

Student Name: Toma Mirchevska NSN:

Date:

AS 91891 & AS91896 & 91897 (15 credits)

2.2A Apply conventions to develop a design for a digital technologies outcome (3 credits)

2.7A Use advanced programming techniques to develop a computer program (6 credits)

2.8A Use Advanced Processes to develop a digital technologies outcome (6 credits)

1- Requirement Analysis	2
-------------------------	---

2- Discover and Plan	3
----------------------	---

2.1 Purpose	3
2.2 End Users	3
2.3 Plan	3
2.4 Planning and Project Management Tool	4
2.5 Components of my Project	4
2.6 Implications Relevant to My Outcome	5

Check Point 1: Planning Sign off:	5
-----------------------------------	---

3. Design	5
-----------	---

3.1 Investigate Design relevant conventions (Convention Analysis - Compare)	
3.2 Design Ideas (Generate at least 3)	5
3.3 Feedback and improvements on Design ideas	6
3.4 Selected Design and its appropriateness	6

4. Develop and Test Components	6
--------------------------------	---

Component 1:	7
--------------	---

Trialling:	7
------------	---

Testing	7
---------	---

How I addressed Relevant Implications:	7
--	---

How I applied design conventions	7
----------------------------------	---

Version Control	7
-----------------	---

Component 2:	7
--------------	---

Trialling:	7
------------	---

Testing	8
---------	---

How I addressed Relevant Implications:	8
--	---

How I applied design conventions	8
----------------------------------	---

Version Control	8
-----------------	---

Component 3:	8
--------------	---

Trialling:	8
------------	---

Testing	8
How I addressed Relevant Implications:	8
How I applied design conventions	9
Version Control	9
Check Point 2	9
5. Final Program Structure	9
6. Final Outcome Testing	9
6.1 Did you comment on your code and described what it does?	9
6.2 Did you check you followed naming conventions?	9
6.3 Overall final program testing	9
7. Usability Testing and Feedback	10
8. Evaluation	10
8. Acknowledge own work	1

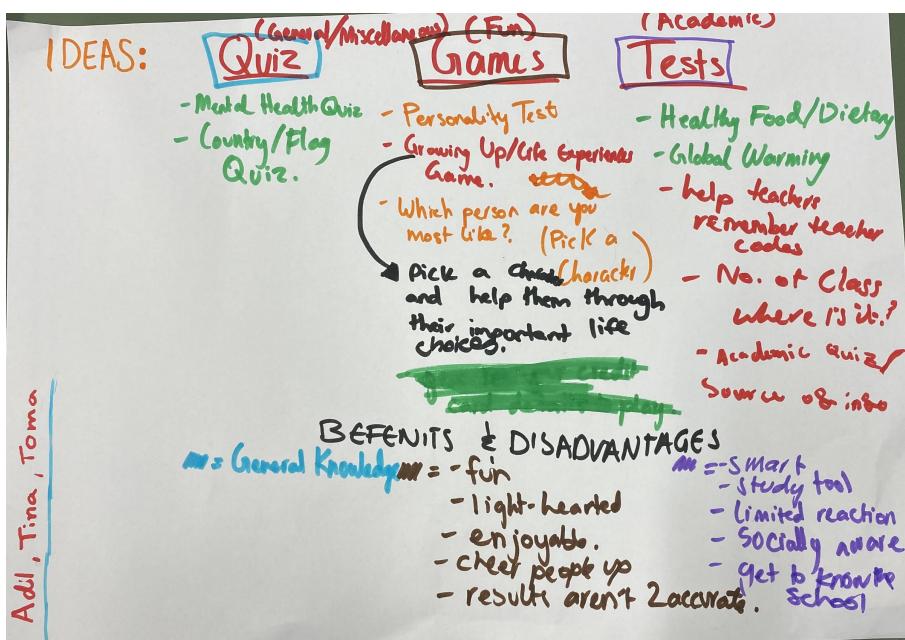
1- Requirement Analysis

- Plan (need to create a program with an engaging way to help the students/school community learn about something important).
- Design (Design the programme that follows the purpose and is effective).
- Use advanced programming techniques (objects and classes AND user interface).
- (code uses variables storing at least two types of data (e.g. numeric, text, Boolean)uses sequence, selection and iteration control structures that take input from a user, sensor(s), or other external source(s)produces output).
- Trial and test code.
- Use advanced programming techniques (objects and classes AND user interface).

2- Discover and Plan

Group brainstorm of GUI ideas:

- This brainstorm was formed from a group discussion in class where we each had inputted ideas and from the ideas we elaborated a bit more on what the advantages and disadvantages are e.g. would making a programme on this topic benefit society or not.



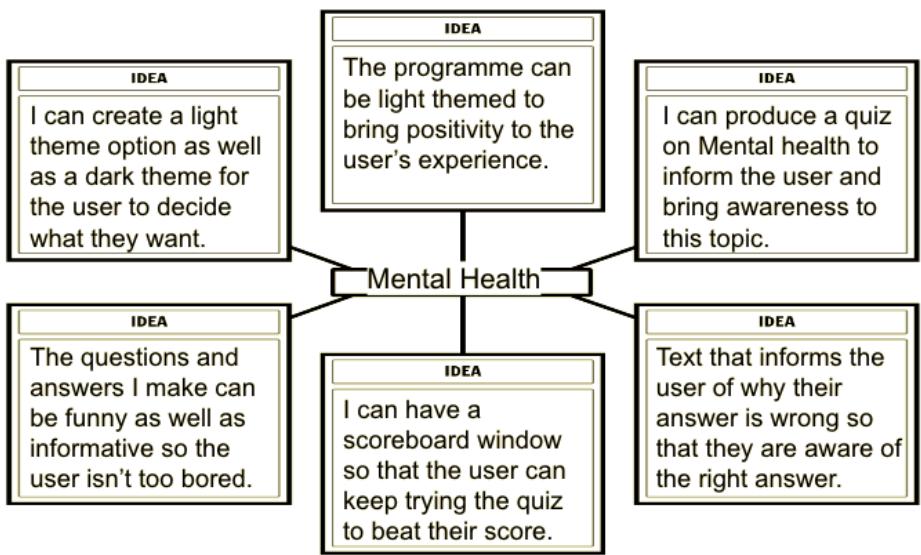
PMI chart of the ideas:

Options:	Plus	Minus	Interesting
Personality test	Fun to do. Many questions to input (original questions).	Too many personality types to try to implement in the results. Might not satisfy users because I would use limited personality types.	Many people, especially teenagers, are attracted to these types of quizzes which could really be good if my end users would be very interested in doing my programme.
Mental health quiz	Very eye opening and informative. Can help society become	It may be irrelevant to people so they wouldn't want to use my programme.	This would really help the societal problem with mental health issues amongst teenagers, the programme can be directed to them so that they can become aware and find coping methods for their

	aware of mental health.		wellbeing.
General knowledge quiz	Very casual Helps to teach fun things that many people should or might want to know.	It's too common (many quizzes are already formed on general knowledge) meaning there would be little questions to write up that would be unique and already overused.	Since it is very popular many people could be attracted to using my programme.

Individual Brainstorm - Mental Health:

- In this brainstorm I have decided that my programme (quiz) will be based around Mental Health and I have elaborated on what my purpose would be and other aspects of the topic.



2.1 Purpose

I have chosen the context of Mental Health issues as it is a large growing problem in New Zealand society amongst teenagers. I am planning to produce a quiz that will bring awareness to mental health and provide different strategies for the users. The user will try to answer the questions correctly for a higher score.

2.2 End Users

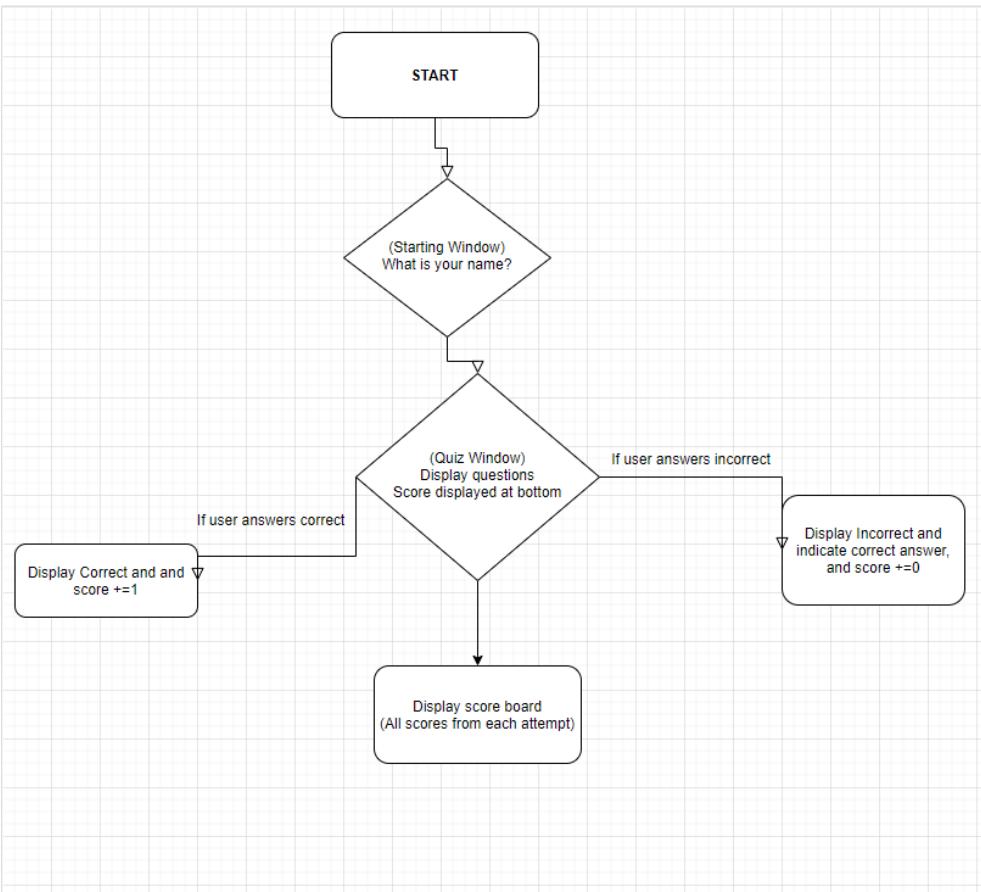
My programme is directed towards teenagers and adults to bring them awareness about mental health. I have sent out a form that will help guide my programme to suit the users opinions.

I created two google forms and sent it out to the Year 12 class of Computer Science. I have received results that show me the users interests and would guide me with how I should create my programme.

<p>Do you think mental health is important?</p> <p>A pie chart with three segments: a large blue segment labeled 'Yes' (66.7%), a small orange segment labeled 'Don't have an opinion' (16.7%), and a small red segment labeled 'Nope' (16.7%).</p> <ul style="list-style-type: none">YesNopeDon't have an opinion	<p>From these results I can say that my selected topic 'Mental Health' is important to the majority of the users and therefore I will continue working on my programme according to this topic.</p>
<p>If I did a quiz on mental health would you like it to be fun or informative?</p> <p>A pie chart with three segments, each labeled 33.3%: a blue segment labeled 'Fun', a red segment labeled 'Informative', and an orange segment labeled 'Don't have an opinion'.</p> <ul style="list-style-type: none">FunInformativeDon't have an opinion	<p>The quiz will be created with informative and entertainment purposes as 50% of the responses preferred one of each option. For my quiz to appeal to all the users I will implement both informative and entertaining questions/answers.</p>
<p>Which colour idea is most aesthetically pleasing to you?</p> <p>A pie chart with five segments: two large blue segments (36.4% each), one large red segment (36.4%), and two small purple segments (9.1% each).</p> <ul style="list-style-type: none">Bright coloursDark themeContrastingColourfulpastel	<p>Majority of the responses prefer a dark or light themed window and to validate their preferences I have decided to provide a light theme option and dark theme option so that the user is free to select their desired format colour.</p>
<p>What is the best answering technique for you?</p> <p>A pie chart with five segments: a large blue segment labeled 'Multiple choice' (70%), and four small segments labeled 10% each: 'True or false', 'Inserting own answers', 'a quiz', and 'A mixture of option one and two'.</p> <ul style="list-style-type: none">Multiple choiceTrue or falseInserting own answersa quizA mixture of option one and two	<p>The users also wanted a multiple choice quiz which I have decided to do but to account for the small number of users wanting true or false questions I have also decided to implement a few of them to satisfy all the users with my program.</p>

2.3 Plan

Simple flowchart of programme:

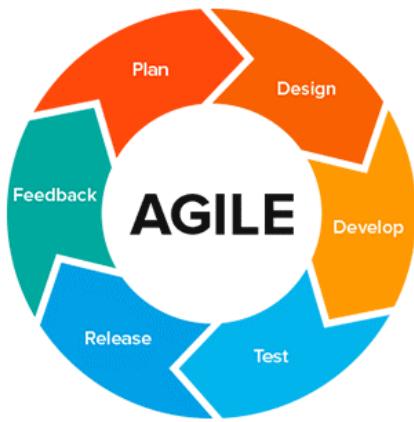


2.4 Planning and Project Management Tool

Agile Methodology:

This method of project management is used to provide a system for documentation of my programme. It is characterised by having a format such as: 1) Planning, 2) Design, 3) Develop, 4) Test, 5) Evaluation, and 6) Feedback. I can follow this format for my documentation which will allow me to work efficiently to produce my programme while documenting. This method of project management makes it easier for me to go back through

my work to find each section and check I have completed my checklists and also gives me guidance along the way of producing my programme such as what I should work on next.



Version Control:

For version control I am able to use a website called GitHub which is connected to the python programme Repl.it. This project management tool allows me to commit different changes made in my code throughout the whole time I am producing my programme. This means I can efficiently save different versions of my code without spending more time explaining my varied versions of the programme.

https://github.com/TomaMirchevska/Toma_Quiz

Organize your issues with project boards

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.

Learn More

Sort tasks

Add issues and pull requests to your board and prioritize them alongside note cards containing ideas or task lists.

Plan your project

Sort tasks into columns by status. You can label columns with status indicators like "To Do", "In Progress", and "Done".

Automate your workflow

Set up triggering events to save time on project management—we'll move tasks into the right columns for you.

Track progress

Keep track of everything happening in your project and see exactly what's changed since the last time you looked.

Share status

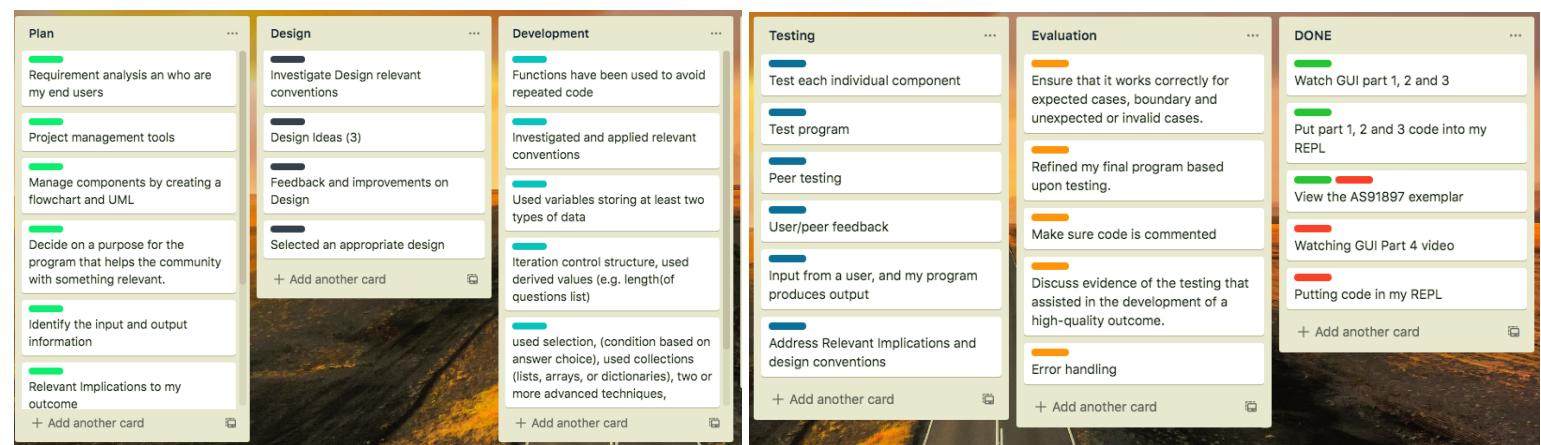
Each card has a unique URL, making it easy to share and discuss individual tasks with your team.

Wrap up

After you wrap up your work, close your project board to remove it from your active projects list. On to the next project!

Trello board:

Trello board is an online project management tool which I can use to keep track of what necessary tasks I need to fulfill and what tasks I have already completed.



2.5 Components of my Project

COMPONENTS (Windows) AND OTHER PARTS	
Questions	
question_answer	Dictionary: Contains all the possible questions and answers associated with them. It will create a more efficient experience with the programme and less code as I can just configure the question using the dictionary.
Window	
MenuPage	GUI
DarkMenuPage	GUI
Contains	
Name collection	Method: collects the user's name as variable through the entry box. Output will be displaying the name with score on the scoreboard.
name	It is a variable used to store the user's input as their name, it will be appended

	to a names_list and stored in a new file for the scoreboard.
Name check	Method: Make sure the name entered contains a reasonable amount of letters and no numerals/special characters/punctuation.
Window	
QuizPage	GUI
DarkQuizPage	GUI
Contains	
Answer check	Selection: Checks that the answer selected by the user is correct or incorrect.
score	Variable: It will keep track of how many questions the user got right by calculations such as if correct: score +=1, if incorrect: score +=0.
Window	
ResultsPage	GUI
DarkResultsPage	GUI
Contains	
Score board	Method: Displays user's score results for the quiz. Stores previous user results.

Planned questions and answers:

Dictionary: question_answer	
Questions	Answers
<p>1. It doesn't matter how much sleep you get, you can always function your best.</p> <p>Resources: https://www.cdc.gov/healthyschools/features/students-sleep.htm</p>	<p>a) True.</p> <p>b) False. 8 - 10 hours is recommended.</p> <p>c) It depends on the day.</p> <p>d) Any amount of sleep is okay as long as you have naps in the day.</p> <p>(Varied age groups have different times recommended but from 13 - 18 years old, 8 - 10 hours sleep is suggested).</p>

<p>2. What is the difference between mental health and mental illness?</p>	<p>a) There isn't a difference. b) Mental illnesses are untreatable whereas mental health can be cared for. c) Illness is branched from mental health and can affect a person's performance, mental health is the state of your emotional well being. d) Mental health is physical wellbeing whereas mental illnesses affect the brain's ability to function.</p> <p>(Mental health is like a tree and mental illness branches off of it.).</p>
<p>3. What type of people are affected by mental health issues?</p>	<p>a) Usually teenagers. b) Old people. c) Depressed kids. d) All of the above.</p> <p>(Everyone endures life challenges and therefore anybody can be affected by issues of mental health).</p>
<p>4. How should you approach someone that looks upset?</p>	<p>a) Be sincere and ask if there's anything you can do for them, show your sympathy and let them have space. b) Punch them, pull their hair and push them to the ground saying "Stop being a cry baby!". c) Insist on helping them with anything you think you can, even if they say no. d) Avoid approaching them, give them their space and let them deal with it on their own.</p> <p>(Along with respecting boundaries, we should reach out to the person.).</p>
<p>5. How many people have been affected by mental health in New Zealand?</p>	<p>a) 10. b) 50%. c) 1 in 4 people. d) Everyone.</p> <p>(Everyone has had a point in time where they were very low or very happy.).</p>
<p>6. What is a good way to cope with poor mental health?</p>	<p>a) Eat, eat, eat and eat the feelings away. b) Take time to enjoy things such as hobbies, things you are good at and talk to someone if you feel it would help release the stress. c) Try to stay occupied by school work and join lots of social clubs. d) Involve yourself more in class or sports and push yourself to the limits to distract you from the problem.</p> <p>(Take a break from everything, even with a friend by your side).</p>
<p>7. Which Mental health issue affects people the most? Resources:</p>	<p>a) Paranoia (distrust of others or feeling like someone is after you). b) Depression (persistent upset emotions or lack</p>

<p>https://www.talkspace.com/blog/the-top-five-most-common-mental-illnesses/</p>	<p>of interest in things, significantly affecting life activities).</p> <p>c) Eating disorder (abnormal/unusual and unhealthy eating habits). d) OCD (Obsessive compulsive disorder, repeating behavioural habits due to immoderate thoughts). (Depression is the most common issue which is suffered by people, majority being teenagers).</p>
<p>8. What is the best thing to do when you see someone enjoying their day?</p>	<p>a) Nothing, or vibe with them. b) Ruin their day. c) Punch them. d) Make jokes and laugh at them with friends. (You don't necessarily have to do anything, just let them be happy or join in!).</p>
<p>9. Do people who suffer from serious mental issues deserve special treatment?</p>	<p>a) No, I think we should punch them. b) Yes, it's nearly impossible to live through life with such difficult problems. c) Acknowledge their struggles and don't persist in bringing them up. Sometimes it's best to treat them like you should treat anyone, with respect and not like a child. d) Treat them with anything you can such as helping them grab their food at lunch or helping out with school work. Never let them be on their own to do things because they need help. (It's good to be nice, but not controlling over peoples' lives).</p>
<p>10. What shouldn't you do to someone who is having a rough time with their mental wellbeing?</p>	<p>a) Give them coffee. b) Suggest different ways of distractions such as colouring a book. c) Offer them to go out into a better space or even go for a run with them. d) Punch them. (Just don't punch them).</p>

2.6 Implications Relevant to My Outcome

Intellectual Property:

The relevant implication 'Intellectual Property' relates to the legal rights a producer has over their intangible assets to keep it safe for a certain amount of time. This means that any user would have to have consent from the owner and give credit when using their creations.

This implication is relevant to my code as I intend to use quotes from other people as a part of my programme which will be displayed for the users to read.

In order to address the Intellectual Property implication I will be referencing each quote with the original author so that they can take credit as well as the user knowing who came up with it.

Usability:

The relevant implication 'Usability' relates to the aspects of the programme that appropriately support users to correctly do tasks with the programme and follow conventions efficiently.

This implication is relevant to my code as usability is relevant to all programs since users are performing actions with the window. A main part of my programme that I would need to consider this implication for is where the user needs to select options of the answers when going through the quiz section of my code. Not all people tend to do quizzes especially online which is why I should clarify how to use the quiz so the programme is easier and more efficiently used by the user.

I will attend to this implication by providing error messages in relation to the 'recognize, diagnose, and recover from errors' heuristic. If the user doesn't understand that they need to select an answer and they press continue, there will be an error message informing the user to select one of the options.

Accessibility:

The relevant implication 'Accessibility' relates to the specific disabilities or roadblocks that users face while using the programme and finding a way to provide easier accessibility for the users.

This implication is relevant to my code as many people are nearsighted, meaning with smaller fonts and picture sizes it would be harder for them to interact with my programme and follow through with actions.

To address this implication I will ensure that my font and button sizes are reasonably large so that the programme is much more accessible to users so they can efficiently perform tasks without the struggle of not fully registering what's on the page.

Functionality:

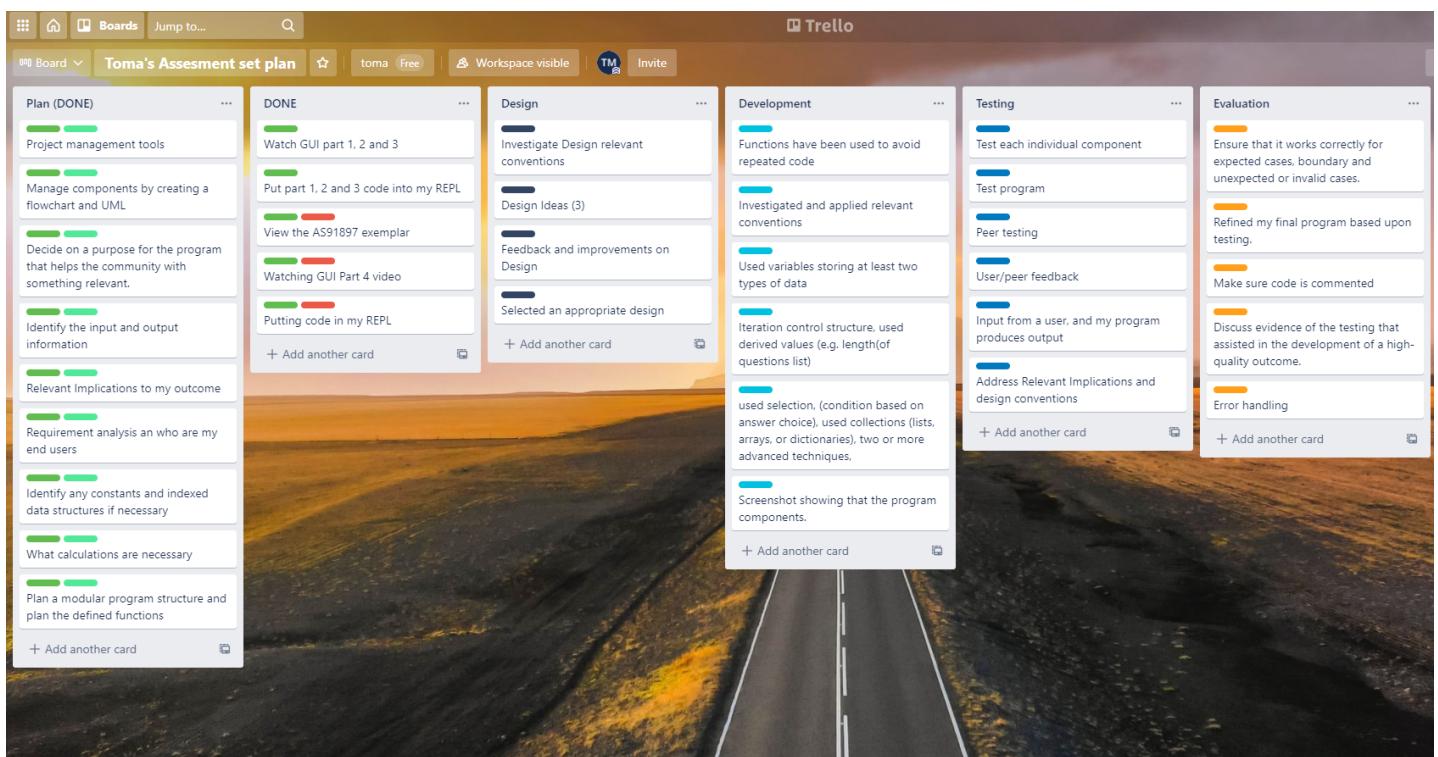
The relevant implication 'Functionality' relates to how functional an interface is for its end users and if the outcome suits what the user wants.

This implication is very relevant to my programme as I need to ensure the user can use my programme efficiently and that it's simply understood by my end users. If my quiz isn't functional then it wouldn't produce an appropriate outcome for the users which defeats the purpose of my programme.

To address this implication I will make sure my layout of buttons is placed reasonably where the user can navigate around the page easily and that I have no misleading content (all buttons should be labelled clearly or what it functions to do). This will ensure the user has a good experience with my quiz as the widgets will be accessible to them so that the user can do what they want.

Trello board progress:

- I have used my trello board as a project management tool to keep track of the tasks I have completed. So far I have completed the basic plan section and now can move on to design.



3. Design

Naming conventions and naming schemes:

Class - For any class I create in my programme, I will follow the naming convention of always starting the class name with a capital letter e.g. QuizPage(). If there are two words in the class name I will use camel casing where each word starts with a capital letter and no spaces between to show it is a class.

Variable - When I create a variable in my code I will use lower case letters to show it is a variable and use underscores to separate two different words for one variable e.g. question_number. If I have a variable with a constant value I will use the UPPER_CASE naming scheme to show it stays the same e.g. PRICE = 5.

Functions/methods - Similar to variables, I will use lower case naming scheme to name my functions and methods in the classes throughout my code to differentiate it from other names e.g. score_calculations().

Usability Heuristics:

- For my programme I will be following Jakob Nielsen's Usability Heuristics to produce a high quality outcome for my end users. These heuristics are a method for examining my interface design and to see whether or not it benefits end user experience the best the programme can. Researching each specific heuristic will help me while making my programme as it will allow me to produce an outcome that benefits my end users which is my main goal.

Visibility of System Status:

The first implication I'm going to look at is 'Visibility of System Status'. According to nngroup.com the definition of Visibility of System Status is 'The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.' That means that in my program, there should be some sort of feedback to the user to make sure they are aware of what is going on at all times. This is important as otherwise, users might start to lose touch with the program and not have a good experience, and that's caused by the user's need to feel satisfied when they are able to understand what consequence their action makes on the window.

The simplest way I could apply this heuristic is by creating a hover effect on each button that the user can select and a change in cursor design so that when the user hovers over any buttons it informs them that they can press the button and also provides satisfying feedback. Another way I could implement this heuristic is by adding a loading bar when my program is loading data up.

Diagnose and recover from errors:

Another heuristic I could focus on with my program is diagnose and recover from errors which is defined as providing suitable messages to help the user recover from an error. I can apply this in my code by providing error messages that clearly state where the user has gone wrong so that they can efficiently resolve the issue.

User control and freedom:

This heuristic is about giving the user the opportunity to exit or have more control over the program so that they also have more freedom to do what they want. I can apply this heuristic into my code by providing back and exit buttons for the user so that they can be free to navigate through the window.

Consistency and Standards:

Ux planet says that Consistency and Standards means: In a simple term, a system or a product should never ever confuse the users by using different words, actions, design, or situations to derive the same meaning. To apply this heuristic in my programme, I should place certain buttons in familiar places for each window similar

to other programs e.g. exit button in the bottom left and continue button in the bottom right and keep it consistent for the page so it meets the standard of other pages.

Relevant composition methods:

- I have done research on multiple different design aspects for a user interface so that I can pick out the most suitable design for my programme which will also produce a good outcome for my end users. All of this research will be put towards choosing my window designs.

Colours:

In a window, colour helps to communicate to users and is a crucial part of visual language. The variety of colours can evoke different feelings in the user which I can focus on to benefit their experience with my code. Red - Red tends to imply importance or an error in the window which makes the user feel that a certain part of the window or text is most relevant. I could use red font colours for error messages so that the users realise the text is very important to read and therefore more efficiently recover from the error.

Blue - The colour blue can provide comfort and also sadness for the user when using my program, a lighter blue can evoke calmness for the user whereas darker blue can make the user upset. A lighter blue colour would be very good to implement in my program as all users should feel comfort while interacting with my window.

Yellow - Commonly yellow is recognised as a happy colour and can bring joy to a user when interacting with my programme. If I want to create a very bright and joyful space for the user then I could use the colour yellow in my code.

Green - Green can be associated with success or being correct. This colour is commonly used when users perform a task correctly to notify them that they are right, I could implement this colour in my code as well when a user gets an answer correct so that the user understands they did something right.



Contrast:

Colour contrast is important for each window to differentiate the content on my page and especially indicates the most important parts to the user on each window. Contrast means having something be distinct from another thing so that it is visible to the user. My window should have darker colours paired with lighter colours so that my text and buttons stand out from the background of my windows. For my light themed windows I intend to use bright colours as the most standing out colours and dark colours to contrast so that the content stands out to the user, vice versa for my dark themed windows.

Alignment And Proximity:

The alignment of my content in the programme is important as it should create a clear and not cluttered space on my window so that the user can easily read and navigate around the programme. My related text should be in the same alignment which can be to the centre, left or right. This is to ensure the user doesn't get lost around the page and also doesn't miss any important information that isn't presented in the right place.

Proximity is important as well relative to alignment as it gives each text and content padding space between each other. I would have to provide space between all my texts so that it shows the user that there's a different context and also doesn't clutter all the content into one space. If I didn't do this it would be hard for the reader to comprehend what is happening or what is being said on my window.

Size of the window:

A reasonable size for my window would be to take up the majority of the user's screen so that all the information is set out clearly and not clumped in a small amount of space. This also allows the user to have visual space to comprehend the program and have focus on my quiz without other tabs being distracting by exceeding the size of my own program.

Text Font:

I will make sure my program design is accessible for users with near sightedness therefore all font sizes should be large enough to read and the font should be eligible for all users. I have researched that the top clearest font to read is Helvetica, PT Sans, PT Serif and Open Sans. Therefore all font sizes will not recede past 12 and the font type will be 'Helvetica' as it is the most simple and clear text and eligible for users so they can understand the programme.

Window Colours:

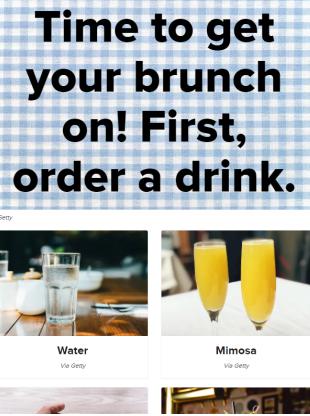
To avoid the issue of users (specifically suffering colour blindness) not clearly registering what is being said on the page, which relates to the relevant implication accessibility, I will make sure my background and font colours contrast. I also have

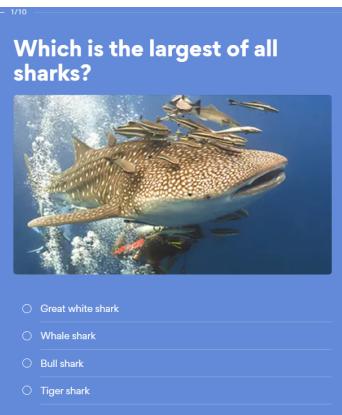
titles with backgrounds as labels so that they stand out to the users. As well as that I have made sure the background colours are contrasted to the content/text used so that the user can read clearly.

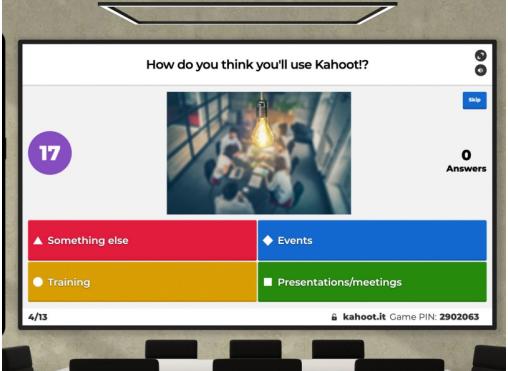
PMI Charts

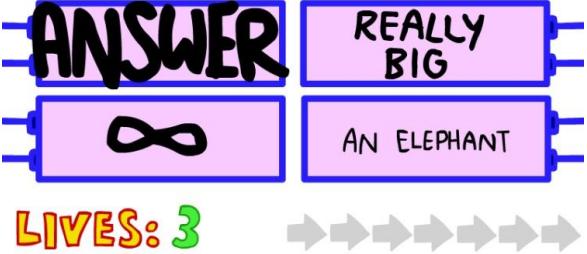
Design layouts:

Here I have researched different question page layouts from public quizzes and analysed the aspects of them.

BuzzFeed	Interesting	Positive	Minus
<p>BuzzFeed News Quizzes Shopping TV & Movies Videos Test</p> <p>Time to get your brunch on! First, order a drink.</p> 	<p>The format is very unique and uses mainly visual techniques rather than wording everything.</p>	<p>The format is large and very easy to register for all users. Pictures allow us to quickly make a decision without having to read.</p>	<p>The images are cluttered close together which can be confusing.</p>

National Geographic	Interesting	Positive	Minus
<p>1/10</p> <p>Which is the largest of all sharks?</p> 	<p>They made the image very large and the answers small, this is something unusual to me which I also didn't like.</p>	<p>The image makes the quiz look less boring. The format is common meaning the user is more familiar with how to navigate around the programme.</p>	<p>The colour scheme doesn't work well together as there isn't much contrast.</p>

Kahoot	Interesting	Positive	Minus
	There are a variety of colours that work well together and there's also a timer which isn't very common.	The format is very clear and simple. Image brings flavour to the window.	The options are too close together.

Impossible Quiz	Interesting	Positive	Minus
<p>7. THE ANSWER IS REALLY BIG</p> 	The design ideas are so unique that I can recognise the quiz anywhere individually without reading the title.	It's a very unique way to display the labels and it creates a humorous experience.	Some words can be non eligible for others.

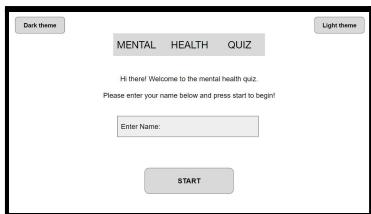
3.2 Design Ideas

Wireframing:

- I have used all my research in the previous section to apply to my wireframing and design an expected layout for my code, according to design conventions and user requirements.

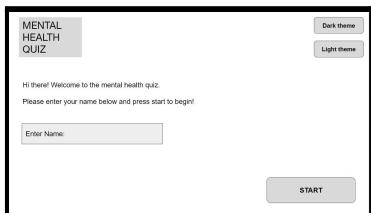
Window 1 (Dark and light theme):

Options: 1, 2 and 3	What conventions are followed	Why I selected them (end users)
---------------------	-------------------------------	---------------------------------



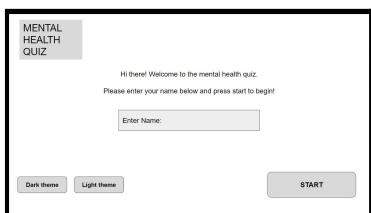
I put the majority of context in the middle of the page, centre aligned and large highlighted title, padding between text.
The start button is big and at the bottom Contrast.

This is so the user's attention is caught by the quiz as the centre of the page is most commonly looked at by users, all content has padding between each other so is clear to read and alignment shows the text is all related and easier to follow for the user. Because commonly a button to move onto the next window would be placed at the bottom of the page. There is contrast (for the light theme there is dark text and light background or vice versa for the dark theme) between the background and all text on the page meaning the content is very distinct and the user will be able to read everything clearly.



I put the logo of my quiz at the top left corner of the window.
The start button is in the bottom right.

I did this because it is a human tendency to look at the top left of a page first and therefore the user will see my title first. The bottom right corner tends to be an option to continue or go next so I placed the start button there for familiarising the programme with others.

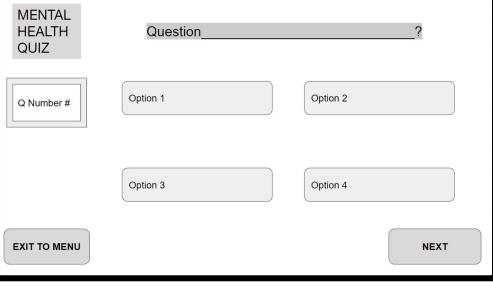
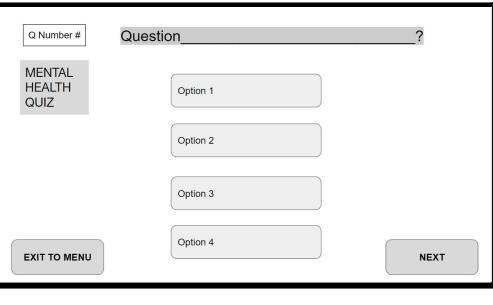


I placed my quiz logo in the top left corner and the entry box in the middle of the page.

The logo is placed in the top left corner as its human tendency to look at the top left to see the title.
The entry box is placed in the middle to be clearly displayed for the user to see.

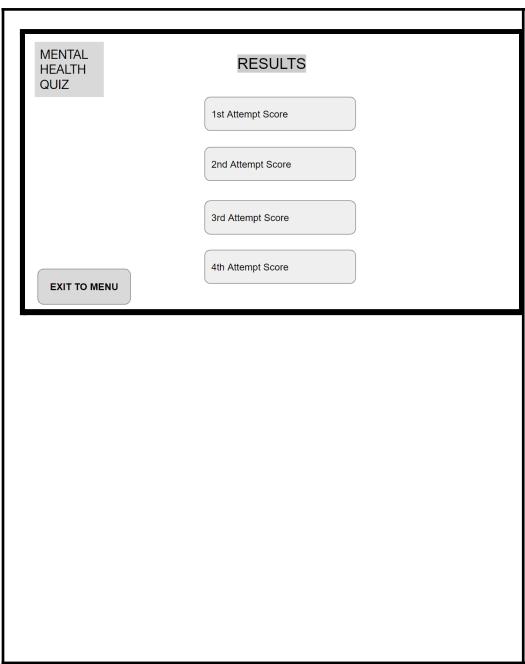
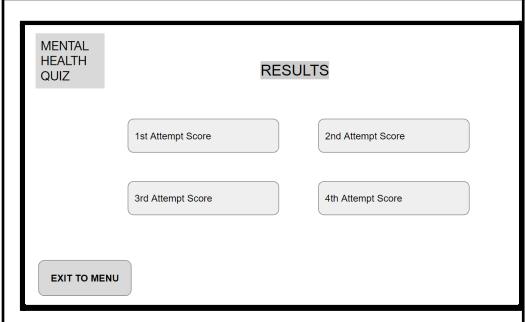
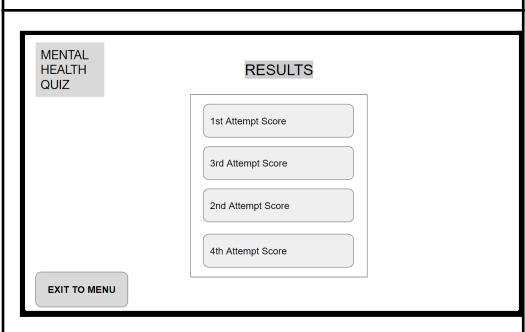
Window 2 (Dark and light theme):

Options: 1, 2 and 3	What conventions are followed	Why I selected them (end users)
	<p>MENTAL HEALTH QUIZ</p> <p>Question _____ ;</p> <p>Q Number #</p> <p>Option 1 Option 2</p> <p>Option 3 Option 4</p> <p>EXIT TO MENU NEXT</p>	<p>Logo in the top left corner. Question displayed at top. Answer options are aligned centre and right. Exit button bottom left. Next button bottom right. There is padding between each text and button.</p> <p>The logo is placed in the top left so that the title doesn't interfere with the questions yet the user still knows they are participating in my mental health quiz. Question is at the top of the window because the user will automatically look at the top for instructions. Answers are in this format because it is commonly used in other quizzes and evenly takes up space on the window. Exit button tends to be at the bottom left in</p>

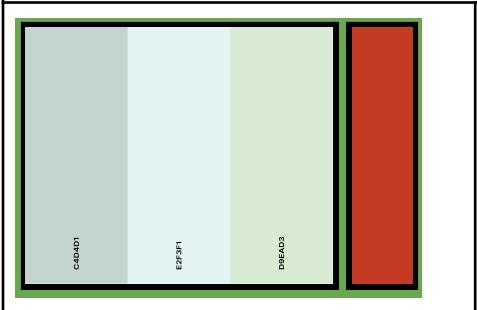
	Contrast.	<p>all programmes and is unconsciously known by users so I decided to keep this familiar format. As well, the Next button is placed bottom right as it is commonly placed there in other quizzes. To do with alignment and proximity I have placed the answer buttons in a centre and right alignment and padding between each one so that the content isn't compact and the user can easily register the content. The contrast (for the light theme there is dark text and light background or vice versa for the dark theme) between the content and background makes the text stand out and easier for the user to understand.</p>
	<p>Smaller answer buttons, horizontal form. Larger and bordered question number. More padding for buttons.</p>	<p>The buttons are aligned to the centre and right and have more padding relative to proximity to create more space between each answer and therefore is easier to differentiate and read by the user. Question number is bordered to separate itself from other content and also indicates it is a numeric value to the user that it isn't a random text on the page.</p>
	<p>Centre alignment of answer buttons. Q Number above logo. Smaller padding.</p>	<p>Commonly quizzes use centre alignment for answer options so this familiarises the user with the window and also shows that a list would indicate that they are answer options. Small padding to give space between answers so they aren't compact but are obviously all answer buttons, they're also not too large that they are separated into different parts of the page which could confuse the user.</p>

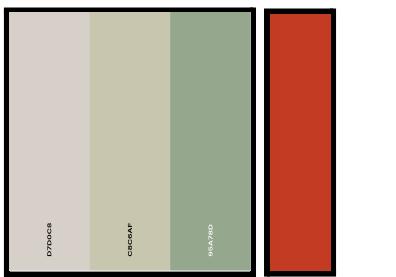
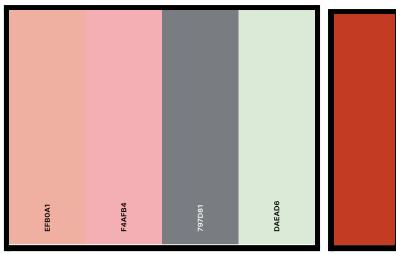
Window 3 (Dark and light theme):

Options: 1, 2 and 3	What conventions are followed	Why I selected them (end users)
---------------------	-------------------------------	---------------------------------

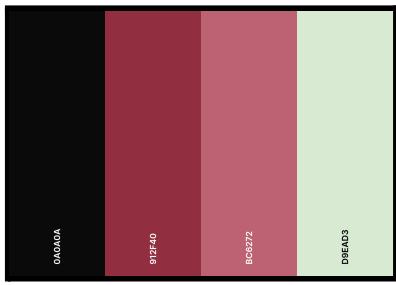
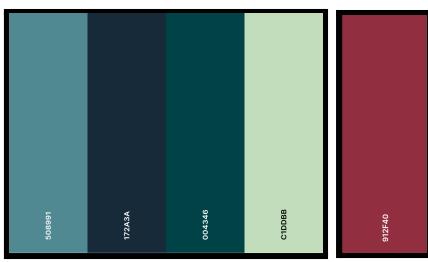
	<p>Logo in top left. Scores centre aligned and listed. Small padding between scores. Exit button bottom left. Contrast.</p>	<p>Logo is in top left to keep it out of the way of the more important content on the page but also to remind the user that they are interacting with the Mental Health quiz if they ever forget. Results are listed in a centre alignment and have some padding between each other so the scores are clearly displayed to the user, it also is common to understand that in a list of results the score will go from highest to lowest which would be easy for the user to understand. The text is contrasted (for the light theme there is dark text and light background or vice versa for the dark theme) from the background so that all content is clearly displayed to the user.</p>
	<p>Scores are centre and right aligned, more padding.</p>	<p>This spreads out the score as there is different alignment and more padding which will fill my page more so the user doesn't feel like they are staring at an empty page.</p>
	<p>Bordered scoreboard.</p>	<p>The border indicates there are numeric values and also makes sure that the user realises it's a scoreboard and doesn't feel like the data is scattered along the window.</p>

Colour schemes (Light theme):

	<p>For my light theme colour palettes I decided to select colours that were bright and created a warm/happy emotion for the user. I made sure that I kept the colour green throughout each pallet as it can be used for indicating that the user has done something correct. The colour red is also present in all colour pallets as it indicates the user has done something incorrect or an error has occurred.</p>
--	---



Colour schemes (Dark theme):



For my dark theme colour palettes I decided to select colours that were strengthening and created a clean and professional appearance. I made sure that I kept the colour green throughout each palette as it can be used for indicating that the user has done something correct. The colour red is also present in all colour palettes as it indicates the user has done something incorrect or an error has occurred.

3.3 Feedback and improvements on Design ideas

Window 1

1

MENTAL HEALTH QUIZ

Hi there! Welcome to the mental health quiz.

Please enter your name below and press start to begin!

Enter Name:

START

Dark theme

Light theme

2

MENTAL HEALTH QUIZ

Hi there! Welcome to the mental health quiz.

Please enter your name below and press start to begin!

Enter Name:

START

Dark theme

Light theme

3

MENTAL HEALTH QUIZ

Hi there! Welcome to the mental health quiz.

Please enter your name below and press start to begin!

Enter Name:

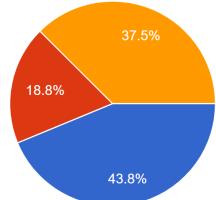
START

Dark theme

Light theme

Results (Google Form)

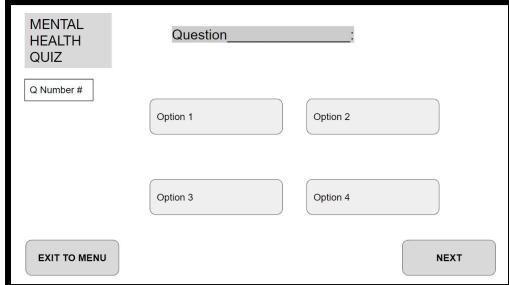
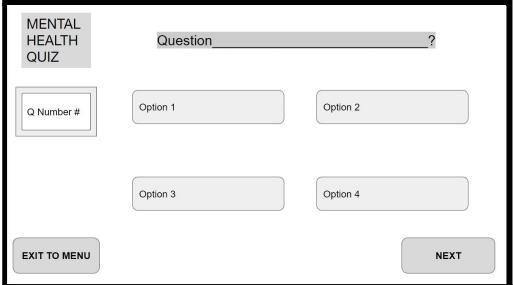
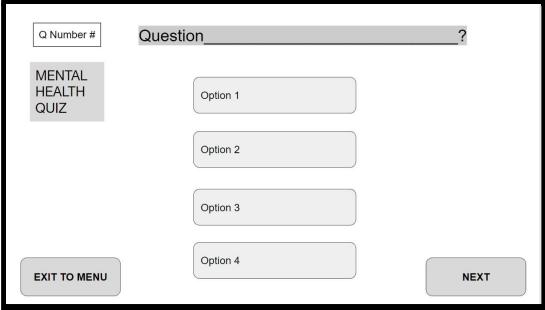
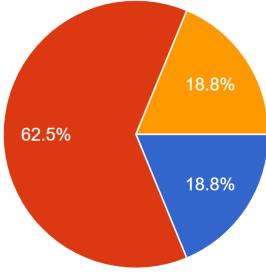
Which layout do you like the most?



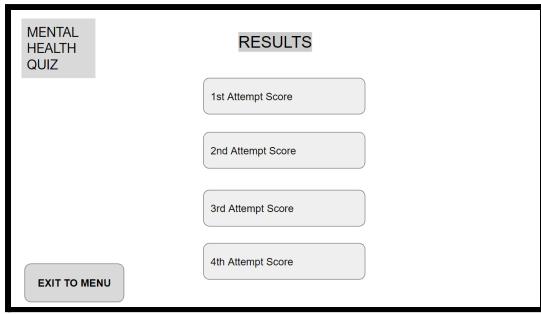
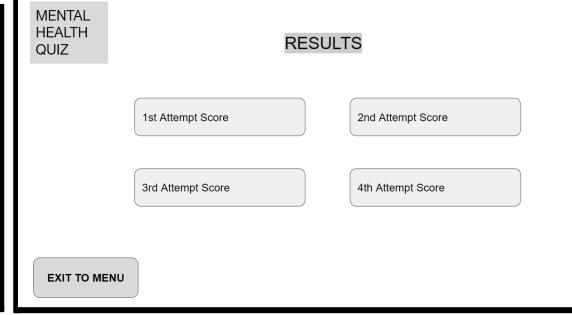
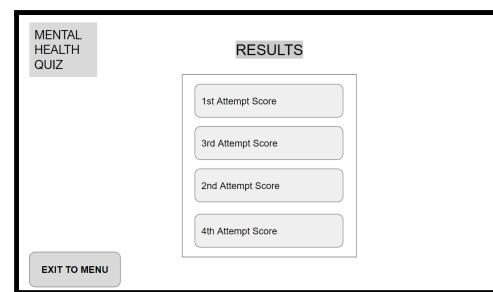
Overall

Majority (43.8%) of the students voted for option 1 to be my layout for the first window. This is mainly because the layout is very familiar to users as most programmes use this format, as well as being very simple for new users to navigate around the programme.

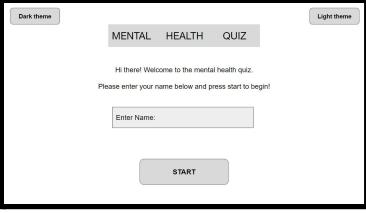
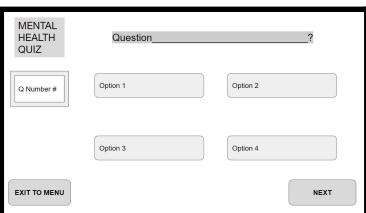
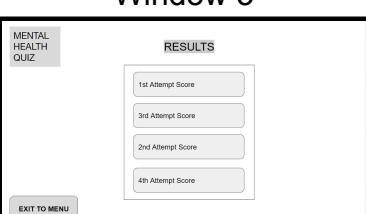
Window 2

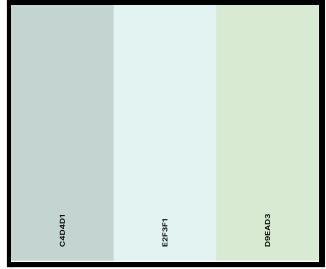
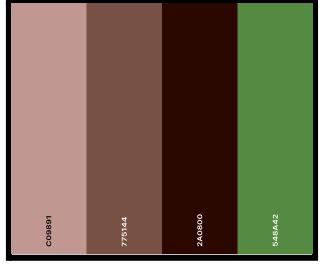
<p>1</p> 	<p>2</p> 								
<p>3</p> 									
<p>Results (Google Form)</p> <p>Which layout do you prefer? (16 responses)</p>  <table border="1"> <thead> <tr> <th>Layout</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Option 1</td> <td>62.5%</td> </tr> <tr> <td>Option 2</td> <td>18.8%</td> </tr> <tr> <td>Option 3</td> <td>18.8%</td> </tr> </tbody> </table>	Layout	Percentage	Option 1	62.5%	Option 2	18.8%	Option 3	18.8%	<p>Overall</p> <p>Majority (62.5%) of the users had chosen option 2 as the best layout for my second window. This was because they liked the consistency of the logo being in the top right, the sizes of the answers are large enough to easily interact with and the labels are placed in suitable places.</p>
Layout	Percentage								
Option 1	62.5%								
Option 2	18.8%								
Option 3	18.8%								

<p>Window 3</p>
<p>Option 1</p>

		
<p>End users' Opinions (Personal Response)</p> <p>I prefer option 3 because it is consistent with the rest of the design.</p> <p>I chose option 3 because it has a table which is easier to follow rather than different text bubbles one after another. This also saves space to read multiple results without having to scroll as much.</p> <p>I like option 3 because of the fact that the layout is centered compared to others and the border is cool.</p> <p>In my genuine opinion I like option 3 because the layout of the scores is similar to those seen on leaderboards in different interfaces keeping with consistency.</p>		<p>Overall</p> <p>Overall, all users have selected option 3 for my window 3 layout with the results table. The users liked how it was consistent with common layouts from other websites for results tables and therefore is more familiar for the user to interact with. It is also a cool design yet simple which makes the user more attracted to the window.</p>

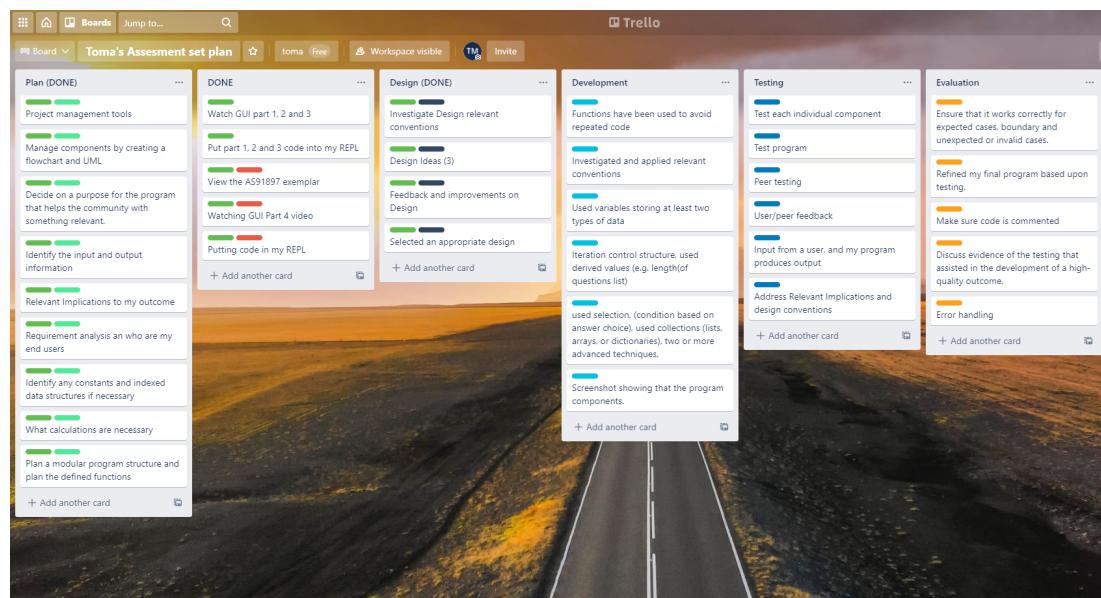
3.4 Selected Design and its appropriateness

Selected design	How it meets the purpose	How it meets the requirements	End-user opinions
<p>Window 1</p> 	<p>The purpose of the menu/starting page is to introduce the user to the mental health quiz clearly and simply.</p>	<p>My selected design for the menu page has a simple yet attractive layout which will entice the user to use my programme and also feel more control (relative to the heuristic user control and freedom) over the quiz with the different buttons therefore will have a better experience.</p>	<p>After getting results from my google forms and in person feedback, this design has been the majority vote as the labels and widgets are placed fairly and it's easy to navigate around the program.</p>
<p>Window 2</p> 	<p>The point of having my question page is to display my quiz's question and answer options in a way that the user understands how to use and enjoys their experience.</p>	<p>My second window's layout is familiar for the user as it is similar to other quizzes I have seen before, this makes it an easier/efficient experience with my quiz for the user as it is understandable. As well it is simply pleasing to the eye rather than cluttered widgets which can stress the user out while using my quiz.</p>	<p>From my form results I have come to the conclusion that this design is the best according to user opinions as the page is similar to other quizzes making it easier to use.</p>
<p>Window 3</p> 	<p>The results page has the purpose of showing the user their previous results and also in order from highest score to lowest in a simple way.</p>	<p>My page is clearly laid out without looking compact which gives the user a specific place to look at (the centre) and doesn't bring confusion as the table is very simple to understand.</p>	<p>From the form results I received it is clear that my end users recommend I use option 3 as my selected design from the leaderboard page. According to user opinions, the format of the leaderboard page is much more familiar for the user as it is like other websites and it is also very simple to register what's on the page.</p>

	<p>The light theme colour scheme has the purpose of having contrasting colours (where light colours are most dominant) so that the content stands out to the user and is easy to read.</p>	<p>The selected light colour scheme has a fair amount of contrast which allows the user to understand what is on each window without feeling overwhelmed if I were to use many completely different colours. The specific colours also have different meanings as researched before which will help the user have more accessibility while using my programme.</p>	<p>From asking around my class which scheme was most aesthetically pleasing and suited my purpose, the majority said this colour scheme was the best as it is bright and the colours work well together so I decided to use this as my light theme.</p>
	<p>The dark theme colour scheme has the purpose of having contrasting colours (where dark colours are most dominant) so that the content is bold and easy for the user to understand.</p>	<p>The selected dark colour scheme has some contrasting which helps the user to understand what the content is on each window without feeling overwhelmed if I were to use many completely different colours. The specific colours I chose also have different meanings as researched before which will help the user have more accessibility while using my programme.</p>	<p>For this dark theme colour scheme, the majority of my classmates suggested that this theme looked the best and the colours fitted most well together so I decided to select this scheme as my dark theme.</p>

Trello board progress:

- I have finished the checklist for my design section meaning I can move onto my development section.



4. Develop and Test Component

Repl.it link:

<https://replit.com/join/ykxaritp-18357toma>

Version Control - Github Link:

https://github.com/TomaMirchevska/Toma_Quiz

Component 1: MenuPage and DarkMenuPage

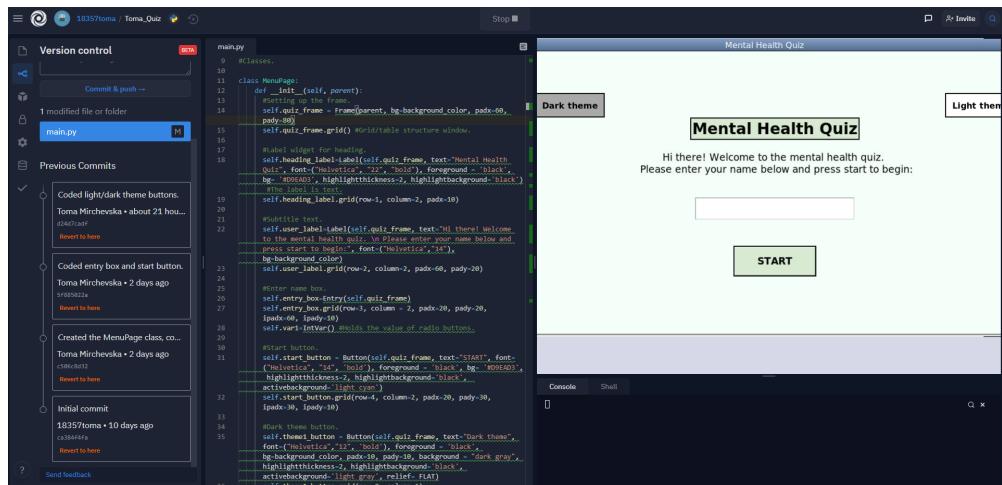
Trialling:

What techniques or components did I trial?
Which did I choose and why?

Trial 1:

MenuPage:

For this trial I tried my first component that is a class called MenuPage. I tried to make the basic look of the window similar to how my wireframe was laid out.



Techniques I have used:

- A grid layout for my widgets for example. `self.quiz_frame.grid()`. I did this because I wanted to use columns and rows so that I was able to align my widgets in a way that follows the design conventions. It also allows me to create more columns and rows which makes it easier to place different content around the page.

- There are hover effects for my buttons. I have implemented this so that the user knows the widget is a button and also

gives them satisfaction while interacting with it. I did this by coding with an activebackground so that I can choose what colour the button changes to when the user hovers over it.

- Created radio buttons

In my code I have created three buttons: continue, dark theme and light theme. These are all radio buttons that the user can interact with. They don't perform any specific actions yet except for hover effects, I intend to command the buttons to redirect the user to other windows.

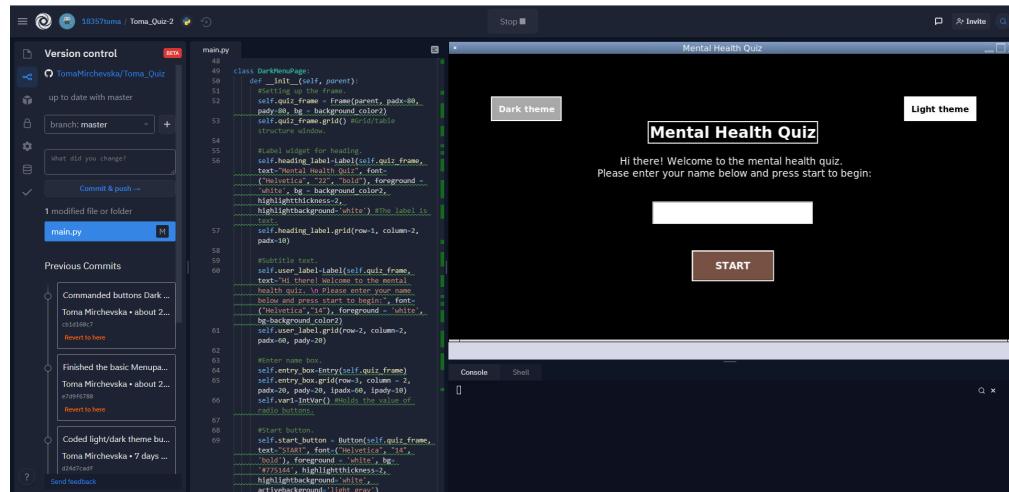
Differences to the expected window:

- The window extends too wide and so the content spreads as well which doesn't follow my design convention of having a reasonably large window.
- All of the text I have decided to remain a black font colour for now so that it stands out more and is easier for the user to read and will configure this in another version.
- Buttons are not round which can be uncomfortable for the user as buttons tend to be rounded so it's easier on the eyes.

Trial 2:

DarkMenuPage:

For my second trial I completed the component class DarkMenuPage. This is a duplicate component of the MenuPage with minor adjustments relative to colour.



Techniques I have used for this window:

- Same techniques as the MenuPage such as activebackground for hover effects, radio buttons and a grid layout.

I have kept the same techniques with a different colour scheme so that the windows remain the same layout but also work according to the two light and dark colour schemes I've selected.

- Radio buttons have commands.

My light theme button has a command to destroy the dark theme window when the user clicks it and opens the light theme window (MenuPage). My dark theme button has the same command except to destroy the light theme window and open the dark theme window (DarkMenuPage) when the button is clicked. The buttons work for both classes.

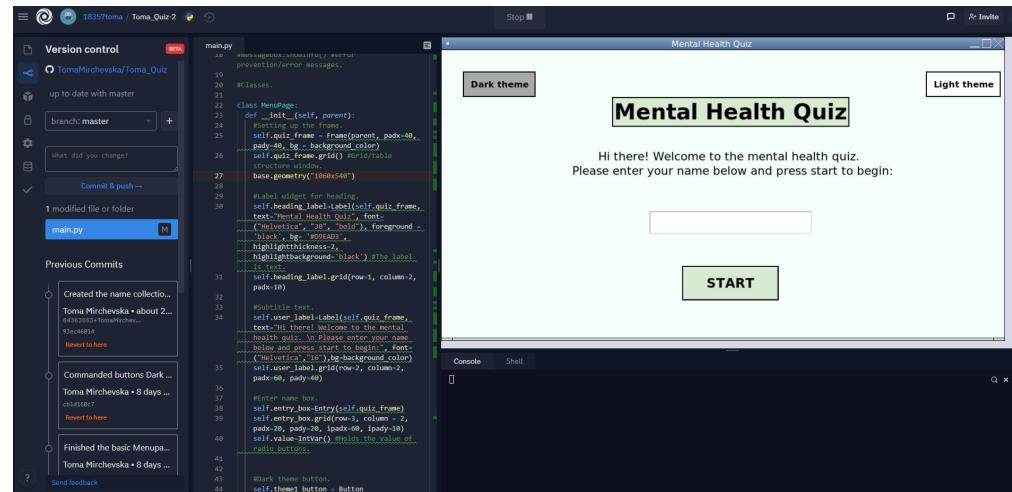
Differences to expected window:

- I have kept most of the font colours white so that it is simple and easier for the user to read instead of multiple colours on the text.

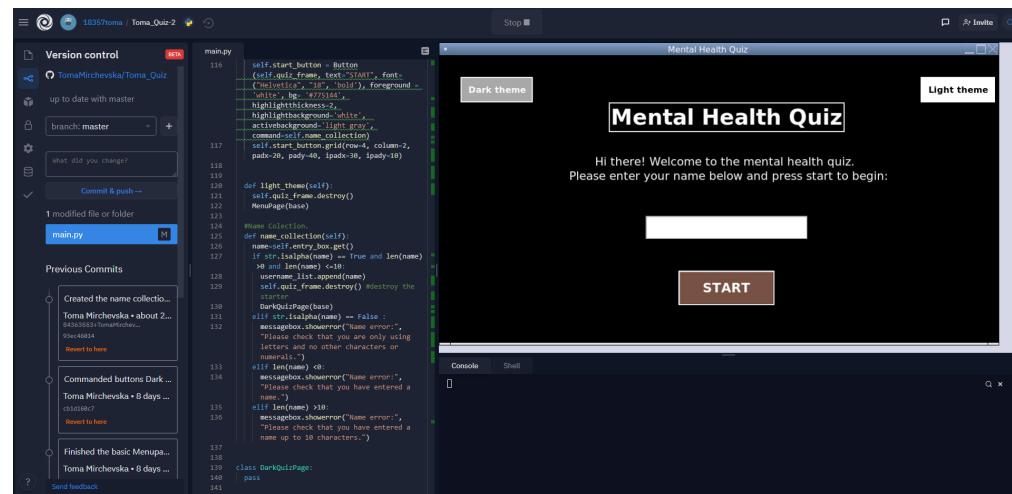
Trial 3:

For the final trial of my first component I have added a name collection method into the classes MenuPage and DarkMenuPage. I have also enlarged some of the labels and buttons so that it is more visible to the user.

MenuPage:



DarkMenuPage:



Techniques I have used:

- `str.isalpha(name) == True`

This alpha technique with str. allows the name_collection method to only allow string variables (letters) to be stored as the name variable. This means if the user enters numerals or other characters it will not go through.

- `len(name) >0 and len(name) <=10`

Len means the length of the name that is entered into the entry box. I have made the range from more than 0 characters and under 10 characters allowed to be stored as the name variable. If the user enters a name outside the boundaries then it will not be stored.

- Error message box

I have used the messagebox.showerror technique which is imported from tkinter so that an error message will be displayed saying what error they have made when the user doesn't meet the requirements of the name_collection method. (This follows the usability heuristic Diagnose and Recover from Errors, the message will help direct the user to efficiently recover from the error made).

- When the user enters the name correctly, the window will be destroyed and the user will be transferred to the next quiz window.

User Testing and Feedback:

MenuPage

Likes:

Tina - The format is very minimal which I like as it is easy to understand and navigate around the programme which makes it accessible to me and other people as it is simple to understand for any user.

Ansh - I like that the colours are contrasting and the colour scheme works together.

Nico - Every text is easy to read despite the different colours.

Improvements:

Tina - Although it is functional it isn't that aesthetically pleasing. You could make it more appealing by changing the background to an image or Making the text less simple.

Ansh - I don't like how the window size isn't consistent with the DarkMenuPage so I suggest that you make a fixed window size for all your classes.

DarkMenuPage:

Likes:

Tina - The dark colour scheme makes the texts pop out which is attractive and easier to read.

Ansh - Having outline for the title and buttons brings a nice minimal design to the page which makes me feel interested in doing the code.

Improvements:

Nico - Like the MenuPage, the design is a bit too simple for me and I suggest that you add a bit more complexity to the design of the title and background.

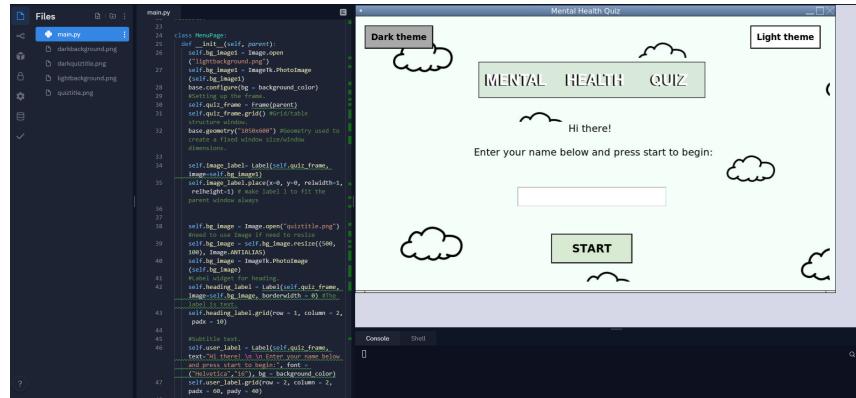
Tina - The buttons don't stand out much which I would be more satisfied with using a button that sticks out of the page like other programmes do.

Final outcome:

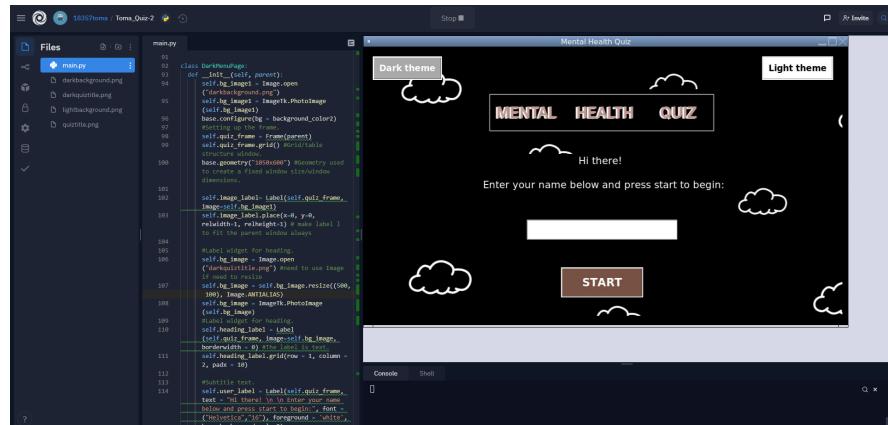
(Following the user feedback)

For the last trial of my MenuPage component I have tried my best to follow the feedback of my users.

MenuPage:



DarkMenuPage:

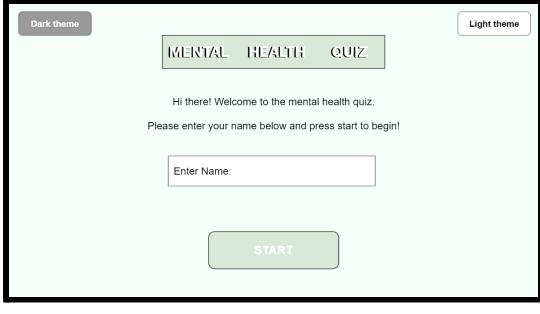
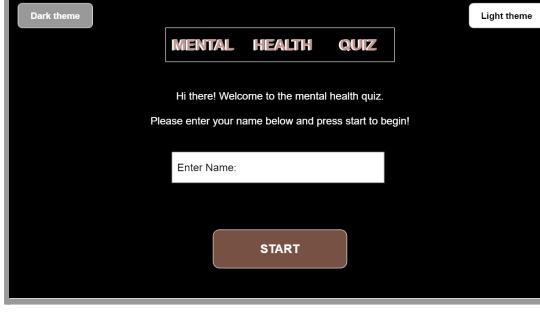


Techniques I have used:

- Label Image.

I have used a label image to display my original wireframe title into the actual programme. I did this because the previous was too minimalistic and didn't entice the user into using my programme.

- Relief = RAISED.

	<p>This technique was used for my buttons so that they stand out of the screen more which is relevant to my feedback. I did this because it is more satisfying to the user when interacting with the button and allows the user to unconsciously know they are buttons as it is familiar for other programmes to make them stand out like this.</p> <ul style="list-style-type: none"> - Background image. <p>By using similar coding to the label image, I have created a minimal yet more complex design for a background image which is more aesthetically pleasing to the user according to my feedback. This means the user will have a more positive experience with my quiz without feeling it is too dull or boring to interact with.</p>
<p>Testing</p> <p>Test case</p>	<p>Expected</p> <p><u>MenuPage:</u></p>  <p>Actual (screen shot)</p> <p><u>MenuPage:</u></p>  <p><u>DarkMenuPage:</u></p>  <p><u>DarkMenuPage :</u></p> 
<p>How did I fix it if it failed the test?</p>	<p>The error message pops up when the user performs the wrong action on my programme such as pressing start without entering their name. When the user doesn't enter their name, an error message pops up stating that they should check they are only using letters whereas the real message should be to check if you have entered a name.</p>



I fixed this mistake by changing my code to say that when the characters are >1 then the error message for 'Please check you have entered a name" pops up instead of when characters are >0 . I realise that this mistake occurred because it isn't possible to have less than 0 characters entered in the box.



How I addressed Relevant Implications:

Design:

In association to the relevant implication Intellectual Property, I have made sure to avoid a problem with any creator by creating my own images. In my new background design I have created the image myself while using cloud images that were labeled as 'Usage right of Creative Common License' meaning I could reuse it.

Relative to the usability implication, I have created a method that will provide an error message relevant to the error made by the user when using the entry box. This makes sure the user can perform tasks correctly and relevant to the usability heuristic, they can diagnose and recover from errors. This allows them to efficiently use my programme and hopefully remember not to make the same error

Outcome:

I ended up using the designed background image which remains the same in my programme. This avoided any legal problems with creators seeing me use their content without permission.

To avoid problems with usability I followed my design and made sure my error messages were accurate to the error made by the user and are helpful enough to let the user recover from the mistake and efficiently use my programme.

In my real outcome I followed very closely to the design I had wireframes which helped me to address the accessibility implication as all my widgets were set out clearly for the user and large enough for them to read. This meant there was less of a chance that my end users would have a bad experience with

	<p>next time using my quiz.</p> <p>I addressed the accessibility function in my wireframe design so the user can navigate around my programme easily and so the layout is simple for any user to understand. I made sure my widgets are large enough for the end users to read and understand. The buttons are also spaced out enough so that the user can differentiate everything and with alignment, the contents that are aligned together are related to each other and easier to understand by the user.</p> <p>I addressed the relevant implication accessibility in my design by using the font Helvetica as it is the easiest font to read and by making sure my widgets appeared large enough for the users to see and understand. This implication was important to me as I wanted to make sure my end users have a good experience with my programme without struggle, so I intended to make clear content in my design layout.</p> <p>The very important implication functionality wasn't too relevant to my design but more so relevant to the actual programme itself. In my design however I planned out where my buttons and widgets were and with the texts/labels I intend to make those widgets function as they are said to do e.g. start button when pressed takes the user to the quiz page to start.</p>	<p>my programme and even become stressed while doing the quiz.</p> <p>Functionality in my programme is very evident as all my widgets perform the action they are meant to when pressed by the user. This addresses the functionality implication and avoids the problem of the user not getting what they want when interacting with my code. As all my widgets perform the correct action that they imply to the user, there wouldn't be any functionality problems meaning my programme works well for my end users. Helvetica was also used to avoid problems for dyslexic users as it is a simple font to read so that they don't mix up the letters and therefore feel less accessible to the programme.</p>
How I applied design conventions	<p>In my design I had applied design conventions such as alignment, fonts, colours, and font sizes to meet the user requirements and produce a good programme. I put the main widgets of my programme in the middle of the page, centre aligned with the purpose that it attracts the user straight away and shows the user that the aligned content is related to each other. My theme buttons are aligned to the left and right so that the user doesn't assume it is the most important and think they have to press those buttons. The buttons and font sizes are large enough to be readable for users even with poor eyesight which makes my programme more accessible to my end users. Helvetica was the main font I used throughout my window as I researched it to be the most simple font for users to understand so that my programme is accessible to users who find reading hard or even have dyslexia. The colours I have used in my design are contrasting for light and dark themes meaning my title and widgets stand out from the background and allows the users to see it easier while being attracted to the design of the page as well.</p>	

	All my design ideas were applied to my programme and real outcome.
Version Control	<p>Github Link: https://github.com/TomaMirchevska/Toma_Quiz</p>

Component 2: Quiz page - QuizPage and DarkQuizPage

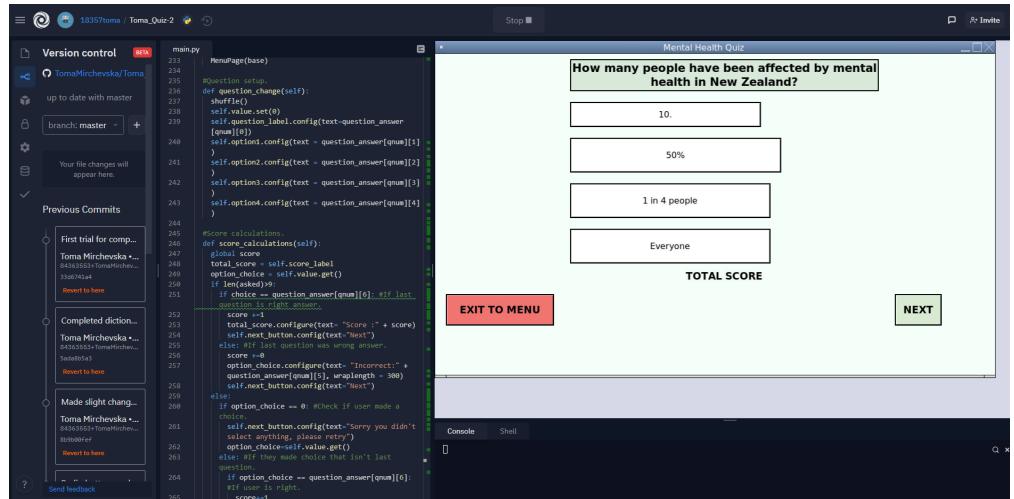
Trialling:

What techniques or components did I trial? Which did I choose and why?

Trial 1:

For the first trial of my second component I have created the base of my quiz page meaning I have created the main methods of the quiz such as question_change (randomises the question and answers after user presses next) and score_calculations (calculates the score after the user gets an answer correct or incorrect).

QuizPage:



DarkQuizPage:

The image shows a GitHub repository interface for a project named 'Toma_Quiz-2'. The main.py file is open in the code editor, showing Python code for a mental health quiz. The code includes functions for question calculations and score tracking. The application window titled 'Mental Health Quiz' is running in the background, displaying a question: 'What is the best thing to do when you see someone enjoying their day?'. Four options are listed: 'Nothing, or vibe with them.', 'Ruin their day.', 'Punch them.', and 'Make jokes and laugh at them with friends.'.

```

version control
main.py
324 self.option2.config(text = question_answer[qnum][2])
325 self.option3.config(text = question_answer[qnum][3])
326 self.option4.config(text = question_answer[qnum][4])
327
328 #score_calculations
329 def score_calculations(self):
330     global score
331     self.score_label.config(text = str(score))
332     option_choice = self.value.get()
333     if len(qnum) > 0:
334         if choice == question_answer[qnum][6]: #If last question is right
335             score += 1
336             total_score.config(text = "Score : " + str(score))
337             else: #If last question was wrong answer.
338                 score -= 0
339                 option_choice.config(text = "Incorrect: " + question_answer[qnum][6])
340
341             if option_choice == 0: #Check if user made a choice.
342                 self.score_label.config(text = "Sorry you didn't select anything, please retry")
343             else: #If they made choice that isn't last question.
344                 if choice == question_answer[qnum][6]: #If user is right.
345                     score += 1
346                     total_score.config(text = str(score))
347                     self.question_change() #Method for next question to come up.
348                     else: #If user chooses wrong answer.
349                         score -= 0
350                         total_score.config(text = "Incorrect: " + question_answer[qnum][6])
351                         foreground = "white"
352                         if question_change():
353
354     if __name__ == "__main__": #If this is the file name then it will be able
355         to run as a stand alone program.
356         base = Tk() #Creates window.
357         base.title("Mental Health Quiz") #Window title.
358         quiz = Quiz(base) #Creates an instance of the class
359         quiz.mainloop() #Shows window until user closes it.
360
361
362
363
364
365
366
367
368
369
370
371

```

Techniques used:

- Grid.

This is a technique I used to place my content in rows and columns in my window. It allows me to align the content in the middle that is most important by putting them in the same column and separating the other widgets in different columns.

- Radio buttons.

I have used radio buttons as a technique to create my question answers so that they can have value in the programme and when the user clicks on an answer and presses next, the code determines whether the answer is right or wrong depending on the dictionary question_answer.

- If/else statements.

This important technique allows the user to have different selections and get different outcomes. I used the statements when creating the score_calculations functions so that the programme can calculate the score when the user selects a right answer, wrong answer or no answer and therefore labels will have different configurations depending on the result.

- Configure and iteration.

Since I have created a dictionary question_answer, I can easily use the configure technique which will allow me to change the question label and the radio buttons to fit all the questions in the dictionary. This technique makes my programme and coding much more efficient as I don't have to repeatedly create new windows, it also makes my programme robust and allows others to copy the code and only have to make minor changes.

I have created a method called question_change which shuffles the questions around which makes it an iteration as it continues to loop until all questions_answers have been displayed from the dictionary and then stops.

Differences to the planned outcome:

- I have decided to display the total score for the user so that

- they can recognise how well they are doing throughout the quiz until the end.
- There are no images (image labels) for background and logo which I intend to work on in the next trial.
 - The radio buttons are placed in the same column at the moment and I hope to change that in my next trial so that the format of buttons is how I planned.
 - When a question with long answers is configured into the label and radio buttons, the rows are shifted lower so that the buttons don't appear on the page anymore which I can fix by trying to use place techniques instead of grid.
 - When I display my short text describing the correct answer, it isn't shown clearly for the user which means I should find a better place/way to display the text that doesn't compact the content.
 - I do not have a question counter yet which I can code in my next trial by using configure e.g. configure a question number label to add 1 each time the user clicks next.

Trial 2:

For this trial of my second component QuizPage and DarkQuizPage I have put my focus on making the window look very appealing to the user and that the content is clear/easy to read.

Techniques I have used:

- Place instead of grid.

In my previous trial the grid format didn't work out for me if I were to follow my wireframe format, therefore, I decided to switch to the place technique which allows me to select specific x and y coordinates so that I can place my labels and buttons exactly where I want them. The main reason I used this was so I could place the radio buttons in my originally planned alignments.

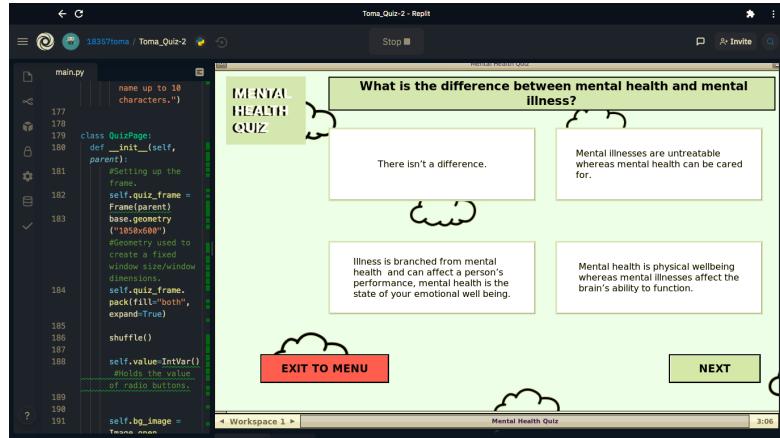
- Wraplength

I have used this attribute for my texts such as the radio buttons and question label so that I can wrap (keep) the text within a certain width value so that it doesn't run off the area I want it to remain inside.

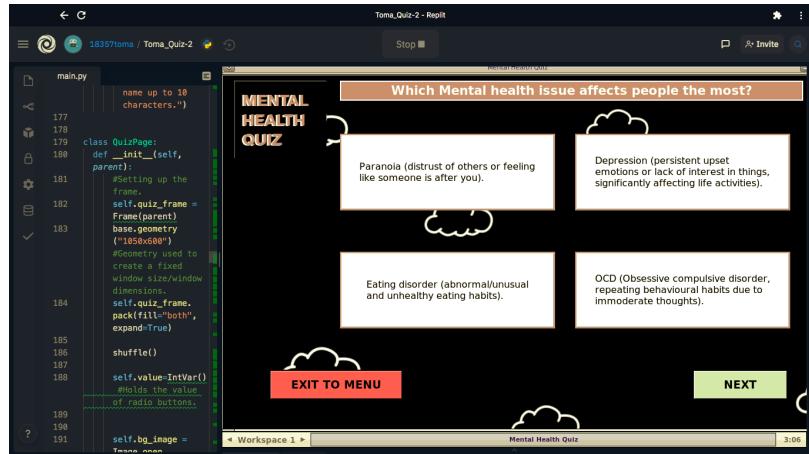
- Selectbackground

In my opinion this was a very important technique that I implemented in my code as it relates to the usability heuristic Visibility of System Status. I used this technique for my radio buttons so the user understands that if they click on the button and it remains a darker colour, it means they have selected that answer. This provides them with reassurance that they have selected the button and also satisfaction as the programme is giving something back to the user when they perform an action.

QuizPage:



DarkQuizPage:



Differences to expected outcome:

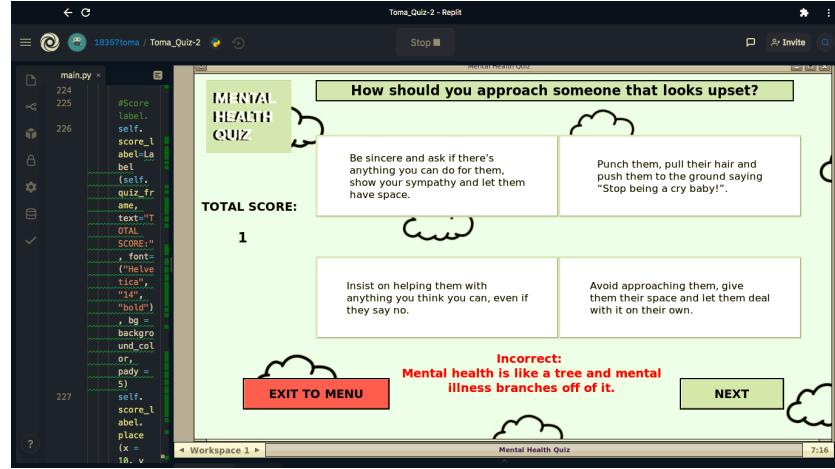
- I haven't implemented the question number counter which was planned in my wireframing. If my users say they recommend it in the end user testing then I will implement it according to their requirements.
- Radio buttons are outlined with the dark and light theme colour so that the buttons stand out more to the user and match the colour schemes.
- The score isn't displayed yet which I intend to work on in the next trial as well as displaying the descriptive text for the answers and if the user is incorrect or correct

Trial 3:

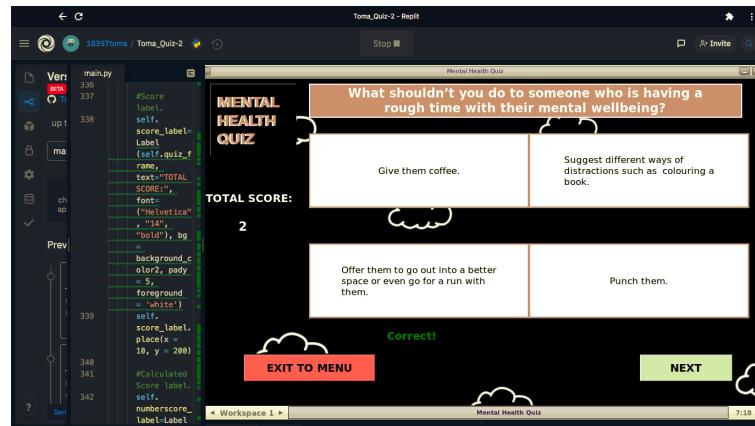
For my 3rd trial I have decided to try following the layout of my wireframe and adding different functions such as the display of score

and displaying a descriptive text for each question when the user gets an answer correct or incorrect.

QuizPage:



DarkQuizPage:



Techniques I have used:

- Configure for score label and answer text label.

In my `score_calculations` method I have added different configurations for my score label so that the users total score is displayed and they can keep track of it. Most importantly I have added configurations for the answer text label which means when the user gets an answer correct, the text will clearly tell them they are correct in green font as it is a commonly known colour for representing achievements. If the user is wrong a text saying incorrect will be in red along with more text to help tell the user why the answer isn't right which slightly relates to Diagnose and Recover from Errors if they want to play the quiz again.

User Testing and Feedback:

MenuPage:

Likes:

Ansh: The colour scheme works well to create a happy interaction with the quiz which relates to my mental health.

Adil: The layout of all content is very easy to register as a user and I like the simple design that brings light to the quiz.

Tina: Having the images and colours work together brings a very positive experience for me as a user and makes the programme aesthetically pleasing for both dark and light themes.

Nico: The buttons are a nice large size and stand out to me which unconsciously lets me know that those are the answer buttons for the questions.

Improvements:

Adil: Although the format of the buttons and labels are very simple and clear to the user, I suggest you indicate slightly to the user what the labels mean e.g. question label should say "Question:" and then the text for more clarity.

Nico: The buttons should have a way to tell the user which option is 1, 2, 3 and 4 because it can be confusing as the buttons aren't in a list alignment.

Tina: The question label and buttons should maybe be placed a little lower because it makes me feel like the content is all packed together at the top of the window. I recommend you space it out a bit more so that it doesn't feel so compact.

Ansh: I think the idea of having a question counter is really effective as it lets the user know what question they are on, I would like to see that you implement this in your programme as you planned in your wireframe.

DarkMenuPage:

Likes:

Ansh: The dark theme colours really allow the content to stand out as there is contrast and it is very appealing to me. All the buttons work as it should which makes the programme functional.

Adil: I feel relaxed in the sense that it feels like I am in a dream. The dark colours really are relatable and personally they are my comfort colours.

Nico: I like the decision of colours. The brown isn't too bold of a colour yet still contrasts very well with the black and makes the text stand out. It makes the programme accessible to users even with colour blindness.

Improvements:

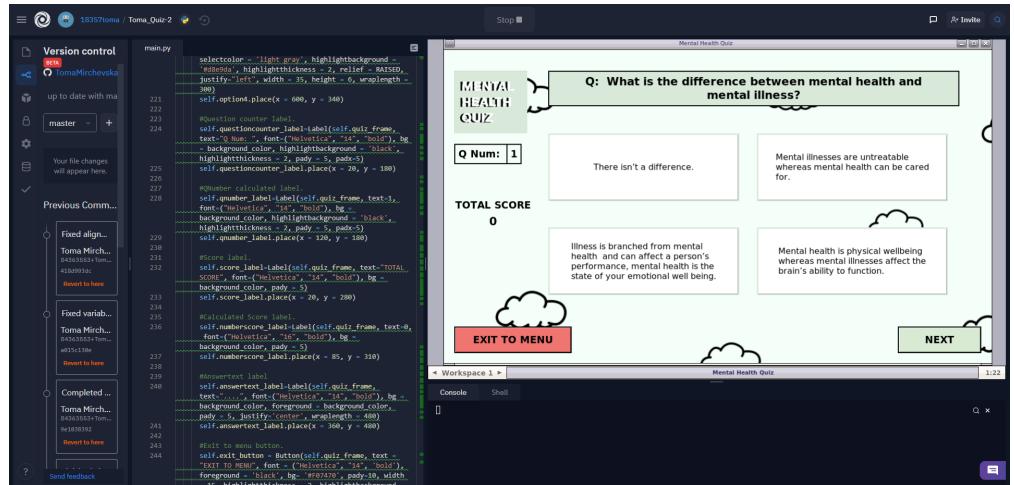
Nico: The widgets are a bit too close together. In my opinion you should spread the content out a bit more from each other so that it is easier on my eyes.

Tina: For both the themes I don't like how the logo image isn't aligned with the total score, it would be more comforting to see the content in line with each other.

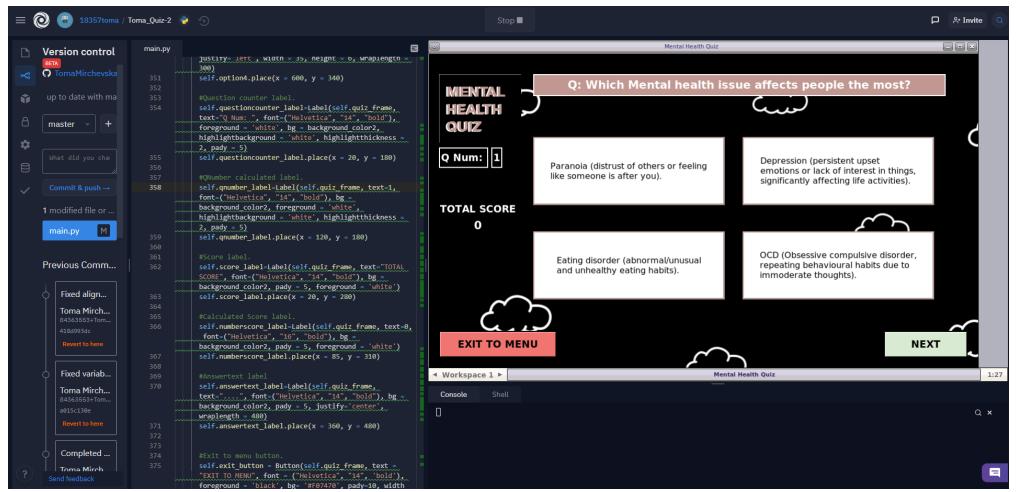
Final Outcome: (Following the user feedback)

In this final trial I have made sure to attend to the different requirements my classmates had as they fit into the selection of my end users. This means their input is really relevant to my programme as I want it to be suitable to my end users. The main thing I sorted out in this trial was aligning my widgets in a format that makes the user feel comfortable, similar to my wireframe layout.

QuizPage:



DarkQuizPage:



```

# Question counter label
self.questioncounter_label = Label(self.quiz_frame,
text="Q Num: 1", font=("Helvetica", "14", "bold"), bg="white",
foreground="white", highlightbackground="white", highlightthickness=2, padx=5)
self.questioncounter_label.place(x=20, y=180)

#Score label
self.score_label = Label(self.quiz_frame, text="TOTAL SCORE", font=("Helvetica", "14", "bold"), bg="white",
foreground="white", highlightbackground="white", highlightthickness=2, padx=5)
self.score_label.place(x=120, y=200)

#Score counter label
self.numberscore_label = Label(self.quiz_frame, text="0", font=("Helvetica", "10", "bold"), bg="white",
foreground="white", highlightbackground="white", highlightthickness=2, padx=5)
self.numberscore_label.place(x=80, y=210)

#Answered label
self.answered_label = Label(self.quiz_frame, text="ANSWERED", font="Helvetica", bg="white",
foreground="white", justify="center", wraplength=400)
self.answered_label.place(x=360, y=480)

#Exit button
self.exit_button = Button(self.quiz_frame, text="EXIT TO MENU", font="Helvetica", "14", "bold"),
foreground="black", bg="#FF4500", padx=10, width=15)
self.exit_button.place(x=100, y=400)

```

Techniques I have used:

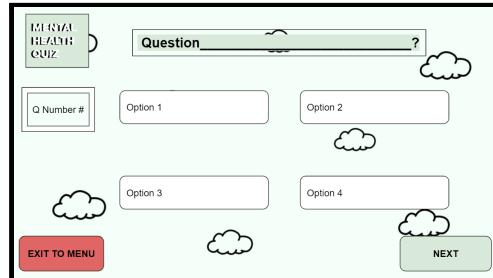
- Configure for question number label.

Very similar to my total score display, I have used configure to change my question number so that the user knows what question they are on. This idea was part of my wireframe and I think it was really good to implement into my programme as it follows the usability heuristic Recognition rather than recall meaning the user doesn't have to memorise which quiz question they are up to (as the programme will calculate it for them).

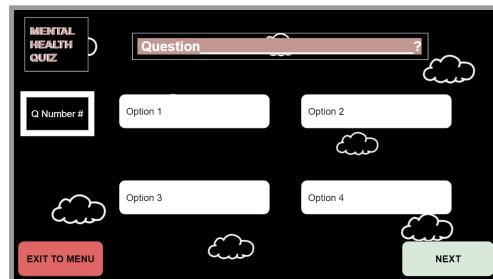
Testing Test case

Expected

QuizPage:



DarkQuizPage:

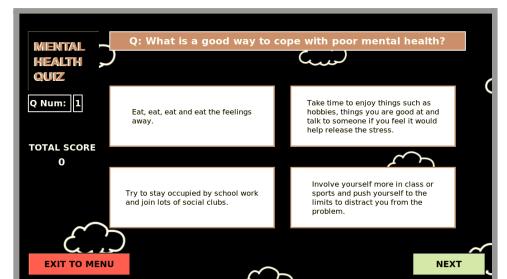


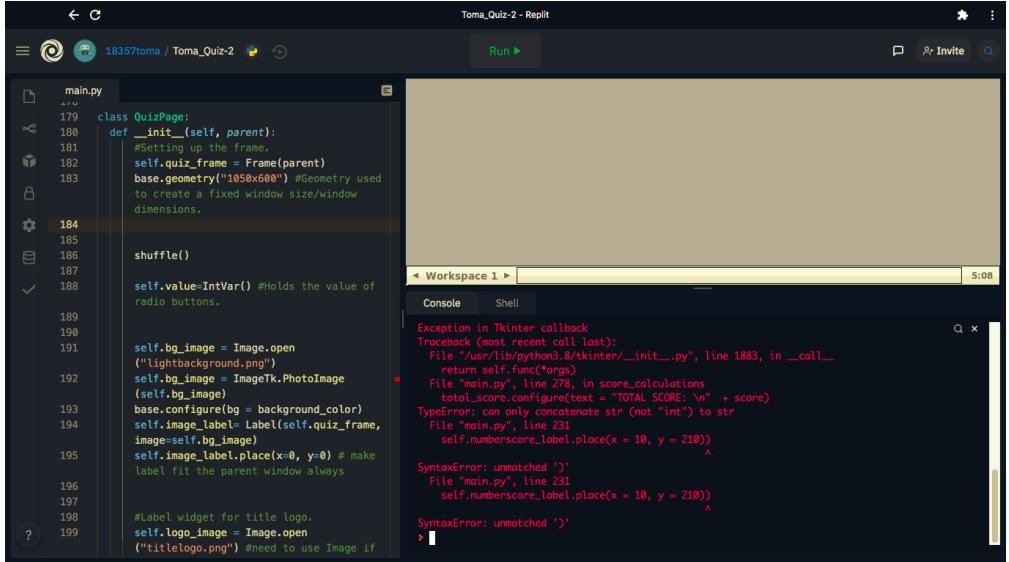
Actual (screen shot)

QuizPage:



DarkQuizPage:



<p>How did I fix it if it failed the test?</p>	<p>Placement error:</p>  <pre> main.py 179 class QuizPage: 180 def __init__(self, parent): 181 #Setting up the frame. 182 self.quiz_frame = Frame(parent) 183 base.geometry("1050x600") #geometry used 184 to create a fixed window size/window 185 dimensions. 186 187 shuffle() 188 189 self.value=IntVar() #Holds the value of 190 radio buttons. 191 192 self.bg_image = Image.open 193 ("lightbackground.png") 194 self.bg_image = ImageTk.PhotoImage 195 (self.bg_image) 196 base.configure(bg = background_color) 197 self.image_label= Label(self.quiz_frame, 198 image=self.bg_image) 199 self.image_label.place(x=0, y=0) # make #Label widget for title logo. self.logo_image = Image.open ("titlelogo.png") #need to use Image if </pre> <p>Exception in Tkinter callback Traceback (most recent call last): File "/usr/lib/python3.8/tkinter/_init_.py", line 1883, in __call__ return self.func(*args) File "main.py", line 278, in score_calculations total_score.configure(text = "TOTAL SCORE: \n" + score) TypeError: can only concatenate str (not "int") to str File "main.py", line 231 self.numberscore_label.place(x = 10, y = 210) SyntaxError: unmatched ')' File "main.py", line 231 self.numberscore_label.place(x = 10, y = 210)) SyntaxError: unmatched ')'</p>	<p>This error occurred during one of my trials due to not using correct syntax before using the place technique throughout my class. I have managed to fix this error by researching how to properly use the placement technique rather than grid and I found out that in my code there wasn't a point where I 'called' the place by saying <code>self.quiz_frame.pack(fill="both", expand=True)</code>. I have added that into my code and the frame now shows all the widgets using the place technique.</p> <p>Configuring label with string text and a variable:</p>

```

18357oma / Toma_Quiz-2
Stop ■
main.py
262
263     #Score calculations.
264
265     def score_calculations(self):
266         global score
267         total_score = self.numberscore_label
268         option_choice = self.value.get()
269
270         if len(asked)>9:
271             if choice == question_answer[qnum][6]: #If last question is
272                 right answer.
273                 score +=1
274                 total_score.configure(text= score)
275             else: #If last question was wrong answer.
276                 score +=0
277                 self.answerstext_label.configure(text= "Incorrect: \n" +
278                 question_answer[qnum][5], wraplength = 500)
279
280             else:
281                 if option_choice == 0: #Check if user made a choice,
282                     self.answerstext_label.config(text="Sorry you didn't select
283                     anything, please retry")
284
285             else: #If they made choice that isn't last question.
286                 if option_choice == question_answer[qnum][6]: #If user is
287                     right.
288                     score+=1
289                     total_score.configure(text = score)
290                     self.question_change() #Run method for next question to
291                     come up.
292             else: #If the user chooses wrong answer.
293                 score +=0

```

```

TOTAL SCORE:
Exception: can only concatenate str (not "int") to str
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.8/tkinter/_init_.py", line 1883, in __call__
    return self._func(*args)
  File "main.py", line 278, in score_calculations
    total_score.configure(text = "TOTAL SCORE: \n" + score)
Exception: can only concatenate str (not "int") to str
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.8/tkinter/_init_.py", line 1883, in __call__
    return self._func(*args)
  File "main.py", line 278, in score_calculations
    total_score.configure(text = "TOTAL SCORE: \n" + score)
Exception: can only concatenate str (not "int") to str

```

This error had occurred because I attempted to configure a label to be a string text and a variable. From reading the error messages I realised that the programme can't configure the text label to be both things so simply I created a separate label that will constantly be the same text and then I configured another label so that the score and question number can be displayed each question.

How I addressed Relevant Implications:

Design:

In association to the relevant implication Intellectual Property, I have made sure to avoid a problem with any creator by creating my own images. These same images have been used throughout my programme for consistency. I have also created all the quiz questions so that I didn't fall under copyrighted content which means I have again addressed the intellectual property implication. I have done research and given some resources on realistic answers for my questions so that they weren't completely irrelevant and incorrect.

Relative to the usability implication, I have created a method that will provide an error message relevant to the error made by the user when using the quiz page. This makes sure the user can perform tasks correctly and relevant to the usability heuristic, they can diagnose and recover from

Outcome:

For the relevant implications of intellectual property, I did exactly as I designed and implemented the images and questions/answers I created into my code to avoid legal issues such as not having the right to use some other creator's creation.

I addressed the usability implication by following how my design addressed this and implemented it into my code. I have provided error messages that clearly state what the user did wrong and allow them to recover from it.

For the accessibility implication I followed my designed wireframe and made sure my widgets were big and easy to read especially for my end users with different eyesight issues meaning the programme is more accessible by them. Helvetica was also used to avoid problems for dyslexic users as it is a simple font to

	<p>errors. This allows them to efficiently use my quiz as an error message will pop up e.g. please select an option, and hopefully remember not to make the same error next time using my quiz.</p> <p>I addressed the relevant implication accessibility in my design by using the font Helvetica as it is the easiest font to read and by making sure my radio buttons and content were large enough for the users to read even with eye problems. This implication was important to ensure my end users have a good experience with my programme without struggle, so I intended to make my widgets clear and large for my end users especially with eyesight problems so that it is accessible.</p> <p>The very important implication functionality wasn't too relevant to my design but more so relevant to the actual programme itself. In my design I planned that my radio buttons would be used as answer options for the user to select when reading the question and the exit/next button function exactly as they are named.</p>	<p>read so that they don't mix up the letters and therefore feel less accessible to the programme.</p> <p>By following my design I have made sure my window is functional for the user as all the buttons perform the tasks that are expected and the user won't find trouble with widgets (mainly buttons) that function incorrectly.</p>
<p>How I applied design conventions</p>	<p>In my design I had applied design conventions such as alignment, fonts, colours, and font sizes to meet the user requirements and produce a good programme.</p> <p>I put the main widgets (my radio buttons) centred in the middle of the page, large for the user to read with the purpose that the text is clear for the user to read and that the buttons are options they are meant to select. It also makes the page more accessible to my end users as the content is simple and clear enough to understand. My other buttons are aligned to the left and right so that the user doesn't mix up the buttons together and are familiar that exit buttons are commonly used in the bottom left corner and the next button used in the bottom right. This allows the user to recognise my programme similar to other programmes as the placing of my widgets are very consistent with other quizzes. My logo remains consistent in the same position (top left corner) so the user remembers what quiz they are using, the total score and question number is also placed to the left aligned with the logo so that it creates space on the page and groups the values to the same place. These were used so the user can remember the question number they are on, how many correct answers they got and what</p>	

quiz they are using (relative to the usability heuristic recognition rather than recall) The buttons and font sizes are large so that my end user can easily read the content without struggle which makes my programme more accessible to my end users. Helvetica was the main font I used throughout my window as I researched it to be the most simple font for users to understand so that my programme is accessible to users who find reading hard or even have dyslexia. The colours I have used are consistent in my design, contrasting for light and dark themes meaning my title and widgets stand out from the background and allows the users to see it easier while being attracted to the design of the page as well.

All my design ideas were applied to my programme and real outcome.

Version Control

Github link:

https://github.com/TomaMirchevska/Toma_Quiz

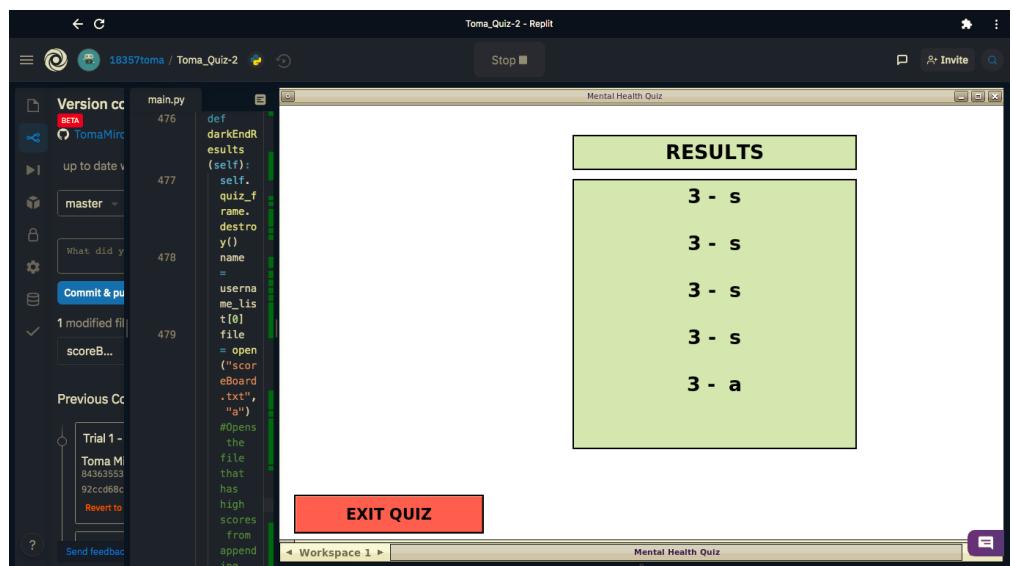
Component 3: ResultsPage and DarkResultsPage

Trialling:

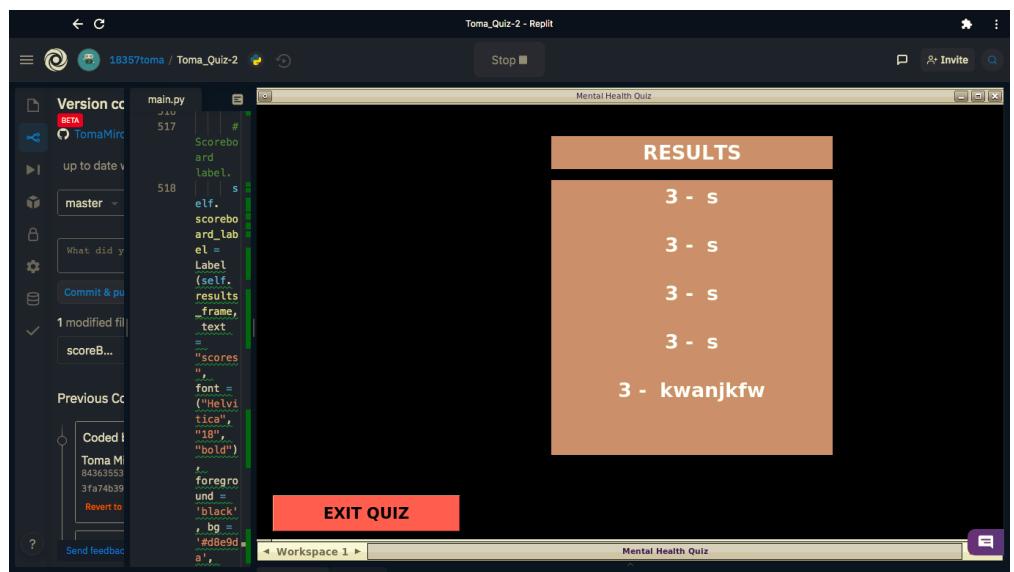
What techniques or components did I trial? Which did I choose and why?

Trial 1:

ResultsPage



DarkResultsPage



Techniques I have used:

- File.

One most important part of my Results window is to display the top 5 scores of whoever participated in the quiz. To do this I found it easiest to create a new file and append all the names with their scores to a list in that file and when the user is transferred to the results page, the file will be printed in the window with the top 5 scores and username.

- Place.

For this page I decided to use the place technique again as it is so easy to select y and x coordinates to place my widgets. Grid wasn't the best choice as I won't have many widgets for this window meaning I don't need to have multiple rows/columns. It is also easiest to change the coordinates if I didn't like the placement of any widgets compared to the grid layout.

- List.

At the beginning of my code I define the list `username_list[]`. This was relevant to my name collection and now the results page as all the names entered into the first window was appended to the `username_list[]`. Using this technique allowed me to display the usernames and similarly the scores from the `endResults` method into my `ResultsPage` class.

Differences to expected window:

- Due to a failed test I have decided to change my exit menu button for now into an exit quiz button so that the user can leave the programme at the end. I intend to try to configure my programme so that the testing can work as I have previously planned.
- There are no proper colour schemes or images yet which I intend to do in the next trial.
- In the coding aspect, I don't have a way to reset the results board so I intend to create a method to reset the scoreboard

file.

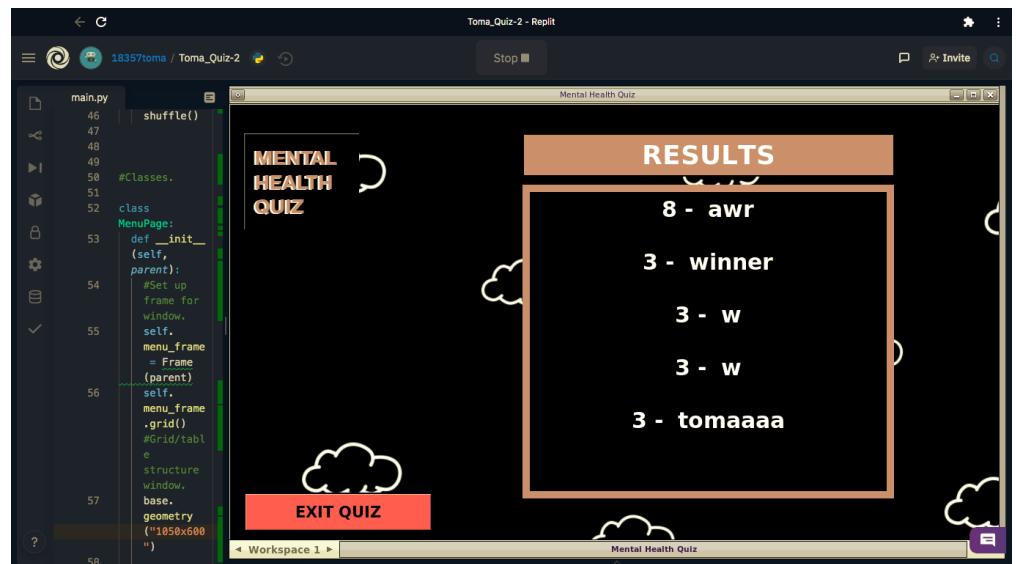
Trial 2:

For my second trial I tried to focus a lot on the design/aesthetic part of my window and made sure to have it as close to my wireframe as possible.

ResultsPage:



DarkResultsPage:



Techniques I have used:

- If/else statement.

I used this common technique in my method endResults to make a

way to reset the file opened (where the names and scores appended to a list is stored in) so that the score and name list is refreshed. The if else statement is determined by what the user inputs as their username, and if name = 'reset' then the programme is commanded to basically reset the scoreboard file.

- Image labels.

Basically the same as the other windows, I have inputted background image and logo image so that my results pages follow the same theme as my other windows for light and dark themes.

- Highlight background and thickness

This technique was used for the configured scoreboard label so that the effect of the box standing out attracts the user and brings a simple aesthetic to the page. The size is very thick so that the user can be redirected to the box and understand it is a list of scores.

Differences to expected window:

- I haven't managed to figure out how to properly implement a transparent background even after research so for the text labels I decided to match the background with the dominant background colour of bg image.

User Testing and Feedback:

ResultsPage and DarkResultsPage

Likes:

Nico: I like how the colours and images for dark and light themes are consistent with the previous windows.

Adil: The colours are contrasting which makes the title stand out a lot.

Tina: The widgets are large which attracts my attention and takes up blank space so that the page doesn't look empty which I like, it makes the page accessible to users with eyesight problems/wears glasses like I do. The page is also very functional as the button works to what I expect it to do, which is to exit the programme.

Improvements:

Nico: I don't really like how the widgets aren't completely centred. As a user it makes me feel slightly unsatisfied and confused.

Ansh: I suggest that you either make the results title and scoreboard the same width or make the title smaller so that it doesn't look awkwardly unaligned.

Adil: In my opinion the numbers and text displayed in the scoreboard might be hard to understand for people especially me so maybe you should label at the top of the board what the numbers and text represents.

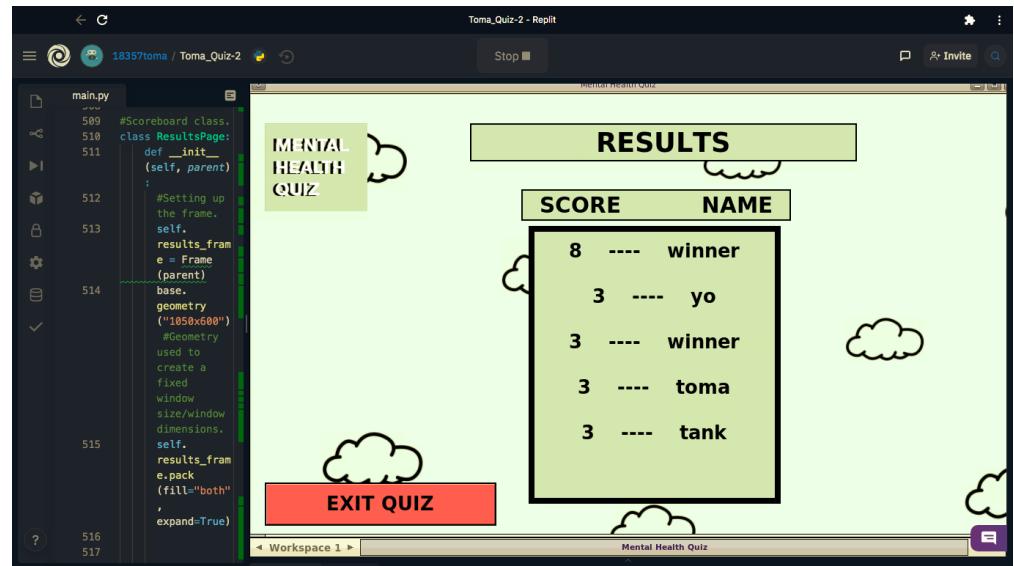
Tina: To be honest the scoreboard doesn't stand out as much as the title does so I recommend you make the background colours the same

so that they both are bold and look like related content.

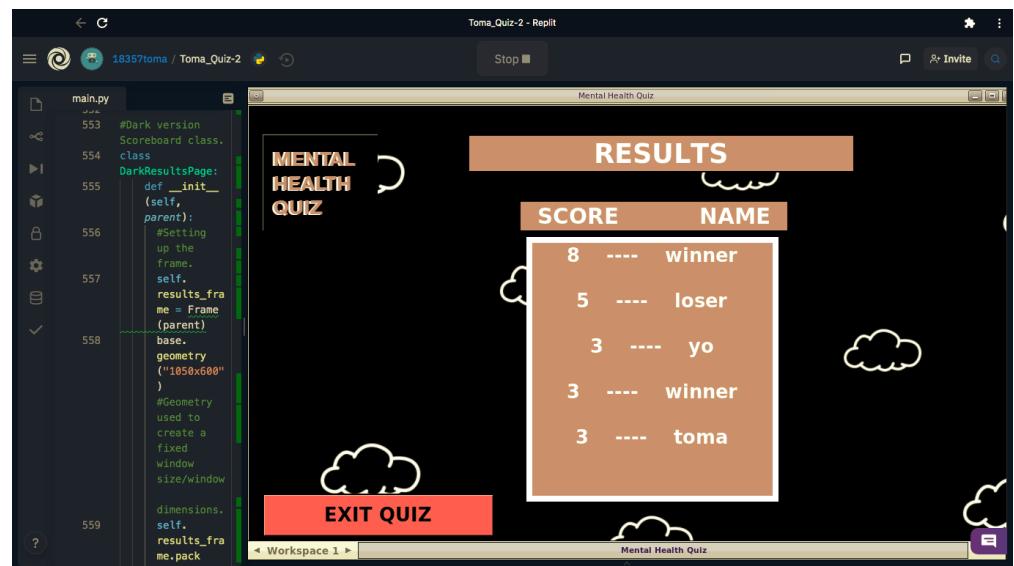
Final outcome: (According to user feedback)

For this final trial I tried to implement all of my end users' feedback for my previous trials so that my programme is suitable for the other users who can play my quiz.

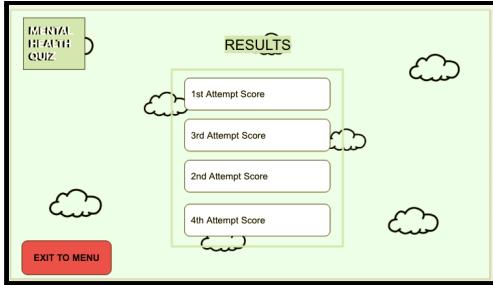
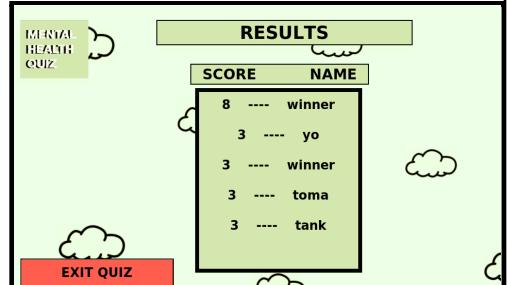
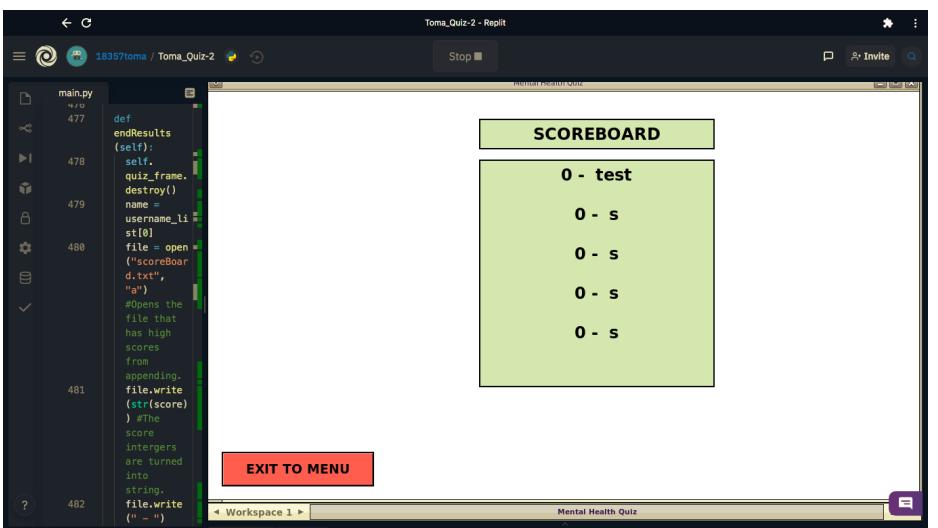
Results page:



DarkResultsPage:



Techniques I have used:

	<p>I haven't used any new techniques for this trial other than adjusting the current ones used such as width and font size for widgets to try to perfect sizes and alignments for this window and finish my overall programme.</p>	
Testing Test case	<p>Expected</p> <p><u>ResultsPage:</u></p>  <p><u>DarkResultsPage:</u></p> 	<p>Actual (screen shot)</p> <p><u>ResultsPage:</u></p>  <p><u>DarkResultsPage:</u></p> 
How did I fix it if it failed the test?	<p><u>Problem</u> - The score calculated wasn't passed through to the scoreboard file opened. Instead the default value of score (0) was used to append to the file lists.</p>  <p>This problem had occurred because I realised the strategy I was following to display the score text was by global variables not by</p>	

	<p>objects of a class. I fixed this simply by adding 'self.' to the 'score' variable so that the object value is passed onto the endResults method. Therefore, the scores display correct values.</p> <p><u>Problem</u> - When the dictionary questions_answers are being displayed in the quiz pages, it won't reset the values (amount of keys displayed) until the quiz is closed.</p> <p>This was because I realised my question_answers values won't reset (all dictionaries were already called) meaning the programme will constantly think the quiz was completed. The way to fix this simply is to change the 'exit' button on the results pages to an 'exit quiz' so that I can figure out later on if there is a way I can restart the test_progress method so the questions can all be reset to 0 being displayed (programme will think 0 questions configured into labels so far).</p>
<p>How I addressed Relevant Implications:</p>	<p>Design:</p> <p>In association to the relevant implication Intellectual Property, I have made sure to avoid a problem with any creator by creating my own images. These same images have been used throughout my programme for consistency. I made sure not to use any other creators' images so that I can avoid using copyright content in my programme or creations without permission.</p> <p>Relative to the usability implication, I have designed a button that is planned to exit to the menu page so that the user can efficiently restart or get off my quiz as they please instead of taking time to figure out how to get out.</p> <p>I addressed the relevant implication accessibility in my design by using the font Helvetica as it is the easiest font to read and making sure my results board and titles/labels were large and clear to the user. This implication was important to make sure my end users don't have a problem reading what is on my page and also enjoy having an experience without problems especially for those with eyesight issues. I designed my wireframe to make my widgets clear and large for my end users so that the window is accessible to them.</p> <p>Outcome:</p> <p>To do with my design, I made sure to continuously use the images I have created through this window like all the other windows to address the relevant implication Intellectual Property.</p> <p>The usability implication was addressed in my programme by putting a button that exits the whole programme instead of redirecting the user back to the menu as I thought it was more efficient for the user to be exited out of the programme instead of having to figure out how to exit by themselves.</p> <p>I addressed the accessibility relevant implication by following my design and making sure my results board as well as texts were large enough like all windows so that the user can see it easily and understand what I am saying. I also managed to have the listed words separated a bit so that users with dyslexia can avoid mixing up the texts/numbers. The font Helvetica was also used to avoid problems for dyslexic users as it is a simple font to read so that they don't mix up the letters and therefore feel less accessible to the programme.</p>

	<p>The very important implication functionality wasn't too relevant to my design but more so relevant to the actual programme itself. In my design I planned that my radio buttons would be used as answer options for the user to select when reading the question and the exit/next button function exactly as they are named.</p>	
How I applied design conventions	<p>In my design I had applied design conventions such as alignment, fonts, colours, and font sizes to meet the user requirements and produce a good programme.</p> <p>I put all my important labels and leaderboard centre aligned in my window so that the user is attracted to the point of the page as soon as they are redirected to it. My exit quiz button is in the bottom left corner which is consistent with my other windows and is similar to other programmes which will familiarise the user with my format. The fonts and widgets are large for the user to read with the purpose that the text is clear for the user to read and that the buttons are options they are meant to select. It also makes the page more accessible to my end users as the content is simple and clear enough to understand. My logo remains consistent in the same position (top left corner) so the user remembers what quiz they are using (relative to the usability heuristic recognition rather than recall). The content is spread out enough so that the user does not mix up the widgets and get lost or confused. The buttons and font sizes are large so that my end user can easily read the content without struggle which makes my programme more accessible to my end users. Helvetica was the main font I used throughout my window as I researched it to be the most simple font for users to understand so that my programme is accessible to users who find reading hard or even have dyslexia. The colours I have used are consistent in my design, contrasting for light and dark themes meaning my title and widgets stand out from the background and allows the users to see it easier while being attracted to the design of the page as well.</p> <p>All my design ideas were applied to my programme and real outcome.</p>	
Version Control	<p>Github Link:</p> <p>https://github.com/TomaMirchevsk/Toma_Quiz</p>	

5. Final Program Structure

- a. Flowcharts of the program(sitemap for website)

6. Final Outcome Testing

6.1 Did you comment on your code and described what it does?

6.2 Did you check you followed naming conventions?

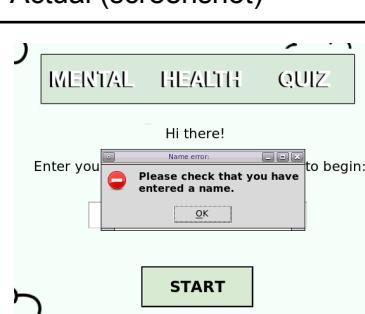
6.3 Overall final program testing

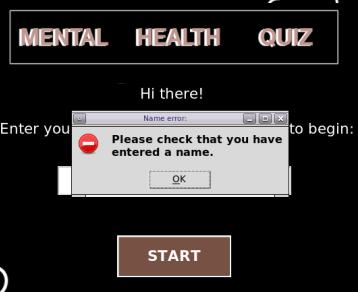
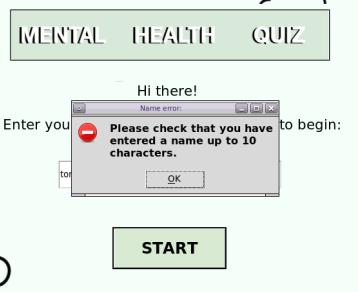
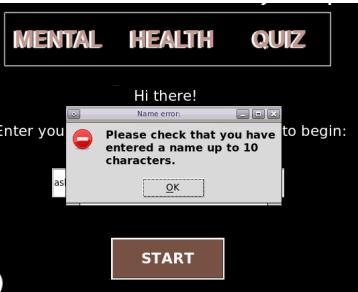
For merit:

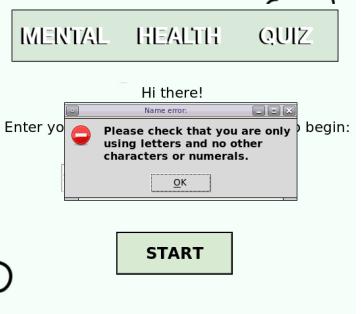
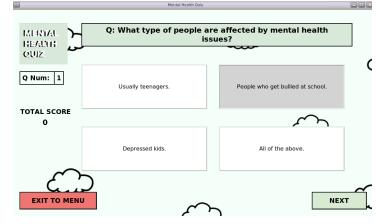
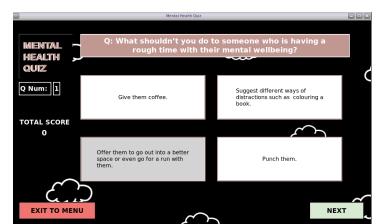
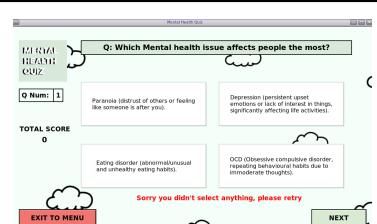
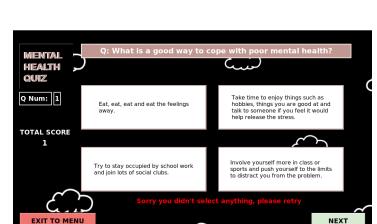
Tested and debugged the program effectively to ensure that it works on a sample of both expected cases and relevant boundary cases.

For excellence:

Tested to ensure that it works correctly for expected cases, boundary and unexpected or invalid cases.

Test Case	Expected	Actual (screenshot)	Pass? Changes?
No name entered in the entry box.	Error message box appears stating that no name was entered.		Pass.

			
More than 10 characters entered in the entry box.	Error message box appears stating too many characters were entered.		Pass.
Numerals/special characters entered.	Error message box appears stating that the user shouldn't enter numerals or special characters.		Pass.

<p>Unexpected case - More than 10 numerals/characters entered.</p>	<p>Error message box appears stating that the user shouldn't enter numerals or special characters.</p>		<p>Pass.</p>
<p>Clicking all radio buttons as an answer.</p>	<p>Only one radio button will be selected and will unselect each time another radio button is clicked on.</p>	 	<p>Pass.</p>
<p>No radio buttons clicked and proceeded to the next question.</p>	<p>Error message will state the user needs to select and answer.</p>	 	<p>Pass.</p>

7. Usability Testing and Feedback

- In my class I asked some students to test through my entire program and comment on the design of the windows, functionality, robustness and usability of the programme I created.

End User:	Likes	Errors and what changes were made to fix them
Safe	The aesthetic and minimalist design of the programme is very attractive and makes me feel positive using the code. I was able to use all the buttons/widgets and they all performed the action I expected them to, which makes the programme really functional as well as usable because the content was easy to understand and read.	No errors occurred.
Tina	The light and dark theme promote different feelings in me which I like as I can choose between which theme I feel to use depending on what I want. The designs and colours schemes are very consistent and the layouts are very familiar to me and users which makes the programme very usable to new users and the quiz is very functional (all errors that can occur have error messages for the user, no errors pass through the code).	No errors occurred.
Nico	I like the design and aesthetics of the components. I find it nice that the entire program is consistent with this theme as it is very nice. The theme is also very welcoming, good for those who may be sad or depressed which are the end users. I also liked the robustness of the program, able to change on a whim with no further affects on the functionality	No errors occurred.

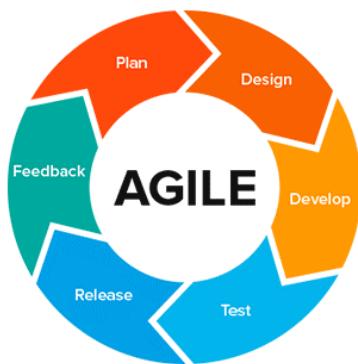
of the program. I have tried a number of boundary tests to try and break the program but I could not find an error which makes the code really strong and easily usable with the simple design layouts.

8. Evaluation

- Discuss how the information from planning, testing and trialling of components assisted in the development of a high-quality outcome.

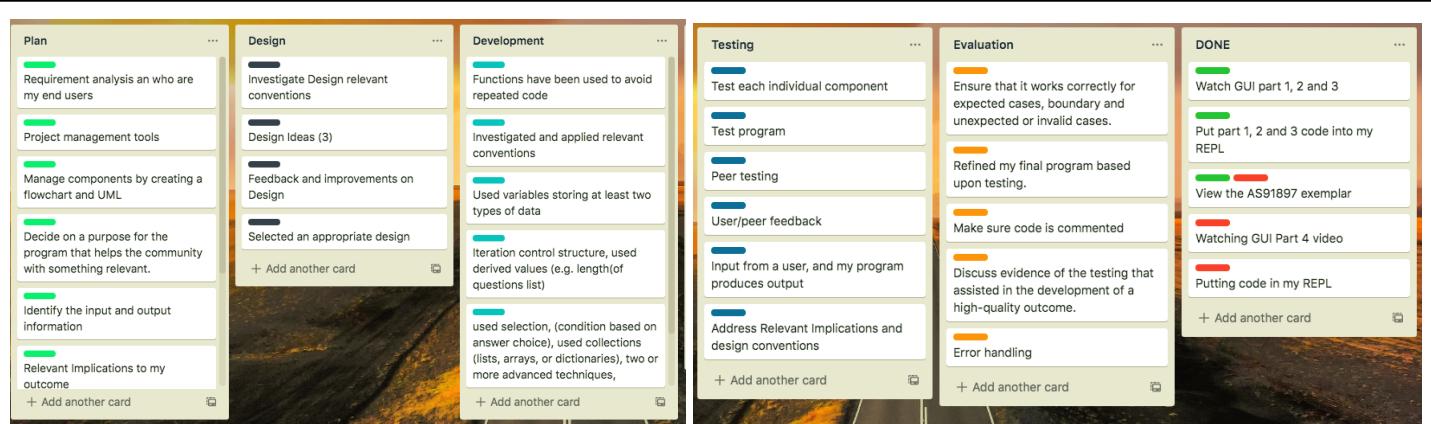
In my planning section I have used different project management tools such as (time management, for stress)

Agile Methodology:



Agile methodology is a project management tool used to format my documentation. This tool was very effective as I was able to follow the certain format without feeling lost or in the wrong direction. I was able to effectively work through the different sections with the help of my other project management tool trello board, which allowed me to complete each section to the best standard. This has allowed me to have beneficial time management and therefore less stress with this assessment as I followed each section over my time producing the program. I was able to use this method to effectively document my code in the most efficient way as well as produce the best version of my programme.

Trello board:



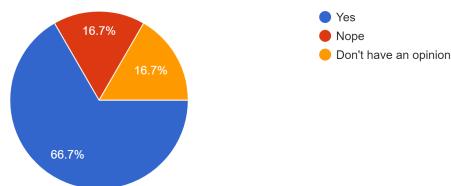
Trello board was one of the project management tools I used to control the way I document and code. This website is very useful to create checklists that I can look at after every section I complete to make sure I haven't missed any important parts of each section. I found this project management tool to be very beneficial as it allowed me to efficiently manage my project by having many different requirements as a checklist so that I know if I have fulfilled them or not. It has helped me manage my time and have less stress as all the work I need to do is already written down and if I ever got lost I was able to refer back to the checklists for my planning, designing, developing, testing, and evaluating sections.

Google forms:

Google forms was an essential way to get user opinions on my ideas and designs. I used google forms to send out questions and images of my design layouts to get important feedback on which options suit the programme the most so that the end users are satisfied and enjoy my quiz. Every selection I made in the planning section I had gained user feedback to decide which topic I should use for my quiz and what format I should follow.

For Example:

Do you think mental health is important?



In my planning section I sent this specific form out to see whether or not my quiz will apply to my end users at all. According to my results, the majority of my users said mental health is important to them and therefore I proceeded to use this topic for my programme. If my results were different however, such as no one felt that mental health was important, I wouldn't have selected this topic as my programme is made to satisfy my end users. I would have to select a new topic and send another google form out to gain feedback on whether it is important to the end users. This form of

project management was very helpful as I could efficiently gain feedback/votes from my end users without consulting them individually. This also meant that the results I get would be a big enough sample to make an informed decision that satisfies different users which makes my programme flexible as I have used different forms to get different feedback e.g. I implemented a dark and light theme because majority votes were equal for light and dark colours.

Repl.it:

A very important tool I used to code my whole programme was repl. This website contains different files which allows me to import and use different code on repl without having to download each file such as PIL onto my computer. This saved a lot of time for me and allowed me to efficiently work through my code without having to use different files. The most beneficial part about using repl was that it is accessible on all computer platforms meaning I was able to adjust my programme at home or on any computer at all times. This made my performance very efficient as I could constantly work on the code whenever I wanted to without having to download/transfer files. Since this savoured my time, I was able to put more effort into perfecting my programme to suit my end users.

Another part of repl is version control. This was my main project management tool specifically for my code. I used the website Github which is attached to repl which allowed me to commit all my major changes to code without having to document different window versions constantly. This helped me to efficiently code and complete my programme to a high standard as well as being able to fix mistakes by reverting or looking back at previous change commits.

- Explain how a number of options were trialled, and the most appropriate option was chosen and justified.

While creating my components, I went through multiple trials and in each trial I was able to identify differences from my expected outcome (wireframes) and what improvements I could make for my next trial. Each trial had different techniques used which I listed for the components and it allowed me to identify at any time which techniques I could change around or improve on. All my components were trialed at least 3 times and in the final trial I had followed user feedback that was given from classmates who tested my windows. From that I improved my windows to suit their requirements which justifies why I chose the last trial versions to produce an appropriate programme for my end users.

- Evaluate your design and justify how the selected design uses appropriate conventions.

Refer to the 3.1 Design section.

- How did you make sure functions have been used to avoid repeated code.

For my programme I decided to produce two separate colour themes 'light' and 'dark' themes. This meant that I would have to have completely different colour schemes implemented in each window. The most efficient way I found to do this was to duplicate each class and configure all of the colour labels/images manually. This meant that my code was slightly repeated for each window but in my opinion, it took away the complexity of trying to separately configure every single colour attribute of each widget. I also found that I had separate methods created in each of the dark and light themes which meant that I couldn't use one code for a class and configure each item.

I had avoided repeating code by using the `questions_change()` method. This method sets different values for each of the radio buttons and question labels (changes the question and answers) so that I don't have to spend time making a lot of code. If I didn't create this function, I would have to repeat the same class 10 times to have each of my questions displayed in the quiz. I avoided this massive repetition of code by creating this `questions_change()` method which is called in the `score_calculations()` method.

- Is it easy to extend the functionality of the code (e.g. a function has been used to check the menu choices, so it would be easy to update the menu to add or delete another item). Explain.

For my programme I have used many different classes and windows yet functions/a dictionary that can easily be changed which makes my quiz very functional and flexible. It is flexible as any user of my programme that wants to use my code can readjust my dictionary and any image labels such as the logo/title simply to make a whole new quiz. The dictionary is the only collection that needs to change if anyone wants to change my quiz which makes my programme very flexible as it is simple to configure to another persons' needs or wants.

- Have you used derived values? (e.g. `length(of questions list)` to iterate through a collection instead of hard coded values. List them

In my code I have defined an `asked` list used in the `shuffle` method, each random question number is added to the list, and there is a check once it's got to 10 it stops the iteration.

In my code I have used an iteration which is the `asked` list. This list appends random question numbers until all 10 questions have been used, then the iteration will stop. Using this derived value allows me to efficiently code different methods to change my questions rather than having to create many if-else statements for each separate question. This has made my code very efficient to make and a functional programme to use.

- What validation have you used? Error handling? Error messages?

Relative to the usability implication and the heuristic to diagnose and recover from errors, I have implemented error messages into my code to help the user have an easy time using my programme and be able to diagnose errors that have been made. In my menu pages, I have coded a method to display message boxes that I have imported into `tkinter` depending on the error the user has made relative to the entry box. If the user doesn't enter a name and presses the start button, a message box will pop up stating that the user has to enter a name to continue with the quiz. This technique simply helps the user perform tasks correctly next time and efficiently use my programme.

For error handling I made sure to avoid the problem of the user selecting all radio buttons at the same time using `IntVar` so that all buttons begin unselected so that the user doesn't make an error by pressing next

when the programme has selected the option from the user's previous answer of another question. The user also can't select all buttons which prevents them from making an error as they will realise not all answers can be selected which is relative to the usability heuristic error prevention.

- Have you used variables of appropriate scope (e.g. the variables that are only needed in a module should be declared in that module and not as a global).

In my code I have avoided using global variables by calling them within my classes. To do this I had to call the variables within the class and make it an instance by relating it to the class e.g. `score = 0`, add `self.score = 0` so it is an instance. This allowed me to pass the variable through the classes or within a class without having to global it which makes my code more robust as the variable won't be affected outside the classes. I have used the same technique with my question number calculations as well so that I don't have to global the variable as they are only needed in the module so must be declared in the module.

8. Acknowledge own work

Please sign your name and date to acknowledge that the program/website and this documentation is your own work

Name:

Date: