pos = (Center_X, Center_Y) ,text_input = "LOGIN", font =
get_font((Height)/20), base_color = "#000317", hovering_color = "Yellow")

EXIT_Button = Button(image = FilledRectangle, pos =(Center_X, Center_Y +(Height)/5) ,text_input = "Exit", font =
get_font((Height)/20), base_color = "#000317", hovering_color = "Yellow")
#LogIn Buttons
Username_Button = Button(image = None ,pos = (Center_X,Center_Y +(Height)/5) ,text_input = "Please Click To
Enter Your Username Here", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

Password_Button = Button(image = None ,pos = (Center_X,Center_Y) ,text_input = "Please Click To Enter Your
Password Here", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

AccessKey_Button = Button(image = None ,pos = (Center_X,Center_Y - (Height)/5) ,text_input = "Please Click To
Enter The Access Key", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

CreateAccount_Button = Button(image = None ,pos = (Center_X,Center_Y - (Height)/5) ,text_input = "Don't have an
account? Click here to make one", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

Check_Button = Button(image = None ,pos = (Center_X +(Width)*(3/8),Center_Y) , text_input = "Check", font =
get_font((Height)/20), base_color = "Red", hovering_color = "Green")

AccessKey_Rect = Button(image = None ,pos = (Center_X,Center_Y +(Height)/5) ,text_input = "Please Click To
Enter AccessKey Here If You Have One", font = get_font((Height)/40), base_color = "Red", hovering_color =
"Green")
#Main Screen Buttons:
Menu_Button = Button(image = Menu_Image, pos =(60,60) ,text_input = "", font = get_font((Height)/15),
base_color = "Red", hovering_color = "Green")

Search_Button = Button(image = Search_Image, pos =(Width - 60,60) ,text_input = "", font =
get_font((Height)/15), base_color = "Red", hovering_color = "Green")

Pin1_Button = Button(image = Pin1_Image ,pos = (60 ,(Height)/6) , text_input = str(UserPins[0]) + (UserPins[0]+
"to " + UserPins[1]) + str(UserPins[1]), font = get_font((Height)/80), base_color = "Red", hovering_color =
"Green")

Pin2_Button = Button(image = Pin2_Image ,pos = (60, 5*(Height)/24) , text_input = str(UserPins[2]) +
(UserPins[2]+ "to " + UserPins[3])+ str(UserPins[3]), font = get_font((Height)/80), base_color = "Red",
hovering_color = "Green")

Pin3_Button = Button(image = Pin3_Image ,pos = {60,(Height)/4} , text_input = str(UserPins[4]) + (UserPins[4]+
"to " + UserPins[5]) + str(UserPins[5]), font = get_font((Height)/80), base_color = "Red", hovering_color =
"Green")
#Search Screen Buttons:
Location_Button = Button(image = None ,pos = (Center_X,Center_Y -(Height)/2.5) ,text_input = "Please Click To
Enter Your Location Here", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

Destination_Button = Button(image = None ,pos = (Center_X,Center_Y -(Height)/2.5) ,text_input = "Please Click
To Enter Your Destination Here", font = get_font((Height)/40), base_color = "Red", hovering_color = "Green")

Locate_Button = Button(image = None ,pos = (Center_X +(Width)*(3/8),Center_Y -(Height)/2.5) , text_input =
"Search", font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")

Result1_Button = Button(image = None ,pos = (Center_X -(Width)/4,Center_Y) , text_input = "Likely Result 1:",
font = get_font((Height)/30), base_color = "Red", hovering_color = "Green")

Result2_Button = Button(image = None ,pos = (Center_X -(Width)/4,Center_Y +(Height)/10) , text_input = "Likely
Result 2:", font = get_font((Height)/30), base_color = "Red", hovering_color = "Green")

Result3_Button = Button(image = None ,pos = (Center_X -(Width)/4,Center_Y +(Height)/5) , text_input = "Likely
Result 3:", font = get_font((Height)/30), base_color = "Red", hovering_color = "Green")

Result4_Button = Button(image = None ,pos = (Center_X -(Width)/4,Center_Y +(Height)/5 +(Height)/10) ,
text_input = "Likely Result 4:", font = get_font((Height)/30), base_color = "Red", hovering_color = "Green")
#Side Menu Buttons:
Close_Button = Button(image = None ,pos = ((Width)/9,Center_Y -(Height)/2.25) , text_input = "Close", font =
get_font((Height)/12.5), base_color = "Red", hovering_color = "Green")

Pins_Button = Button(image = None ,pos = ((Width)/12,Center_Y -(Height)/7) , text_input = "Pins", font =
get_font((Height)/12.5), base_color = "Red", hovering_color = "Green")

Status_Button = Button(image = None ,pos = ((width)/9,Center_Y) , text_input = "Status", font =
get_font((Height)/15), base_color = "Red", hovering_color = "Green")

LogIn_Button = Button(image = None ,pos = ((Width)/9,Center_Y -(Height)/3.5) , text_input = "Log In", font =
get_font((Height)/12.5), base_color = "Red", hovering_color = "Green")

LogOff_Button = Button(image = None ,pos = ((Width)/9,Center_Y -(Height)/3.5) , text_input = "Log Off", font =
get_font((Height)/15), base_color = "Red", hovering_color = "Green")

Home_Button = Button(image = None ,pos = ((Width)/12,Center_Y +(Height)/2.25) , text_input = "Home", font =
get_font((Height)/12.5), base_color = "Red", hovering_color = "Green")

```

```
#Pin Edit Screen Buttons:
Pin1Data_Button = Button(image = None ,pos = (Center_X -(Width)/8,Center_Y) , text_input = "1", font =
get_font((Height)/20), base_color = "Red", hovering_color = "Green")

Pin2Data_Button = Button(image = None ,pos = (Center_X -(Width)/8,Center_Y + (Height)/8) , text_input = "2",
font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")

Pin3Data_Button = Button(image = None ,pos = (Center_X -(Width)/8,Center_Y + (Height)/4) , text_input = "3",
font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")
#Status Edit Screen Buttons:
RedStatus_Button = Button(image = RedStatus_Image ,pos = (Center_X +(Width)/8,Center_Y) , text_input = "",
font = get_font((Height)/20), base_color = "Red", hovering_color = "Green")

YellowStatus_Button = Button(image = YellowStatus_Image ,pos = (Center_X ,Center_Y) , text_input = "", font =
get_font((Height)/20), base_color = "Red", hovering_color = "Green")

GreenStatus_Button = Button(image = GreenStatus_Image ,pos = (Center_X -(Width)/8,Center_Y) , text_input = "",
font = get_Font((Height)/20), base_color = "Red", hovering_color = "Green")
#Path Display Screen Buttons:
NextPath_Button = Button(image = None ,pos = (Center_X +(Width)/3,Center_Y) , text_input = "Next", font =
get_font((Height)/20), base_color = "Red", hovering_color = "Green")

LastPath_Button = Button(image = None ,pos = (Center_X +(Width)/3,(Center_Y)/2) , text_input = "Back", font =
get_font((Height)/20), base_color = "Red", hovering_color = "Green")
-----Screen -----##
def Start_Sc():
 while True:
 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,(0,0))

 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [START_Button, LOGIN_Button, EXIT_Button]:# goes through each button and uploads it onto
 the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if START_Button.checkForInput(Start_Mouse_Pos):#goes to main screen if clicked
 Main_Sc()
 if LOGIN_Button.checkForInput(Start_Mouse_Pos):#goes to login screen if clicked
 LogIn_Sc()
 if EXIT_Button.checkForInput(Start_Mouse_Pos):#exits program
 pygame.quit()
 sys.exit()

 pygame.display.update()
 clock.tick(60)
```

```

def LogIn_Sc():
 global Username_Input
 global Password_Input
 global Username_Rect
 global Password_Rect
 global CreateAccount_Button
 global X_Button
 global Check_Button
 global Username_Type
 global Password_Type
 global LoggedIn
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [Username_Button,Password_Button,CreateAccount_Button, X_Button , Check_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Username_Button.checkForInput(Start_Mouse_Pos) and Password_Type == False:#Allows user to type if clicked unless the password button has also been clicked
 Username_Type = True
 if Password_Button.checkForInput(Start_Mouse_Pos) and Username_Type == False:#Allows user to type if clicked unless the username button has also been clicked
 Password_Type = True
 if CreateAccount_Button.checkForInput(Start_Mouse_Pos):#Goes to sign up screen if clicked
 SignUp_Sc()
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
 Start_Sc()
 if Check_Button.checkForInput(Start_Mouse_Pos):#Checks if the username and password is correct and if so it signs the user in
 LOGIN_Check(Username_Input,Password_Input)
 if Username_Check == True and Password_Check == True:
 LoggedIn = True
 Load_Account(Username_Input)
 Main_Sc()
 elif Password_Check == False or Username_Check == False:
 Fail_Sc()
 else:
 Error_Sc()

 if Username_Type == True:#if the username button has been clicked the user can now type into the username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Username_Input = Username_Input[0:-1] #Username input is now from the intial character to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Username_Type = False
 elif int(len(Username_Input)) > 16:#Stops them from writing outside the text box's width
 Username_Type = False
 else:
 Username_Input += event.unicode#key pressed gets uploaded to screen
 Username_Input = Username_Input.upper()#Makes username none case sensitive by always making it uppercase
 Username_Input = Username_Input.replace(" ", "")#removes spaces

 if Password_Type == True:#if the password button has been clicked the user can now type into the password rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Password_Input = Password_Input[0:-1] #Username input is now from the intial character to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Password_Type = False
 elif int(len(Password_Input)) > 16:#Stops them from writing outside the text box's width
 Password_Type = False
 else:
 Password_Input += event.unicode#Username input is now from the intial character to the 2nd to last character essentially deleting the last char
 Password_Input = Password_Input.upper()#Stops typing if user presses enter key
 Password_Input = Password_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,(0,0))

 Password_rect = pygame.Rect(Center_X - 400 , Center_Y +50,800,100)#rectangle to identify the password being typed
 Username_rect = pygame.Rect(Center_X - 400 , Center_Y + (Height)/5 + 50,800,100)#rectangle to identify the username being typed
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),Username_rect,2)
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),Password_rect,2)

 Username_surface = get_font((Height)/20).render(Username_Input,True,(240,248,255))#text of the username inputted
 Password_surface = get_font((Height)/20).render(Password_Input,True,(240,248,255))#text of the password inputted

 text_rect = Username_surface.get_rect(midleft =((Center_X, Center_Y + (Height)/5 + 5)))
 Screen.blit(Username_surface,(Center_X - 400, Center_Y + (Height)/5 + (Height)/12))#displays the text of the username inputted
 Screen.blit(Password_surface,(Center_X - 400 , Center_Y +50))#displays the text of the password inputted

```

```

def SignUp_Sc():
 global Username_Input, Password_Input, Accesskey_Input, Username_Rect, Password_Rect, CreateAccount_Button, X_Button, Check_Button, Username_Type, Password_Type, Accesskey_Type, LoggedIn
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 for button in [Accesskey_Button, Username_Button, Password_Button, X_Button, Check_Button]:# goes through each button and uploads it onto the screen
 button.changeColor(Start_Mouse_Pos)
 button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#checks if mouse is clicking on something
 if Username_Button.checkForInput(Start_Mouse_Pos) and Password_Type == False and Accesskey_Type == False:#allows user to type if clicked unless the password button has also been clicked
 Username_Type = True
 if Password_Button.checkForInput(Start_Mouse_Pos) and Username_Type == False and Accesskey_Type == False:#allows user to type if clicked unless the password button has also been clicked
 Password_Type = True
 if Accesskey_Button.checkForInput(Start_Mouse_Pos) and Username_Type == False and Password_Type == False:#allows user to type if clicked unless the username button has also been clicked
 Accesskey_Type = True
 if X_Button.checkForInput(Start_Mouse_Pos):#removes screen and goes back to main menu
 Start_Sc()
 if Check_Button.checkForInput(Start_Mouse_Pos):#checks if the username and password is correct and if so it signs the user in
 if Accesskey_Input == "":
 User_List.append(Username_Input)
 Password_List.append(Password_Input)
 AccessKey_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Store_Accounts()
 LoggedIn = True
 Load_Account(Username_Input)
 Main_Sc()
 elif Accesskey_Input == "ADMINSE":
 User_List.append(Username_Input)
 Password_List.append(Password_Input)
 AccessKey_List.append(Accesskey_Input)
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Pins_List.append("0")
 Store_Accounts()
 LoggedIn = True
 Load_Account(Username_Input)
 Main_Sc()
 elif Password_Input == "" or Username_Input == "":
 False_Sc2()
 else:
 False_Sc()

 if Username_Type == True:#if the username button has been clicked the user can now type into the Username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Username_Input = Username_Input[0:-1]#username input is now from the total character to the end to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#stops typing if user presses enter key
 Username_Type = False
 elif int(len(Username_Input)) > 10:#stops them from writing outside the text box's width
 Username_Type = False
 else:
 Username_Input += event.unicode
 Username_Input = Username_Input.upper()
 Username_Input = Username_Input.replace(" ", "")#removes spaces

 if Password_Type == True:#if the password button has been clicked the user can now type into the password rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Password_Input = Password_Input[0:-1]#username input is now from the total character to the end to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#stops typing if user presses enter key
 Password_Type = False
 elif int(len(Password_Input)) > 10:#stops them from writing outside the text box's width
 Password_Type = False
 else:
 Password_Input += event.unicode
 Password_Input = Password_Input.upper()
 Password_Input = Password_Input.replace(" ", "")#removes spaces

 if Accesskey_Type == True:#if the password button has been clicked the user can now type into the password rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Accesskey_Input = Accesskey_Input[0:-1]#username input is now from the total character to the end to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#stops typing if user presses enter key
 Accesskey_Type = False
 elif int(len(Accesskey_Input)) > 10:#stops them from writing outside the text box's width
 Accesskey_Type = False
 else:
 Accesskey_Input += event.unicode
 Accesskey_Input = Accesskey_Input.upper()
 Accesskey_Input = Accesskey_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlays any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,(0,0))

 Password_rect = pygame.Rect(Center_X - 400 , Center_Y + (Height)/5 + 50,500,100)rectangle to identify what text is in the password
 Username_rect = pygame.Rect(Center_X - 400 , Center_Y + (Height)/5 + 50,500,100)rectangle to identify what text is in the username
 Accesskey_rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/5 + 50,500,100)rectangle to identify what text is in the access key
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),username_rect,2)
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),Password_rect,2)
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),Accesskey_rect,2)

 Username_surface = get_font((Height)/20).render(Username_Input,True,(240,240,255))text of the username
 Password_surface = get_font((Height)/20).render(Password_Input,True,(240,240,255))text of the password
 Accesskey_surface = get_font((Height)/20).render(Accesskey_Input,True,(240,240,255))text of the access key
 text_rect = Username_surface.get_rect(midleft =(Center_X, Center_Y + (Height)/5 + 50))
 Screen.blit(Username_surface,(Center_X - 400, Center_Y + (Height)/5 + (Height)/20))displays the text of the username
 Screen.blit(Password_surface,(Center_X - 400, Center_Y + 50))displays the text of the password
 Screen.blit(Accesskey_surface,(Center_X - 400, Center_Y - (Height)/5))displays the text of the access key

```

```
def Main_Sc():
 global Menu_Button, Search_Button ,LoggedIn
 while True:
 if LoggedIn == True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [Menu_Button , Search_Button ,Pin1_Button ,Pin2_Button ,Pin3_Button]:# goes through
each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Menu_Button.checkForInput(Start_Mouse_Pos) :
 Menu_Sc()
 if Search_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
username button has also been clicked
 LocationSearch_Sc()
 if Pin1_Button.checkForInput(Start_Mouse_Pos):
 False_Sc5()
 if Pin2_Button.checkForInput(Start_Mouse_Pos):
 if UserPins[2] == "0" and UserPins[3] == "1":#if the pin is empty it goes to a false
screen to correct the user accordingly
 False_Sc5()
 else:#Inputs pin into path finding algorithim
 StartLocation = UserPins[2]
 EndLocation = UserPins[3]
 Path1_Sc()
 if Pin1_Button.checkForInput(Start_Mouse_Pos):
 if UserPins[4] == "0" and UserPins[5] == "1":#if the pin is empty it goes to a false
screen to correct the user accordingly
 False_Sc5()
 else:#Inputs pin into path finding algorithim
 StartLocation = UserPins[4]
 EndLocation = UserPins[5]
 Path1_Sc()

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,((Center_X)/4, 0 - (Height)/2))

 else:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

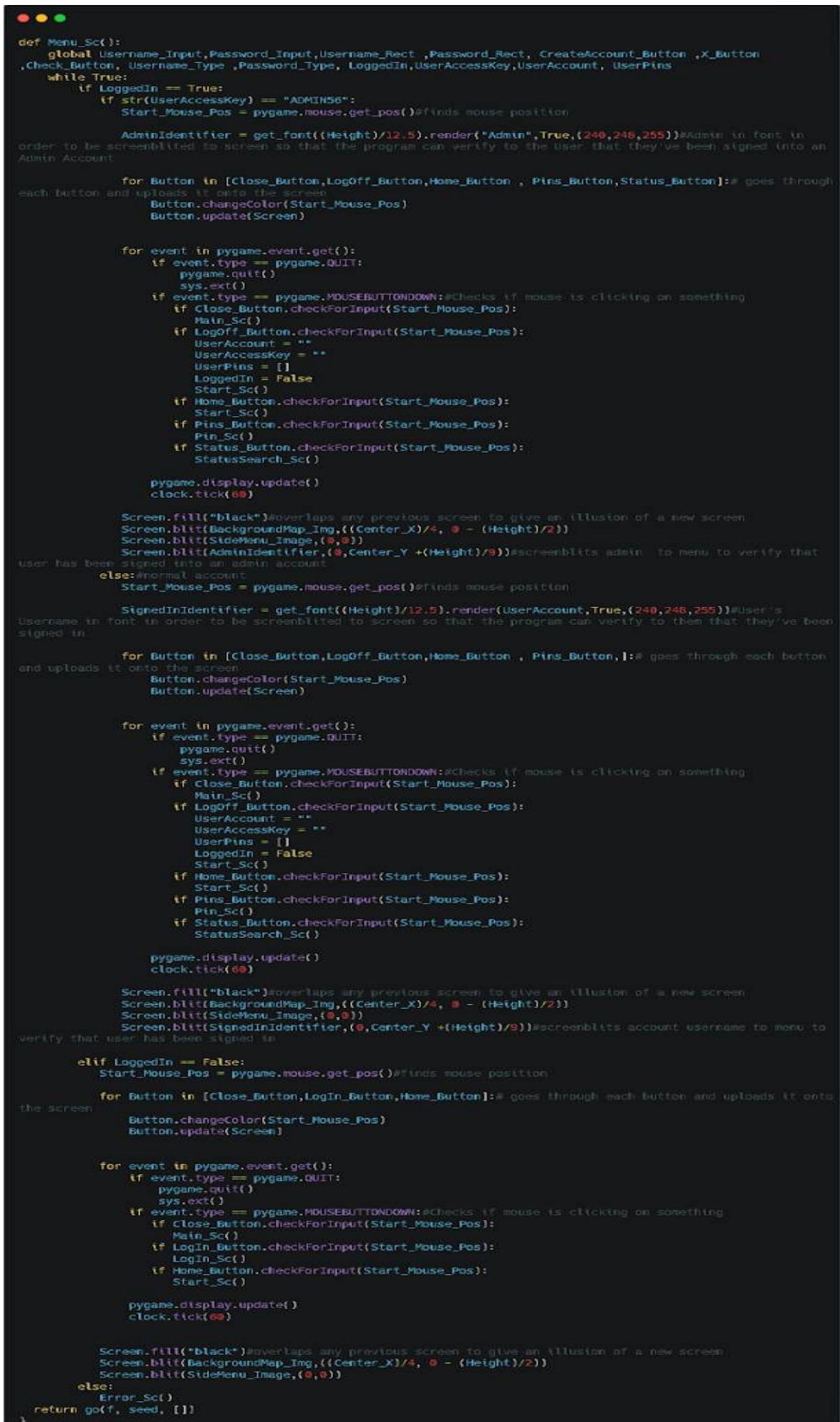
 for Button in [Menu_Button , Search_Button]:# goes through each button and uploads it onto the
screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Menu_Button.checkForInput(Start_Mouse_Pos) :
 Menu_Sc()
 if Search_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
username button has also been clicked
 LocationSearch_Sc()

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,((Center_X)/4, 0 - (Height)/2))
```

```


def Menu_Sc():
 global Username_Input, Password_Input, Username_Rect, Password_Rect, CreateAccount_Button, X_Button, Check_Button, Username_Type, Password_Type, LoggedIn, UserAccessKey, UserAccount, UserPins
 while True:
 if LoggedIn == True:
 if str(UserAccessKey) == "ADMIN56":
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 AdminIdentifier = get_font((Height)/12.5).render("Admin",True,(240,240,255))#admin in font in order to be screenblitted to screen so that the program can verify to the user that they've been signed into an Admin Account
 for Button in [Close_Button,LogOff_Button,Home_Button, Pins_Button,Status_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Close_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if Logoff_Button.checkForInput(Start_Mouse_Pos):
 UserAccount = ""
 UserAccessKey = ""
 UserPins = []
 LoggedIn = False
 Start_Sc()
 if Home_Button.checkForInput(Start_Mouse_Pos):
 Start_Sc()
 if Pins_Button.checkForInput(Start_Mouse_Pos):
 Pin_Sc()
 if Status_Button.checkForInput(Start_Mouse_Pos):
 StatusSearch_Sc()
 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,((Center_X)/4, 0 - (Height)/2))
 Screen.blit(SideMenu_Image,(0,0))
 Screen.blit(AdminIdentifier,(0,Center_Y +(Height)/9))#screenblits admin to menu to verify that user has been signed into an admin account
 else:#normal account
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 SignedIdentifier = get_font((Height)/12.5).render(UserAccount,True,(240,240,255))#user's username in font in order to be screenblitted to screen so that the program can verify to them that they've been signed in
 for Button in [Close_Button,LogOff_Button,Home_Button, Pins_Button,]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Close_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if Logoff_Button.checkForInput(Start_Mouse_Pos):
 UserAccount = ""
 UserAccessKey = ""
 UserPins = []
 LoggedIn = False
 Start_Sc()
 if Home_Button.checkForInput(Start_Mouse_Pos):
 Start_Sc()
 if Pins_Button.checkForInput(Start_Mouse_Pos):
 Pin_Sc()
 if Status_Button.checkForInput(Start_Mouse_Pos):
 StatusSearch_Sc()
 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,((Center_X)/4, 0 - (Height)/2))
 Screen.blit(SideMenu_Image,(0,0))
 Screen.blit(SignedIdentifier,(0,Center_Y +(Height)/9))#screenblits account username to menu to verify that user has been signed in
 elif LoggedIn == False:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 for Button in [Close_Button,LogIn_Button,Home_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Close_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if LogIn_Button.checkForInput(Start_Mouse_Pos):
 Login_Sc()
 if Home_Button.checkForInput(Start_Mouse_Pos):
 Start_Sc()
 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,((Center_X)/4, 0 - (Height)/2))
 Screen.blit(SideMenu_Image,(0,0))
 else:
 Error_Sc()
 return goff, seed, []

```

```

def LocationSearch_Sc():
 global Location_Input, Location_Rect, X_Button ,Location_Type, Result1_Button ,Result2_Button
 ,Result3_Button ,Result4_Button ,Result1_Input , Result2_Input,Result3_Input,Result4_Input, ActualAllocation
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [X_Button, Location_Button, Locate_Button, Result1_Button , Result2_Button ,
Result3_Button ,Result4_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Location_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
password button has also been clicked
 Location_Type = True
 if Locate_Button.checkForInput(Start_Mouse_Pos):#searches the room list for inputted value and
values similar to it
 Search(Location_Input)
 if Result1_Button.checkForInput(Start_Mouse_Pos):#selects Result 1 as the starting position
 ActualAllocation = Result1_Input
 if ActualAllocation == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 DestinationSearch_Sc()
 if Result2_Button.checkForInput(Start_Mouse_Pos):#selects Result 2 as the starting position
 ActualAllocation = Result2_Input
 if ActualAllocation == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 DestinationSearch_Sc()
 if Result3_Button.checkForInput(Start_Mouse_Pos):#selects Result 3 as the starting position
 ActualAllocation = Result3_Input
 if ActualAllocation == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 DestinationSearch_Sc()
 if Result4_Button.checkForInput(Start_Mouse_Pos):#selects Result 4 as the starting position
 ActualAllocation = Result4_Input
 if ActualAllocation == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 DestinationSearch_Sc()
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
 ClearSearchData()#resets search data for new room to be searched
 Main_Sc()

 if Location_Type == True:#if the username button has been clicked the user can now type into the
username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Location_Input = Location_Input[0:-1] #Username input is now from the intial character
 to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Location_Type = False
 elif int(len(Location_Input)) > 10:#Stops them from writing outside the text box's width
 Location_Type = False
 else:
 Location_Input += event.unicode#key pressed gets uploaded to screen
 Location_Input = Location_Input.upper()#Makes username none case sensitive by always
 making it uppercase
 Location_Input = Location_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen

 Location_Rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/2.5 + 50, 600, 100)
 pygame.draw.rect(Screen,pygame.Color('lightskyblue'),Location_Rect,2)

 Location_surface = get_font((Height)/20).render(Location_Input,True,(240,248,255))#text of the inputted
location
 Result1_surface = get_font((Height)/30).render(Result1_Input,True,(240,248,255))#text of the result 1
 Result2_surface = get_font((Height)/30).render(Result2_Input,True,(240,248,255))#text of the result 2
 Result3_surface = get_font((Height)/30).render(Result3_Input,True,(240,248,255))#text of the result 3
 Result4_surface = get_font((Height)/30).render(Result4_Input,True,(240,248,255))#text of the result 4

 text_rect = Location_surface.get_rect(midleft =((Center_X, Center_Y - (Height)/2.5 + 5)))
 Screen.blit(Location_surface,(Center_X - 400, Center_Y - (Height)/2.5 + (Height)/12))
 Screen.blit(Result1_surface,(Center_X -200 , Center_Y))#displayed text of the result 1
 Screen.blit(Result2_surface,(Center_X -200 , Center_Y +(Height)/10))#displayed text of the result 2
 Screen.blit(Result3_surface,(Center_X -200 , Center_Y +(Height)/5))#displayed text of the result 3
 Screen.blit(Result4_surface,(Center_X -200 , Center_Y +(Height)/10 +(Height)/5))#displayed text of the
result 4
 Screen.blit(R1StatusDisplay,(Center_X,Center_Y))#displayed status of the result 1
 Screen.blit(R2StatusDisplay,(Center_X,Center_Y +(Height)/10))#displayed status of the result 2
 Screen.blit(R3StatusDisplay,(Center_X,Center_Y +(Height)/5))#displayed status of the result 3
 Screen.blit(R4StatusDisplay,(Center_X,Center_Y +(Height)/10 +(Height)/5))#displayed status of the
result 4

```

```

def DestinationSearch_Sc():
 global Destination_Input
 global Destination_Rect
 global X_Button
 global Destination_Type
 global Result1_Button
 global Result2_Button
 global Result3_Button
 global Result4_Button
 global Result1_Input , Result2_Input,Result3_Input,Result4_Input
 global ActualDestination
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [X_Button, Destination_Button, Locate_Button, Result1_Button , Result2_Button ,
Result3_Button,Result4_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Destination_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
password button has also been clicked
 Destination_Type = True
 if Locate_Button.checkForInput(Start_Mouse_Pos):#Searches the room list for inputted value and
values similar to it
 Search(Destination_Input)
 if Result1_Button.checkForInput(Start_Mouse_Pos):#selects Result 1 as the ActualDestination
 ActualDestination = Result1_Input
 if ActualDestination == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc4()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Path1_Sc()
 if Result2_Button.checkForInput(Start_Mouse_Pos):#selects Result 2 as the ActualDestination
 ActualDestination = Result2_Input
 if ActualDestination == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc4()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Path2_Sc()
 if Result3_Button.checkForInput(Start_Mouse_Pos):#selects Result 3 as the ActualDestination
 ActualDestination = Result3_Input
 if ActualDestination == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc4()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Path1_Sc()
 if Result4_Button.checkForInput(Start_Mouse_Pos):#selects Result 4 as the ActualDestination
 ActualDestination = Result4_Input
 if ActualDestination == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc4()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Path1_Sc()
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
 ClearSearchData()#resets search data for new room to be searched
 Main_Sc()

 if Destination_Type == True:#if the username button has been clicked the user can now type into
the username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Destination_Input = Destination_Input[0:-1] #username input is now from the 1st
character to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Destination_Type = False
 elif len(Destination_Input) > 16:#Stops them from writing outside the text box's width
 Destination_Type = False
 else:
 Destination_Input += event.unicode#key pressed gets uploaded to screen
 Destination_Input = Destination_Input.upper()#makes username none case sensitive by
always making it uppercase
 Destination_Input = Destination_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlays any previous screen to give an illusion of a new screen
 Destination_Rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/2.5 + 50,800,100)
 pygame.draw.rect(Screen,pygame.Color('lightskyblue'),Destination_Rect,2)

 Destination_surface = get_font((Height)/20).render(Destination_Input,True,(240,240,255))#text of the
inputted location
 Result1_surface = get_font((Height)/30).render(Result1_Input,True,(240,240,255))#text of the result 1
 Result2_surface = get_font((Height)/30).render(Result2_Input,True,(240,240,255))#text of the result 2
 Result3_surface = get_font((Height)/30).render(Result3_Input,True,(240,240,255))#text of the result 3
 Result4_surface = get_font((Height)/30).render(Result4_Input,True,(240,240,255))#text of the result 4

 text_rect = Destination_surface.get_rect(midleft =((Center_X , Center_Y - (Height)/2.5 + 5)))
 Screen.blit(Destination_surface,(Center_X - 400 , Center_Y - (Height)/2.5 + (Height)/12))#displayed text
of the location input
 Screen.blit(Result1_surface,(Center_X -200 , Center_Y))#displayed text of the result 1
 Screen.blit(Result2_surface,(Center_X -200 , Center_Y +(Height)/10))#displayed text of the result 2
 Screen.blit(Result3_surface,(Center_X -200 , Center_Y +(Height)/5))#displayed text of the result 3
 Screen.blit(Result4_surface,(Center_X -200 , Center_Y +(Height)/10 +(Height)/5))#displayed text of the
result 4
 Screen.blit(R1StatusDispaly,(Center_X,Center_Y))#displayed status of the result 1
 Screen.blit(R2StatusDispaly,(Center_X,Center_Y +(Height)/10))#displayed status of the result 2
 Screen.blit(R3StatusDispaly,(Center_X,Center_Y +(Height)/5))#displayed status of the result 3
 Screen.blit(R4StatusDispaly,(Center_X,Center_Y +(Height)/10 +(Height)/5))#displayed status of the
result 4

```

```

def StatusSearch_Sc():
 global Location_Input , Location_Rect, X_Button ,Location_Type , Result1_Button ,Result2_Button
 ,Result3_Button , Result4_Button ,Result1_Input , Result2_Input,Result3_Input,Result4_Input ,SearchedStatus
 global R1StatusDispaly ,R2StatusDispaly,R3StatusDispaly,R4StatusDispaly
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [X_Button, Location_Button, Locate_Button, Result1_Button , Result2_Button ,
Result3_Button ,Result4_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Location_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
password button has also been clicked
 Location_Type = True
 if Locate_Button.checkForInput(Start_Mouse_Pos):
 Search(Location_Input)
 if Result1_Button.checkForInput(Start_Mouse_Pos):
 SearchedStatus = Result1_Input
 if SearchedStatus == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Status_Sc()
 if Result2_Button.checkForInput(Start_Mouse_Pos):
 SearchedStatus = Result2_Input
 if SearchedStatus == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Status_Sc()
 if Result3_Button.checkForInput(Start_Mouse_Pos):
 SearchedStatus = Result3_Input
 if SearchedStatus == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Status_Sc()
 if Result4_Button.checkForInput(Start_Mouse_Pos):
 SearchedStatus = Result4_Input
 if SearchedStatus == "":
 ClearSearchData()#resets search data for new room to be searched
 False_Sc3()
 else:
 ClearSearchData()#resets search data for new room to be searched
 Status_Sc()
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to main menu
 ClearSearchData()#resets search data for new room to be searched
 Main_Sc()

 if Location_Type == True:#if the username button has been clicked the user can now type into the
username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Location_Input = Location_Input[0:-1] #Username input is now from the initial character
 to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Location_Type = False
 elif int(len(Location_Input)) > 16:#Stops them from writing outside the text box's width
 Location_Type = False
 else:
 Location_Input += event.unicode#key pressed gets uploaded to screen
 Location_Input = Location_Input.upper()#Makes username none case sensitive by always
making it uppercase
 Location_Input = Location_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

Screen.fill("black")#overlays any previous screen to give an illusion of a new screen

Location_Rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/2.5 + 50,800,100)
pygame.draw.rect(Screen,pygame.Color('lightskyblue'),Location_Rect,2)

Location_surface = get_font((Height)/20).render(Location_Input,True,(240,240,255))#text of the inputted
location
Result1_surface = get_font((Height)/30).render(Result1_Input,True,(240,240,255))#text of the result 1
Result2_surface = get_font((Height)/30).render(Result2_Input,True,(240,240,255))#text of the result 2
Result3_surface = get_font((Height)/30).render(Result3_Input,True,(240,240,255))#text of the result 3
Result4_surface = get_font((Height)/30).render(Result4_Input,True,(240,240,255))#text of the result 4

text_rect = Location_surface.get_rect(midleft =((Center_X, Center_Y - (Height)/2.5 + 5)))

Screen.blit(Location_surface,(Center_X - 400, Center_Y - (Height)/2.5 + (Height)/12))#displayed text of
the location input
Screen.blit(Result1_surface,(Center_X -200, Center_Y))#displayed text of the result 1
Screen.blit(Result2_surface,(Center_X -200 , Center_Y +(Height)/10))#displayed text of the result 2
Screen.blit(Result3_surface,(Center_X -200 , Center_Y +(Height)/5))#displayed text of the result 3
Screen.blit(Result4_surface,(Center_X -200 , Center_Y +(Height)/10 +(Height)/5))#displayed text of the
result 4
Screen.blit(R1StatusDispaly,(Center_X,Center_Y))#displayed status of the result 1
Screen.blit(R2StatusDispaly,(Center_X,Center_Y +(Height)/10))#displayed status of the result 2
Screen.blit(R3StatusDispaly,(Center_X,Center_Y +(Height)/5))#displayed status of the result 3
Screen.blit(R4StatusDispaly,(Center_X,Center_Y +(Height)/10 +(Height)/5))#displayed status of the
result 4

```

```

def Status_Sc():
 global RoomStatus,StatusDisplay
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [RedStatus_Button,YellowStatus_Button , GreenStatus_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if RedStatus_Button.checkForInput(Start_Mouse_Pos):#if red status button is pressed then status of the room becomes red
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Red")
 Main_Sc()
 if YellowStatus_Button.checkForInput(Start_Mouse_Pos):#if yellow status button is pressed then status of the room becomes yellow
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Yellow")
 Main_Sc()
 if GreenStatus_Button.checkForInput(Start_Mouse_Pos):#if green status button is pressed then status of the green becomes red
 RoomStatus_List.insert(int(Room_List.index(SearchedStatus)),"Green")
 Main_Sc()

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,(0,0))

 if RoomStatus_List[int(Room_List.index(SearchedStatus))] == "Red":
 StatusDisplay = RedStatus_Image
 if RoomStatus_List[int(Room_List.index(SearchedStatus))] == "Yellow":
 StatusDisplay = YellowStatus_Image
 if RoomStatus_List[int(Room_List.index(SearchedStatus))] == "Green":
 StatusDisplay = GreenStatus_Image

 Room_surface = get_font((Height)/20).render(SearchedStatus + "'s Current Status",True,(240,248,255))

 Screen.blit(Room_surface,(Center_X - 400, Center_Y - (Height)/5 - (Height)/12))#displayed text of the location input
 Screen.blit(StatusDisplay,(Center_X, Center_Y - (Height)/5))#displayed status of the location input

def Pin_Sc():
 global Pin1Searching, Pin2Searching ,Pin3Searching,UserPins
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [Pin1Data_Button, Pin2Data_Button ,Pin3Data_Button ,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if Pin1Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin1 based on the room searched for
 Pin1Searching = True
 PinSearch_Sc()
 if Pin2Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin2 based on the room searched for
 Pin2Searching = True
 PinSearch_Sc()
 if Pin3Data_Button.checkForInput(Start_Mouse_Pos):#links to a search screen that changes the value of user's pin3 based on the room searched for
 Pin3Searching = True
 PinSearch_Sc()

 pygame.display.update()
 clock.tick(60)

 PinHeader = get_font((Height)/10).render("Pins",True,(240,248,255))
 RoomHeader = get_font((Height)/10).render("Rooms",True,(240,248,255))

 Pin1Edit_surface = get_font((Height)/20).render(str(UserPins[0])+" to "+str(UserPins[1]),True,(240,248,255))#text of the input
 Pin2Edit_surface = get_font((Height)/20).render(str(UserPins[2])+" to "+str(UserPins[3]),True,(240,248,255))#text of the input
 Pin3Edit_surface = get_font((Height)/20).render(str(UserPins[4])+" to "+str(UserPins[5]),True,(240,248,255))#text of the input

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Screen.blit(BackgroundMap_Img,(0,0))
 Screen.blit(PinHeader,(Center_X -(width)/4 , Center_Y - (Height)/5))#displays the pin name of the selected pin
 Screen.blit(RoomHeader,(Center_X , Center_Y - (Height)/5))#displays the roomheader of the selected pin
 Screen.blit(Pin1Edit_surface,(Center_X, Center_Y +(Height)/8))#displays the text of the input
 Screen.blit(Pin2Edit_surface,(Center_X, Center_Y +(Height)/8))#displays the text of the input
 Screen.blit(Pin3Edit_surface,(Center_X, Center_Y +(Height)/4))#displays the text of the input

```

```

def PinSearch_Sc():
 global Location_Input, Location_Rect ,X_Button ,Location_Type , Result1_Button, Result2_Button,
 Result3_Button ,Result4_Button , Result1_Input , Result2_Input,Result3_Input,Result4_Input , Pin1Searching ,
 Pin2Searching , Pin3Searching,UserPins
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [X_Button, Location_Button, Locate_Button, Result1_Button , Result2_Button ,
 Result3_Button ,Result4_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if Location_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
 password button has also been clicked
 Location_Type = True
 if Locate_Button.checkForInput(Start_Mouse_Pos):
 Search(Location_Input)
 if Result1_Button.checkForInput(Start_Mouse_Pos):#stores 1st result as starting location for the
 pin
 if Pin1Searching == True:
 UserPins.insert(0,Result1_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result1_Input)
 if Pin3Searching == True:
 UserPins.insert(4,Result1_Input)
 ClearSearchData()#resets search data for new room to be searched
 PinSearch2_Sc()#goes to search screen for end location for the pin
 if Result2_Button.checkForInput(Start_Mouse_Pos):#stores 2nd result as starting location for the
 pin
 if Pin1Searching == True:
 UserPins.insert(0,Result2_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result2_Input)
 if Pin3Searching == True:
 UserPins.insert(4,Result2_Input)
 ClearSearchData()#resets search data for new room to be searched
 PinSearch2_Sc()#goes to search screen for end location for the pin
 if Result3_Button.checkForInput(Start_Mouse_Pos):#stores 3rd result as starting location for the
 pin
 if Pin1Searching == True:
 UserPins.insert(0,Result3_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result3_Input)
 if Pin3Searching == True:
 UserPins.insert(4,Result3_Input)
 ClearSearchData()#resets search data for new room to be searched
 PinSearch2_Sc()#goes to search screen for end location for the pin
 if Result4_Button.checkForInput(Start_Mouse_Pos):#stores 4th result as starting location for the
 pin
 if Pin1Searching == True:
 UserPins.insert(0,Result4_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result4_Input)
 if Pin3Searching == True:
 UserPins.insert(4,Result4_Input)
 ClearSearchData()#resets search data for new room to be searched
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 PinSearch2_Sc()#goes to search screen for end location for the pin

 if X_Button.checkForInput(Start_Mouse_Pos):#leaves screen and goes back to main menu
 ClearSearchData()#resets search data for new room to be searched
 Location_Input = ""
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Pin_Sc()

 if Location_Type == True:#if the username button has been clicked the user can now type into the
 username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Location_Input = Location_Input[0:-1] #username input is now from the intial character
 to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#Stops typing if user presses enter key
 Location_Type = False
 elif int(len(Location_Input)) > 16:#stops them from writing outside the text box's width
 Location_Type = False
 else:
 Location_Input += event.unicode#key pressed gets uploaded to screen
 Location_Input = Location_Input.upper()#makes username non case sensitive by always
 making it uppercase
 Location_Input = Location_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen

 Location_Rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/2.5 + 50,800,100)
 pygame.draw.rect(Screen,pygame.Color('lightskyblue'),Location_Rect,2)

 Location_surface = get_font((Height)/20).render(Location_Input,True,(240,240,255))#text of the location
 repeated
 Result1_surface = get_font((Height)/30).render(Result1_Input,True,(240,240,255))#text of result 1
 Result2_surface = get_font((Height)/30).render(Result2_Input,True,(240,240,255))#text of result 2
 Result3_surface = get_font((Height)/30).render(Result3_Input,True,(240,240,255))#text of result 3
 Result4_surface = get_font((Height)/30).render(Result4_Input,True,(240,240,255))#text of result 4

 text_rect = Location_surface.get_rect(midleft = ((Center_X , Center_Y - (Height)/2.5 + 5)))

 Screen.blit(Location_surface,(Center_X - 400, Center_Y - (Height)/2.5 + (Height)/12))#displayed text of
 the location inputed
 Screen.blit(Result1_surface,(Center_X - 200, Center_Y))#displayed text of result 1
 Screen.blit(Result2_surface,(Center_X - 200 , Center_Y +(Height)/10))#displayed text of result 2
 Screen.blit(Result3_surface,(Center_X - 200 , Center_Y +(Height)/5))#displayed text of result 3
 Screen.blit(Result4_surface,(Center_X - 200 , Center_Y +(Height)/10 +(Height)/5))#displayed text of
 result 4

```

```

def PinSearch2_Sc():
 global Location_Input
 global Location_Rect
 global X_Button
 global Location_Type
 global Result1_Button
 global Result2_Button
 global Result3_Button
 global Result4_Button
 global result1_Input , result2_Input,result3_Input,Result4_Input, Pin1Searching , Pin2Searching ,
 Pin3Searching,UserPins
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 for Button in [X_Button, location_Button, Locate_Button, Result1_Button , Result2_Button ,
 Result3_Button ,Result4_Button]:# goes through each button and uploads it onto the screen
 button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#checks if mouse is clicking on something
 if Location_Button.checkForInput(Start_Mouse_Pos):#Allows user to type if clicked unless the
 password/button has also been clicked
 Location_Type = True
 if Locate_Button.checkForInput(Start_Mouse_Pos):
 Search(Location_Input)
 if Result1_Button.checkForInput(Start_Mouse_Pos):
 if Pin1Searching == True:
 UserPins.insert(1,Result1_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result1_Input)
 if Pin3Searching == True:
 UserPins.insert(3,Result1_Input)
 ClearSearchData()#resets search data for new room to be searched
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Store_Pins(UserAccount)
 Store_Accounts()
 Pin_Sc()
 if Result2_Button.checkForInput(Start_Mouse_Pos):
 if Pin1Searching == True:
 UserPins.insert(1,Result2_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result2_Input)
 if Pin3Searching == True:
 UserPins.insert(3,Result2_Input)
 ClearSearchData()#resets search data for new room to be searched
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Store_Pins(UserAccount)
 Store_Accounts()
 Pin_Sc()
 if Result3_Button.checkForInput(Start_Mouse_Pos):
 if Pin1Searching == True:
 UserPins.insert(1,Result3_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result3_Input)
 if Pin3Searching == True:
 UserPins.insert(3,Result3_Input)
 ClearSearchData()#resets search data for new room to be searched
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Store_Pins(UserAccount)
 Store_Accounts()
 Pin_Sc()
 if Result4_Button.checkForInput(Start_Mouse_Pos):
 if Pin1Searching == True:
 UserPins.insert(1,Result4_Input)
 if Pin2Searching == True:
 UserPins.insert(2,Result4_Input)
 if Pin3Searching == True:
 UserPins.insert(3,Result4_Input)
 ClearSearchData()#resets search data for new room to be searched
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Store_Pins(UserAccount)
 Store_Accounts()
 Pin_Sc()

 if X_Button.checkForInput(Start_Mouse_Pos):#leaves screen and goes back to main menu
 ClearSearchData()#resets search data for new room to be searched
 Location_Input = ""
 Pin1Searching = False
 Pin2Searching = False
 Pin3Searching = False
 Pin_Sc()

 if Location_Type == True:#if the username/button has been clicked the user can now type into the
 #username rectangle
 if event.type == pygame.KEYDOWN:
 if event.key == pygame.K_BACKSPACE:
 Location_Input = Location_Input[0:-1]#username input is now from the initial character
 to the 2nd to last character essentially deleting the last char
 elif event.key == pygame.K_RETURN:#stops typing if user presses enter key
 Location_Type = False
 elif len(Location_Input) > 30:#stops them from writing outside the text box/its width
 Location_Type = False
 else:
 Location_Input += event.unicode#key pressed gets uploaded to screen
 Location_Input = Location_Input.upper()#makes username case sensitive by always
 making it uppercase
 Location_Input = Location_Input.replace(" ", "")#removes spaces

 pygame.display.update()
 clock.tick(60)

 Screen.fill("black")#overlaps any previous screen to give an illusion of a new screen
 Location_Rect = pygame.Rect(Center_X - 400 , Center_Y - (Height)/2.5 + 50,500,100)
 pygame.draw.rect(Screen,pygame.color('lightskyblue'),Location_Rect,2)
 Location_surface = get_font((Height)/20).render(Location_Input,True,(240,240,255))#text of the location
 inputted
 Result1_surface = get_font((Height)/30).render(result1_Input,True,(240,240,255))#text of result 1
 Result2_surface = get_font((Height)/30).render(result2_Input,True,(240,240,255))#text of result 2
 Result3_surface = get_font((Height)/30).render(result3_Input,True,(240,240,255))#text of result 3
 Result4_surface = get_font((Height)/30).render(result4_Input,True,(240,240,255))#text of result 4

 text_rect = Location_surface.get_rect(midleft = ((Center_X , Center_Y - (Height)/2.5 + 5)))
 Screen.blit(Location_surface,(Center_X - 400 , Center_Y - (Height)/2.5 + (Height)/12))#displayed text of
 the location inputed
 Screen.blit(result1_surface,(Center_X - 200 , Center_Y + (Height)/10))#displayed text of result 1
 Screen.blit(result2_surface,(Center_X - 200 , Center_Y +(Height)/10))#displayed text of result 2
 Screen.blit(result3_surface,(Center_X - 200 , Center_Y +(Height)/10))#displayed text of result 3
 Screen.blit(result4_surface,(Center_X - 200 , Center_Y +(Height)/5))#displayed text of

```

```
def Fail_Sc():#if the user enters the wrong login details when trying to sign in
 global Username_Input
 global Password_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("The Password Or Username Is Incorrect Please Try Again", True,
"Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Username_Input = ""
 Password_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 LogIn_Sc()

 pygame.display.update()
 clock.tick(60)

def False_Sc():#if the accesskey entered when making an account is incorrect
 global Username_Input
 global Password_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("The AccessKey Entered Is Incorrect.(Note that an access key is not necessary to make an account)", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Username_Input = ""
 Password_Input = ""
 AccessKey_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 SignUp_Sc()

 pygame.display.update()
 clock.tick(60)
```

```
def False_Sc2():#if the username or password entered when making an account does not match the requirements
 global Username_Input
 global Password_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("Please enter an appropriate username and password", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Username_Input = ""
 Password_Input = ""
 AccessKey_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 SignUp_Sc()

 pygame.display.update()
 clock.tick(60)

def False_Sc3():#if the inputted location is not found within the text files
 global Location_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("There are no matches for the location you picked please try again.", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Location_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 LocationSearch_Sc()

 pygame.display.update()
 clock.tick(60)
```

```
def False_Sc4():#if an empty result is picked when searching for rooms
 global Destination_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("There are no matches for the location you picked please try again.", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Destination_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 DestinationSearch_Sc()

 pygame.display.update()
 clock.tick(60)

def False_Sc5():#if an empty pin is selected
 global Destination_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("white")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("Please store rooms in the pin before use.", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Destination_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 DestinationSearch_Sc()

 pygame.display.update()
 clock.tick(60)

def Error_Sc():#if an unkown error occurs in the system
 global Username_Input
 global Password_Input
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("red")#overlaps any previous screen to give an illusion of a new screen
 Fail_TEXT = get_font(20).render("An Error Has Occured Please Try Again", True, "Black")
 Fail_RECT = Fail_TEXT.get_rect(center=(Center_X,Center_Y))
 Screen.blit(Fail_TEXT, Fail_RECT)
 Username_Input = ""
 Password_Input = ""

 for Button in [X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:#Checks if mouse is clicking on something
 if X_Button.checkForInput(Start_Mouse_Pos):#Leaves screen and goes back to login screen
 LogIn_Sc()

 pygame.display.update()
 clock.tick(60)
```

```
def Path1_Sc():
 global NextPath_Button ,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))

 if ActualLocation != "SixthFormCenter" and ActualLocation != "SilentStudy":
 CreatePath_Inside(ActualLocation,True)#Screen blits path from room to exit onto the screen
 print("ActualLocation")

 elif ActualLocation == "SixthFormCenter" or ActualLocation == "SilentStudy":
 Path2_Sc()

 for Button in [NextPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if (RoomExit_List[Room_List.index(ActualLocation)] in Stair_Nodes) == True:#Goes to the unique
path creator for stair nodes in the instance the path has a stairwell
 Path1_5_Sc()
 else:#goes to the next part of the path
 Path2_Sc()

 pygame.display.update()
 clock.tick(60)
```

```

def Path1_5_Sc():#path from stairwell entrance to stairwell exit
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")

 if RoomExit_List[Room_List.index(ActualLocation)] == "LB_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "LB_ST2":#checks if the room is upstairs in the language block
from the rooms exit node
 find_path_alg(LanguageStairsFromA_G, LanguageStairsA_C ,int((LanguageStairsFromA_G.shape)[1]),
int((LanguageStairsFromA_G.shape)[0]),int(LBStairsFromA_Entrance),int(LBStairsFromA_Exit) ,
path_color)#creates path
 cv2.imwrite("LanguageStairsFromAChainging.jpg", LanguageStairsFromA_C)#draws path onto the chainging
version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("LanguageStairsFromAChainging.jpg"),(Center_X,Center_Y))#dispalys map
with the path on it
 cv2.imwrite("LanguageStairsFromAChainging.jpg", LanguageStairsFromA_P)#reverts map to original version
without the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "M_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "M_ST2":#checks if the room is upstairs in the maths block
from the rooms exit node
 find_path_alg(MathsStairsFrom_G, MathsStairsFrom_C ,int((MathsStairsFrom_G.shape)[1]),
int((MathsStairsFrom_G.shape)[0]),MathsStairs_Entrance,MathsStairs_Exit , path_color)#creates path
 cv2.imwrite("MathStairsChainging.jpg", MathsStairsFrom_C)#draws path onto the chainging version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MathStairsChainging.jpg"),(Center_X,Center_Y))#dispalys map with the
path on it
 cv2.imwrite("MathStairsChainging.jpg", MathsStairsFrom_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "DT_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "DT_ST2":#checks if the room is upstairs in the dt block from
the rooms exit node
 find_path_alg(DTStairsFromA_G, DTStairsFromA_C ,int((DTStairsFromA_G.shape)
[1]),int((DTStairsFromA_G.shape)[0]),int(DTStairsFromA_Entrance),int(DTStairsFromA_Exit) , path_color)#creates path
 cv2.imwrite("DTStairsFromAChainging.jpg", DTStairsFromA_C)#draws path onto the chainging version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("DTStairsFromAChainging.jpg"),(Center_X,Center_Y))#dispalys map with the
path on it
 cv2.imwrite("DTStairsFromAChainging.jpg", DTStairsFromA_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "C_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "C_ST2":#checks if the room is upstairs in the main block from
the rooms exit node
 find_path_alg(MainBlockFromA_G,MainBlockFromA_C ,int((MainBlockFromA_C.shape)
[1]),int((MainBlockFromA_C.shape)[0]),ComputingBlockStairsFrom_Entrance,ComputingBlockStairsFrom_Exit,
path_color)#creates path
 cv2.imwrite("MainBlockFromAChainging.jpg",MainBlockFromA_C)#draws path onto the chainging version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromAChainging.jpg") ,(Center_X,Center_Y))#dispalys map with
the path on it
 cv2.imwrite("MainBlockFromAChainging.jpg",MainBlockFromA_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "G_ST2":#checks if the room is upstairs in the main block from
the rooms exit node
 find_path_alg(GeographyStairsFrom_G,GeographyStairsFrom_C ,int((GeographyStairsFrom_G.shape)
[1]),int((GeographyStairsFrom_G.shape)[0]),(GeographyBlockStairsFromA_Entrance),
(GeographyBlockStairsFromA_Exit) , path_color)#creates path
 cv2.imwrite("GeographyStairsFromChainging.jpg",GeographyStairsFrom_C)#draws path onto the chainging
version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("GeographyStairsFromChainging.jpg"),(Center_X,Center_Y))#dispalys map
with the path on it
 cv2.imwrite("GeographyStairsFromChainging.jpg",GeographyStairsFrom_P)#reverts map to original version
without the path on it

 for Button in [LastPath_Button,X_Button,NextPath_Button]:# goes through each button and uploads it onto
the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if RoomExit_List[Room_List.index(ActualLocation)] == "M_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "M_ST2":#goes to the outside section of the path
 Path2_Sc()
 else:#goes to the unique path creator in the instance that the stairwell has a second section
 Path1_5B_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#returns to the first part of the path
 Path1_Sc()

 pygame.display.update()
 clock.tick(60)

```

```

def Path1_5B_Sc():#path from stairwell exit to building's exit
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Screen.fill("black")
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position

 if RoomExit_List[Room_List.index(ActualLocation)] == "LB_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "LB_ST2":#checks if the room is upstairs in the language block
from the rooms exit node
 find_path_alg(LanguageStairsFromB_G, LanguageStairsB_C ,int((LanguageStairsFromB_G.shape)[1]),
int((LanguageStairsFromB_G.shape)[0]),int(LBstairsFromB_Entrance),int(LBstairsFromB_Exit), path_color)#creates
path
 cv2.imwrite("LanguageStairsFromBChainging.jpg", LanguageStairsFromB_C)#draws path onto the chaing
version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("LanguageStairsFromBChainging.jpg"),(Center_X,Center_Y))#dispalys map
with the path on it
 cv2.imwrite("LanguageStairsFromBChainging.jpg", LanguageStairsFromB_P)#reverts map to original version
without the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "DT_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "DT_ST2":#checks if the room is upstairs in the dt block from
the rooms exit node
 find_path_alg(DTStairsFromB_G, DTStairsFromB_C ,int((DTStairsFromB_G.shape)
[1]),int((DTStairsFromB_G.shape)[0]),int(DTStairsFromB_Entrance),int(DTStairsFromB_Exit) , path_color)#creates
path
 cv2.imwrite("DTStairsFromBChainging.jpg", DTStairsFromB_C)#draws path onto the chaing version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("DTStairsFromBChainging.jpg"),(Center_X,Center_Y))#dispalys map with the
path on it
 cv2.imwrite("DTStairsFromBChainging.jpg", DTStairsFromB_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "C_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "C_ST2":#checks if the room is upstairs in the main block from
the rooms exit node
 find_path_alg(MainBlockFromB_G,MainBlockFromB_C,int((MainBlockFromB_C.shape)
[1]),int((MainBlockFromB_C.shape)
[0]),MainBlockStairsFromB_Entrance,MainBlockStairsFromB_Exit,path_color)#creates path
 cv2.imwrite("MainBlockFromBChainging.jpg", MainBlockFromB_C)#draws path onto the chaing version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromBChainging.jpg") ,(Center_X,Center_Y))#dispalys map with
the path on it
 cv2.imwrite("MainBlockFromBChainging.jpg", MainBlockFromB_P)#reverts map to original version without
the path on it

 if RoomExit_List[Room_List.index(ActualLocation)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "G_ST2":#checks if the room is upstairs in the main block from
the rooms exit node
 find_path_alg(MainBlockFromA_G,MainBlockFromA_C ,int((MainBlockFromA_G.shape)
[1]),int((MainBlockFromA_G.shape)
[0]),int(GeographyBlockStairsFromB_Entrance),int(ComputingBlockStairsFrom_B_Exit), path_color)#creates path
 cv2.imwrite("MainBlockFromAChainging.jpg", MainBlockFromA_C)#draws path onto the chaing version of the
map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainBlockFromAChainging.jpg"),(Center_X,Center_Y))#dispalys map with
the path on it
 cv2.imwrite("MainBlockFromAChainging.jpg", MainBlockFromA_P)#reverts map to original version without the
path on it

 for Button in [LastPath_Button,X_Button,NextPath_Button]:# goes through each button and uploads it onto
the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#exits the path and returns to the main screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if RoomExit_List[Room_List.index(ActualLocation)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualLocation)] == "G_ST2":#Goes to the unique path creator in the instance the
stairwell still has another section
 Path1_5C_Sc()
 else:#goes to the outside scetion of the path
 Path2_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#goes back to the previous part of the path
 Path1_5_Sc()

 pygame.display.update()
 clock.tick(60)

```

```
def Path1_5C_Sc():
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 find_path_alg(MainBlockFromB_G,MainBlockFromB_C,int((MainBlockFromB_G.shape)
[1]),int((MainBlockFromB_G.shape)
[0])),int(MainBlockStairsFromB_Entrance),int(MainBlockStairsFromB_Exit),path_color)
 cv2.imwrite("MainBlockFromBChaining.jpg", MainBlockFromB_C)
 Screen.fill("black")
 Screen.blit(MainBlockFromB_Img,(Center_X,Center_Y))
 cv2.imwrite("MainBlockFromBChaining.jpg",MainBlockFromB_P)

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#exits the path and returns to the main screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):#goes to the outside section of the path
 Path2_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#goes to the previous section of the path
 Path1_5B_Sc()

 pygame.display.update()
 clock.tick(60)
```

```

def Path2_Sc():
 global NextPath_Button , LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))

 if ActualDestination != "SixthFormCenter" and ActualDestination != "SilentStudy":
 CreatePath_Outside(ActualLocation,ActualDestination)
 for Button in [NextPath_Button,LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#exits the path and returns to the main screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if (RoomExit_List[Room_List.index(ActualDestination)]) in Stair_Nodes == True:#Goes to the unique path creator for stair nodes in the instance the path has a stairwell
 Path2_5_Sc()
 else:#goes to the final section of the path
 Path3_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#goes back to previous section of path
 if (RoomExit_List[Room_List.index(ActualLocation)]) in Stair_Nodes == True:
 Path1_5_Sc()
 else:
 Path1_Sc()

 pygame.display.update()
 clock.tick(60)
 else:
 Screen.blit(Loading_Img)
 for i in range(len(SixthFormCenter_Nodes)):
 for p in range(len((SixthFormCenter_Nodes)[i])):
 if RoomExit_List[Room_List.index(ActualDestiantion)] == str(((SixthFormCenter_Nodes)[i])[p]):#checks if destination room is within the sixthfrom center
 find_path_alg(SixthFormCenter_G , SixthFormCenter_C , int((SixthFormCenter_G.shape)[1]),int((SixthFormCenter_G.shape)[0]),[(globals()((SixthFormCenter_Nodes)[i]))[0],(globals()((SixthFormCenter_Nodes)[i]))[1]] , [int(RoomX_List[Room_List.index(ActualDestiantion)]),int(RoomY_List[Room_List.index(ActualDestiantion)])],path_color)#creates path
 cv2.imwrite("SixthFormCenterChainging.jpg", SixthFormCenter_C)#draws path onto the chainging version of the map
 Screen.fill("black")
 Screen.blit(SixthFormCenter_C,(Center_X,Center_Y))#displays map with path on it
 cv2.imwrite("SixthFormCenterChainging.jpg", SixthFormCenter_P)#reverts map to oringla without path on it

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#Exits path screen and enters
 Main_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#Goes to previous section of path
 if str(RoomExit_List[Room_List.index(ActualLocation)]) in Stair_Nodes == True:
 if RoomExit_List[Room_List.index(ActualLocation)] == "G_ST1" or
 RoomExit_List[Room_List.index(ActualLocation)] == "G_ST2":
 Path1_5C_Sc()
 elif RoomExit_List[Room_List.index(ActualLocation)] == "M_ST2" or
 RoomExit_List[Room_List.index(ActualLocation)] == "M_ST1":
 Path1_5_Sc()
 else:
 Path1_5B_Sc()
 else:
 Path1_Sc()

```

```

def Path2_5_Sc():
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 print(RoomExit_List[Room_List.index(ActualDestination)])

 if RoomExit_List[Room_List.index(ActualDestination)] == "LB_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "LB_ST2":#checks if the room is upstairs in the language
block from the rooms exit node
 find_path_alg(LanguageStairsToA_G, LanguageStairsA_C ,int((LanguageStairsToA_G.shape)
[1]),int((LanguageStairsToA_G.shape)[0]), int(LBStairsToA_Entrance),int(LBStairsToA_Exit) , path_color)#creates
path
 cv2.imwrite("LanguageStairsToAChainging.jpg", LanguageStairsToA_C)#draws path onto the chaing version
of the map
 Screen.fill("black")
 Screen.blit(LanguageStairsToA_Img,(Center_X,Center_Y))
 cv2.imwrite("LanguageStairsToAChainging.jpg", LanguageStairsToA_P)#reverts map to original version
without the path on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "M_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "M_ST2":#checks if the room is upstairs in the maths block
from the rooms exit node
 find_path_alg(MathsStairsTo_G, MathsStairsTo_C ,int((MathsStairsTo_G.shape)
[1]),int((MathsStairsTo_G.shape)[0]),int(MathsStairsTo_Entrance),int(MathsStairsTo_Exit) , path_color)#creates
path
 cv2.imwrite("MathStairsChainging.jpg", MathsStairsTo_C)#draws path onto the chaing version of the map
 Screen.fill("black")
 Screen.blit(MathsStairsTo_Img,(Center_X,Center_Y))
 cv2.imwrite("MathStairsChainging.jpg", MathsStairsTo_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "DT_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "DT_ST2":#checks if the room is upstairs in the dt block
from the rooms exit node
 find_path_alg(DTStairsToA_G, DTStairsToA_C ,int((DTStairsToA_C.shape)[1]),int((DTStairsToA_C.shape)
[0]),(DTStairsToA_Entrance),(DTStairsToA_Exit), path_color)#creates path
 cv2.imwrite("DTBlockToAChainging.jpg", DTStairsToA_C)#draws path onto the chaing version of the map
 Screen.fill("black")
 Screen.blit(pygame.image.load("DTBlockToAChainging.jpg"),(Center_X,Center_Y))
 cv2.imwrite("DTBlockToAChainging.jpg", DTStairsToA_P)#reverts map to original version without the path
on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "C_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "C_ST2" or
RoomExit_List[Room_List.index(ActualDestination)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "G_ST2":#checks if the room is upstairs in the main block
from the rooms exit node
 find_path_alg(MainBlockToA_G,MainBlockToA_C,int((MainBlockToA_G.shape)[1]),int((MainBlockToA_G.shape)
[0]),int(MainBlockStairsToA_Entrance),int(MainBlockStairsToA_Exit),path_color)
 cv2.imwrite("MainBlockToAChainging.jpg", MainBlockToA_C)#draws path onto the chaing version of the map
 Screen.fill("black")
 Screen.blit(MainBlockToA_Img,(Center_X,Center_Y))
 cv2.imwrite("MainBlockToAChainging.jpg", MainBlockToA_P)#reverts map to original version without the
path on it

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if RoomExit_List[Room_List.index(ActualDestination)] == "M_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "M_ST2":#Goes to final section of path
 Path3_Sc()
 else:#Goes to the unique path creator in the instance that the stainwell has another section
 Path2_5B_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#Goes to previous section of path
 Path2_Sc()

 pygame.display.update()
 clock.tick(60)

```

```

def Path2_5B_Sc():
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))

 if RoomExit_List[Room_List.index(ActualDestination)] == "LB_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "LB_ST2":
 find_path_alg(LanguageStairsToB_G, LanguageStairsB_C ,int((LanguageStairsToB_G.shape)[1]),int((LanguageStairsToB_G.shape)[0]),int(LBStairsToB_Entrance),int(LBStairsToB_Exit), path_color)#creates
path
 cv2.imwrite("LanguageStairsToBChainging.jpg", LanguageStairsToB_C)#draws path onto the chaing version
of the map
 Screen.fill("black")
 Screen.blit(LanguageStairsToB_Img,(Center_X,Center_Y))
 cv2.imwrite("LanguageStairsToBChainging.jpg", LanguageStairsToB_P)#reverts map to original version
without the path on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "DT_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "DT_ST2":
 find_path_alg(DTStairsToB_G, DTStairsToB_C ,int((DTStairsToB_G.shape)[1]),int((DTStairsToB_G.shape)[0]),int(DTStairsToB_Entrance),int(DTStairsToB_Exit), path_color)#creates path
 cv2.imwrite("DTBlockToBChainging.jpg", DTStairsToB_C)#draws path onto the chaing version of the map
Screen.fill("black")
 Screen.blit(DTStairsToB_Img,(Center_X,Center_Y))
 cv2.imwrite("DTBlockToBChainging.jpg", DTStairsToB_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "G_ST2":
 find_path_alg(MainBlockToB_G, MainBlockToB_C ,(MainBlockToB_G.shape)[1], (MainBlockToB_G.shape)[0],GeographyBlockStairsToA_Entrance ,GeographyBlockStairsToA_Exit , path_color)#creates path
 cv2.imwrite("MainBlockToBChainging.jpg", MainBlockToB_C)#draws path onto the chaing version of the map
Screen.fill("black")
 Screen.blit(MainBlockToB_Img,(Center_X,Center_Y))
 cv2.imwrite("MainBlockToBChainging.jpg", MainBlockToB_P)#reverts map to original version without the
path on it

 if RoomExit_List[Room_List.index(ActualDestination)] == "C_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "C_ST2":
 find_path_alg(MainBlockToB_G,MainBlockToB_C,(MainBlockToB_G.shape)[1],(MainBlockToB_G.shape)[0],ComputingBlockStairsTo_Entrance,ComputingBlockStairsTo_Exit, path_color)#creates path
 cv2.imwrite("MainBlockToBChainging.jpg", DTStairsToB_C)#draws path onto the chaing version of the map
Screen.fill("black")
 Screen.blit(MainBlockToB_Img,(Center_X,Center_Y))
 cv2.imwrite("MainBlockToBChainging.jpg", MainBlockToB_P)#reverts map to original version without the
path on it

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):
 if RoomExit_List[Room_List.index(ActualDestination)] == "G_ST1" or
RoomExit_List[Room_List.index(ActualDestination)] == "G_ST2":#Goes to the unique path creator in the instance
that the stairwell has another section
 Path2_5C_Sc()
 else:#Goes to final section of path
 Path3_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#Goes to previous section of path
 Path2_5_Sc()

 pygame.display.update()
 clock.tick(60)

```

```
def Path2_5C_Sc():
 global NextPath_Button,LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))

 find_path_alg(GeographyStairsTo_G,GeographyStairsTo_C,int((GeographyStairsTo_G.shape)[1]),int((GeographyStairsTo_G.shape)[0]),int(GeographyBlockStairsToB_Entrance),int(GeographyBlockStairsToB_Exit),path_color)#creates path
 cv2.imwrite("GeographyStairsToChainging.jpg", GeographyStairsTo_C)#draws path onto the chainging version
 of the map
 Screen.fill("black")
 Screen.blit(GeographyStairsTo_Img,(Center_X,Center_Y))
 cv2.imwrite("GeographyStairsToChainging.jpg",GeographyStairsTo_P)#reverts map to original version
 without the path on it

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#exits path and goes to main screen
 Main_Sc()
 if NextPath_Button.checkForInput(Start_Mouse_Pos):#Goes to final section of path
 Path3_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#Goes to previous section of path
 Path2_5B_Sc()

 pygame.display.update()
 clock.tick(60)

def Path3_Sc():
 global LastPath_Button,X_Button
 while True:
 Start_Mouse_Pos = pygame.mouse.get_pos()#finds mouse position
 Screen.fill("black")
 Screen.blit(Loading_Img,(Center_X,Center_Y))
 CreatePath_Inside(ActualDestination,False)

 for Button in [LastPath_Button,X_Button]:# goes through each button and uploads it onto the screen
 Button.changeColor(Start_Mouse_Pos)
 Button.update(Screen)

 for event in pygame.event.get():
 if event.type == pygame.QUIT:
 pygame.quit()
 sys.exit()
 if event.type == pygame.MOUSEBUTTONDOWN:
 if X_Button.checkForInput(Start_Mouse_Pos):#exits path and goes to main screen
 Main_Sc()
 if LastPath_Button.checkForInput(Start_Mouse_Pos):#Goes to previous section of path
 if str(RoomExit_List[Room_List.index(ActualDestination)]) in Stair_Nodes == True:
 if RoomExit_List[Room_List.index(ActualDestination)] == "M_ST1" or
 RoomExit_List[Room_List.index(ActualDestination)] == "M_ST2":
 Path2_5_Sc()
 else:
 Path2_5B_Sc()
 else:
 Path2_Sc()

 pygame.display.update()
 clock.tick(60)
```

```
-----Log In-----##

def LOGIN_Check.Username, Password):
 global Username_Check
 global Password_Check
 x = str(username) in User_List#X becomes true if the inputed username is within the usernames array in the
 accounts dictionary
 y = str(password) in Password_List#Y becomes true if the inputed password is within the passwords array in
 the accounts dictionary
 if x == True:#Username and password check allows us to see if the username is the problem or the password
 is the problem
 Username_Check = True
 if y == True:
 Password_Check = True

def Store_Accounts():
 global User_List, Password_List ,AccessKey_List,NumberOfPins_List,Pins_List
 with open("Accounts.txt", 'w') as f:
 f.truncate(0)
 with open("Accounts.txt", 'w') as f:
 for i in range(len(User_List)):
 f.write(str(User_List[i -1]) + " " +str(Password_List[i -1]) + " " + str(AccessKey_List[i-1]) + " " +
 str(Pins_List[i -1])+ " " + str(Pins_List[i]) + " " + str(Pins_List[i + 1]) + " " + str(Pins_List[i+2])+ " " +
 str(Pins_List[i+3]) + " " + str(Pins_List[i + 4]))
 f.write('\n')

def Store_Pins(username):
 global UserPins, Pins_List
 Pins_List.insert(3*int(User_List.index(username)),UserPins[0])#Stores the start location of the 1st pin
 Pins_List.insert(3*int(User_List.index(username))+1,UserPins[1])#Stores the end location of the 1st pin
 Pins_List.insert(3*int(User_List.index(username))+2,UserPins[2])#Stores the start location of the 2nd pin
 Pins_List.insert(3*int(User_List.index(username))+3,UserPins[3])#Stores the end location of the 2nd pin
 Pins_List.insert(3*int(User_List.index(username))+4,UserPins[4])#Stores the start location of the 3rd pin
 Pins_List.insert(3*int(User_List.index(username))+5,UserPins[5])#Stores the end location of the 3rd pin

def Load_Account(username):
 global UserAccount, UserAccessKey , UserPins
 UserAccount = username
 UserAccessKey = AccessKey_List[int(User_List.index(username))]
 UserPins.Insert(0,str(Pins_List[3*int(User_List.index(username))]))
 UserPins.insert(1,str(Pins_List[3*int(User_List.index(username))+1]))
 UserPins.insert(2,str(Pins_List[3*int(User_List.index(username))+2]))
 UserPins.insert(3,str(Pins_List[3*int(User_List.index(username))+3]))
 UserPins.insert(4,str(Pins_List[3*int(User_List.index(username))+4]))
 UserPins.insert(5,str(Pins_List[3*int(User_List.index(username))+5]))
```

```
-----Search-----##
def Search(Place):
 global Result1_Input,Result2_Input ,Result3_Input ,Result4_Input,
 RepeatResult,R1StatusDispaly,R2StatusDispaly,R3StatusDispaly,R4StatusDispaly
 for i in range(len(Place)-1):#repeats for the number of characters within the input
 for x in range(len(Room_List)):#repeats for the number of rooms in total to compare each room with the
 input reducing the number of characters each time to see which ones are most similar and then storing them in
 results
 if Place == Room_List[x]:
 Result1_Input = Room_List[x]
 RepeatResult = True
 if Place[0:-i] == (Room_List[x])[0:-i] and Result1_Input == "" and i!=0:
 Result1_Input = Room_List[x]
 RepeatResult = True
 if Place[0:-i] == (Room_List[x])[0:-i] and Result2_Input == "" and RepeatResult == False and i!=0:
 Result2_Input = Room_List[x]
 RepeatResult = True
 if Place[0:-i] == (Room_List[x])[0:-i] and Result3_Input == "" and RepeatResult == False and i!=0:
 Result3_Input = Room_List[x]
 RepeatResult = True
 if Place[0:-i] == (Room_List[x])[0:-i] and Result4_Input == "" and RepeatResult == False and i!=0:
 Result4_Input = Room_List[x]
 RepeatResult = True
 RepeatResult = False

 R1StatusDispaly = RoomStatusFinder(Result1_Input)
 R2StatusDispaly = RoomStatusFinder(Result2_Input)
 R3StatusDispaly = RoomStatusFinder(Result3_Input)
 R4StatusDispaly = RoomStatusFinder(Result4_Input)

def RoomStatusFinder(Room):
 if Room == '':
 return RedStatus_Image
 else:
 if RoomStatus_List[int(Room_List.index(Room))] == "Red" :
 return RedStatus_Image
 if RoomStatus_List[int(Room_List.index(Room))] == "Yellow":
 return YellowStatus_Image
 if RoomStatus_List[int(Room_List.index(Room))] == "Green":
 return GreenStatus_Image

def ClearSearchData():#resets search data for new room to be searched
 global Result1_Input,Result2_Input ,Result3_Input
 ,Result4_Input,R1StatusDispaly,R2StatusDispaly,R3StatusDispaly,R4StatusDispaly,Location_Input
 R1StatusDispaly = RedStatus_Image
 R2StatusDispaly = RedStatus_Image
 R3StatusDispaly = RedStatus_Image
 R4StatusDispaly = RedStatus_Image
 Result1_Input = ""
 Result2_Input = ""
 Result3_Input = ""
 Result4_Input = ""
 Location_Input = ""
```

```
----- Path Finder-----##
def IdentifyMap(Exit):
 global MapInfo
 for i in range(len(ScienceBlock_Nodes)):
 if ScienceBlock_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(ScienceBlock_G)
 MapInfo.append(ScienceBlock_C)
 MapInfo.append(ScienceBlock_P)
 MapInfo.append("ScienceBlockChainging.jpg")

 for i in range(len(LanguageBlock_Nodes)):
 if LanguageBlock_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(LanguageBlock_G)
 MapInfo.append(LanguageBlock_C)
 MapInfo.append(LanguageBlock_P)
 MapInfo.append("LanguageBlockChainging.jpg")

 for i in range(len(EnglishAndMusicBlock_Nodes)):
 if EnglishAndMusicBlock_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(EnglishBlock_G)
 MapInfo.append(EnglishBlock_C)
 MapInfo.append(EnglishBlock_P)
 MapInfo.append("EnglishBlockChainging.jpg")

 for i in range(len(MathsBlockDownstairs_Nodes)):
 if MathsBlockDownstairs_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(MathsBlockDownstairs_G)
 MapInfo.append(MathsBlockDownstairs_C)
 MapInfo.append(MathsBlockDownstairs_P)
 MapInfo.append("MathsDownstairsChainging.jpg")

 for i in range(len(MathsBlockUpstairs_Nodes)):
 if MathsBlockUpstairs_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(MathsBlockUpstairs_G)
 MapInfo.append(MathsBlockUpstairs_C)
 MapInfo.append(MathsBlockUpstairs_P)
 MapInfo.append("MathsUpstairsChainging.jpg")

 for i in range(len(ComputingBlock_Nodes)):
 if ComputingBlock_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(ComputingBlock_G)
 MapInfo.append(ComputingBlock_C)
 MapInfo.append(ComputingBlock_P)
 MapInfo.append("ComputingBlockChainging.jpg")

 for i in range(len(GeographyBlock_Nodes)):
 if GeographyBlock_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(GeographyBlock_G)
 MapInfo.append(GeographyBlock_C)
 MapInfo.append(GeographyBlock_P)
 MapInfo.append("GeographyBlockChainging.jpg")

 for i in range(len(DTBlockDownstairs_Nodes)):
 if DTBlockDownstairs_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(DTBlockDownstairs_G)
 MapInfo.append(DTBlockDownstairs_C)
 MapInfo.append(DTBlockDownstairs_P)
 MapInfo.append("DtBlockDownstairsChainging.jpg")

 for i in range(len(DTBlockUpstairs_Nodes)):
 if DTBlockUpstairs_Nodes[i] == str(Exit):
 MapInfo.clear()#get rid of previous data
 MapInfo.append(DTBlockUpstairs_G)
 MapInfo.append(DTBlockUpstairs_C)
 MapInfo.append(DTBlockUpstairs_P)
 MapInfo.append("DtBlockUpstairsChainging.jpg")
```

```

def IdentifyNode(Room):
 global NodeInfo
 NodeInfo.clear()#get rid of previous data
 for i in range(len(NodeName_List)):#Looks for room the exit of the room from the node text file and adds the
 #exit's x and y cords to the current node info list
 if str(RoomExit_List[Room_List.index(Room)]) == str(NodeName_List[i]):
 NodeInfo.append(NodeX_List[i])#appends the x cord to node information list
 NodeInfo.append(NodeY_List[i])#appends the y cord to node information list

 if (RoomExit_List[Room_List.index(Room)]) == "G_ST1" or (RoomExit_List[Room_List.index(Room)]) == "G_ST2" or
 (RoomExit_List[Room_List.index(Room)]) == "C_ST1" or (RoomExit_List[Room_List.index(Room)]) == "C_ST2":#checks
 #to see if the room is upstairs in the main block
 NodeInfo.append(int(NodeX_List[NodeName_List.index('MN_EX')]))#appends the x cord to node information
 list
 NodeInfo.append(int(NodeY_List[NodeName_List.index('MN_EX')]))#appends the y cord to node information
 list

 if (RoomExit_List[Room_List.index(Room)]) == "LB_ST1" or (RoomExit_List[Room_List.index(Room)]) ==
 "LB_ST2":#checks to see if the room is upstairs in the language block
 NodeInfo.append(int(NodeX_List[NodeName_List.index('LB_EX1')]))#appends the x cord to node
 information list
 NodeInfo.append(int(NodeY_List[NodeName_List.index('LB_EX1')]))#appends the y cord to node
 information list

 if (RoomExit_List[Room_List.index(Room)]) == "M_ST1" or (RoomExit_List[Room_List.index(Room)]) ==
 "M_ST2":#checks to see if the room is upstairs in the maths block
 NodeInfo.append(int(NodeX_List[NodeName_List.index('M_EX1')]))#appends the x cord to node information
 list
 NodeInfo.append(int(NodeY_List[NodeName_List.index('M_EX1')]))#appends the y cord to node information
 list

 if (RoomExit_List[Room_List.index(Room)]) == "DT_ST1" or (RoomExit_List[Room_List.index(Room)]) ==
 "DT_ST1":#checks to see if the room is upstairs in the dt block
 NodeInfo.append(int(NodeX_List[NodeName_List.index('DT_E1'))]))#appends the x cord to node
 information list
 NodeInfo.append(int(NodeY_List[NodeName_List.index('DT_E1'))]))#appends the y cord to node
 information list

def CreatePath_Inside(Location,GoingTo):
 global pos1, pos2, MapInfo
 IdentifyMap(RoomExit_List[Room_List.index(str(Location))])#Identifies the inside map needed
 print(str(RoomExit_List[Room_List.index(str(Location))]))
 print(str(MapInfo[3]))
 print(str(MapInfo[2]))
 print(str(MapInfo[1]))
 cv2.imwrite(MapInfo[3],MapInfo[2])#restores chainging map back to it's orignal state
 pos1 = [int((RoomX_List[Room_List.index(Location)])),int((RoomY_List[Room_List.index(Location)])))]#rooms
 cords
 print(pos1)
 pos2 = globals()[RoomExit_List[Room_List.index(Location)]]#exit cords
 print(pos2)
 if GoingTo == True:
 find_path_alg(MapInfo[0], MapInfo[1] ,int(((MapInfo[1]).shape)[1]),int(((MapInfo[1]).shape)
 [0]),pos1,pos2, path_color)
 cv2.imwrite(MapInfo[3],MapInfo[1])#draws path onto chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load(MapInfo[3]),((((Center_X)+(MapInfo[1]).shape)[0])/2),(((Center_Y)-
 (int((MapInfo[1]).shape)[1]))/2)))#screenblits map with path onto the screen
 elif GoingTo == False:
 find_path_alg(MapInfo[0],MapInfo[1],int(((MapInfo[1]).shape)[1]),int(((MapInfo[1]).shape)[0]),pos2,pos1,
 path_color)
 cv2.imwrite(MapInfo[3],MapInfo[1])#draws path onto chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load(MapInfo[3]),((((Center_X)+(MapInfo[1]).shape)[0])/2),(((Center_Y)-
 (int((MapInfo[1]).shape)[1]))/2)))#screenblits map with path onto the screen

def CreatePath_Outside(StartLocation,EndLocation):
 global pos1, pos2,NodeInfo
 cv2.imwrite("MainMapChainging.jpg", MainMap_P)#reverts chainging map to it's orignal state
 IdentifyNode(StartLocation)
 print(NodeInfo)
 pos1 = [int(NodeInfo[0]),int(NodeInfo[1])]# cords of the starting room's building
 IdentifyNode(EndLocation)
 pos2 = [int(NodeInfo[0]),int(NodeInfo[1])]#cords of the end room's building
 find_path_alg(MainMap_G, MainMap_C ,int((MainMap_C.shape)[1]),int((MainMap_C.shape)[0]), pos1 ,pos2 ,
 path_color)#creates path from cords
 cv2.imwrite("MainMapChainging.jpg", MainMap_C)#path drawn onto the chainging map
 Screen.fill("black")
 Screen.blit(pygame.image.load("MainMapChainging.jpg"),((Center_X)/4, 0 - (Height)/2))#screens blits map with
 path on it

```

```

----- Edited CocoXiao1998's Path Finder-----

def exists(Blank_Img, x, y, width, height):
 if x < width and x >= 0 and y < height and y >= 0:
 return True
 else:
 return False

checks if the node is traversable
def is_traversable(Blank_Img, x, y):
 r = Blank_Img[y][x][0]
 g = Blank_Img[y][x][1]
 b = Blank_Img[y][x][2]

 if r >= 250 and g >= 250 and b >= 250:
 return True
 else:
 return False

checks if the destination has been reached
def is_destination(Blank_Img, x, y, ending_position):
 if x == ending_position[0] and y == ending_position[1]:
 return True
 else:
 return False

calculates heuristics (diagonal)
def heuristics(Blank_Img, x, y, ending_position):
 # getting V/H distance
 d1 = 10 # 1 * 10
 x_end = ending_position[0]
 y_end = ending_position[1]
 x_distance = abs(x - x_end)
 y_distance = abs(y - y_end)

 # getting diagonal distance
 d2 = 14 # sqrt(2) * 10
 diag_distance = abs(x_distance - y_distance)

 h = min(x_distance, y_distance) * d1 + diag_distance * d2
 return h

finds the optimized path using the parent coords in node details
sets the path to the color green
sets the start and end to the color red
def print_path(Colorued_Img, starting_position, ending_position, path_color, node_details):
 endpoint_color = (255, 0, 0)
 start_x = starting_position[0]
 start_y = starting_position[1]
 end_x = ending_position[0]
 end_y = ending_position[1]
 parent_x = node_details[end_x][end_y][3]
 parent_y = node_details[end_x][end_y][4]
 Colorued_Img[parent_y][parent_x] = path_color

 # will loop until the parent x and y are the starting position
 while True:
 if parent_x == start_x and parent_y == start_y:
 Colorued_Img[start_y][start_x] = endpoint_color
 Colorued_Img[end_y][end_x] = endpoint_color
 return

 # getting the parent coords and then setting the coords to the path color
 temp_x = parent_x
 temp_y = parent_y
 parent_x = node_details[temp_x][temp_y][3]
 parent_y = node_details[temp_x][temp_y][4]
 Colorued_Img[parent_y][parent_x] = path_color

```

```

A* alg
def find_path_alg(Blank_Img, Coloured_Img, width, height, starting_position, ending_position, path_color):
 # If the user chose the starting and ending position to be the same
 if is_destination(Blank_Img, starting_position[0], starting_position[1], ending_position):
 print("The starting and ending position you have chosen is the same!")
 print("Exiting program.")

 # creating a list to hold the f, g, and h values and the parent coords for each node
 # i.e. for f, g, h values and parent x, y values
 node_details = np.full((width, height, 5), -1)

 # creating a 2D-list that marks all nodes that have been evaluated by 1, 0 if not
 closed_list = np.zeros((width, height))

 # creating an open list that will contain nodes with calculated f costs
 # each index will contain [f, (x, y)]
 open_list = []

 # putting the starting position in open list with its f as 0
 open_list.append([0, [starting_position[0], starting_position[1]]])

 while len(open_list) != 0:
 # get current equal to node with the lowest f-value that blockStartsToA_End
 if len(open_list) == 1:
 current = open_list[0]
 else:
 current = open_list[0]
 for node in open_list:
 node_x = node[1][0]
 node_y = node[1][1]
 cur_x = current[1][0]
 cur_y = current[1][1]

 if node[0] < current[0]:
 current = node
 elif f-values are equal:
 # setting current equal to node if the h-value is lower
 node_h = node_details[node_x][node_y][2]
 current_h = node_details[cur_x][cur_y][2]
 if node_h < current_h:
 current = node

 # current node coords vars
 cur_x = current[1][0]
 cur_y = current[1][1]

 # removing current node from open list
 open_list.remove(current)

 # marking node as 1 in closed list
 closed_list[cur_x][cur_y] = 1

 # checking if current node is the destination
 if is_destination(Blank_Img, cur_x, cur_y, ending_position):
 print("Path was successfully found!")
 print_path(Coloured_Img, starting_position, ending_position, path_color, node_details)
 return

 # getting the neighbor nodes
 else:
 neighbor_list = [(cur_x - 1, cur_y), (cur_x + 1, cur_y), (cur_x, cur_y + 1), (cur_x, cur_y - 1),
 (cur_x - 1, cur_y + 1), (cur_x - 1, cur_y - 1), (cur_x + 1, cur_y + 1), (cur_x + 1, cur_y - 1)]
 direction_counter = 0

 for neighbor in neighbor_list:
 # neighbor node x and y vars
 n_x = neighbor[0]
 n_y = neighbor[1]

 # check that neighbor exists, is traversable and is marked as 0 in closed list
 if (exists(Blank_Img, n_x, n_y, width, height) and
 is_traversable(Blank_Img, n_x, n_y) and
 closed_list[n_x][n_y] == 0):
 # checking if current node is starting position in order to get right value g
 if current[1] == starting_position:
 if direction_counter <= 3:
 g = 10
 else:
 g = 14
 else:
 if direction_counter <= 3:
 g = node_details[cur_x][cur_y][0] + 10
 else:
 g = node_details[cur_x][cur_y][0] + 14

 # getting other values
 h = heuristics(Blank_Img, n_x, n_y, ending_position)
 f = g + h

 # checking if neighbor needs to be updated
 # checking if f-cost to neighbor is shorter or if neighbor not in open list
 update = True
 in_open = False
 index = -1
 count = 0
 for node in open_list:
 node_x = node[1][0]
 node_y = node[1][1]
 if n_x == node_x and n_y == node_y:
 in_open = True
 index = count
 if f >= node[0]:
 update = False
 count += 1

 # update node details and updating open list
 if update == True:
 # starting details in node details
 node_details[n_x][n_y][0] = f
 node_details[n_x][n_y][1] = g
 node_details[n_x][n_y][2] = h
 node_details[n_x][n_y][3] = cur_x
 node_details[n_x][n_y][4] = cur_y

 # if neighbor is not in open, then append neighbor, else just update f-value
 if in_open == False:
 open_list.append([f, [n_x, n_y]])
 else:
 open_list[index][0] = f

 direction_counter += 1

 #-----Running -----#
 Start_Sc()

elif event.key == pygame.K_KP_ENTER:#checks if enter button is being pressed

```