8/19/2022

Kyle Goosey (D12 Year 13)

[company name]

Computer science NEA

By Kyle Goosey

Contents page

Analysis page 2

Design page 11

Development and testing page 35

Evaluation page 72

Final code page 77

Section 1- analysis

Problem definition

Social workers are overworked and underappreciated. The near-impossible job of taking some of this country's most vulnerable people and getting them the best possible care is one that many cannot do. However, truly little support is given, and funding for children's services has been slashed by a third since 2010, and in 'real terms', this could be as high as 52%. This leaves upwards of thousands of children at risk of falling into 'abuse and neglect', according to the Independent newspaper. Councils are facing a £3 billion funding gap for children's services by 2025. According to the same article, this is before the current cost of living crisis impacting the UK. Tom Isaac, a youth worker, said, "A lot of young people are already up against it because of the violence they see around them and often their family situations. They are always looking for belonging, identity, and purpose; if they cannot find it in a healthy place, they go to an unhealthy place for it. That risk needs to be mitigated."

These conditions are not fit for any child to grow up in, leading to a child potentially being placed in the state's care. However, this is known to have its issues. *A chief case of the failings in protecting children is the example of Charlotte from Morecambe, whom BBC Newsnight interviewed, "16-year-olds go in completely fine [and] coming out with ketamine addictions"*. So clearly, there is a problem at hand, but the question is how to solve it. There is a way to help improve the outcome of those in care: to create a way for the best possible housing choice for the children entering care. This is what my code will set out to achieve.

My clients

My clients are social workers working with vulnerable children. They want a way to decide the best option for housing for those in need of care. They also want a way to track a child's process through the years and see in what areas they could improve to be the best person they could be. I will reach out to these clients, find the best way to approach this problem and get a list of essential requirements.

Why is it suited to a computational solution?

This project is suited to computational methods for a variety of reasons. The problem requires a lot of data to be processed quickly. It also requires an easily searchable database to pull out previous results and data points. Examples are below:

Abstraction

As a large amount of data will go into making an assessment, large amounts of abstraction will be used. An example of this is simplifying each answer the social worker gives from a long paragraph down to a number, giving a concise but ample overview of the child in this area. This removes much of the unnecessary detail, thus making the best use of hardware performance and the user's time.

Decomposition

The problem is complex and would be hard to explain and understand without breaking the problem down. For instance, breaking the problem down into the different sections the user will encounter in order, the login page, a data retrieval/entry system, the questionnaire, and finally, a set of graphical outputs and actionable opinions in future actions. This process of breaking the problem down allows me to work on parts in order of importance rather than in chronological order, which would make coding much harder. This means I am using the 'divide and conquer' method.

Branching

For the code to have any function, it must have branches, as without it the user would have a completely linear path, meaning that no decisions can be made and the code would have the same output every time.

Current solutions

Chart, line chart

Description automatically generated'BERRI' is "the only online assessment tool to cover mental health, behaviour, emotional wellbeing, relationships, risk, and attachment. It was developed by a clinician with extensive experience of Looked After Children and those with complex needs through focus groups with foster carers, residential care staff, and a variety of other professionals." It consists of a short questionnaire about the child and outputs a set of graphs to help compare a child to the previous test completion. I like the automatic graph output at the end and the login system with so-called 'super users' who can access more than standard accounts. I also like that it backs up to a database. Example of a BERRI output

Description automatically generated

Another current solution some councils employ is using a spreadsheet, as below. Any graphs or actionable outputs need to be created manually, which comes at the risk of user error and the cost in person-hours. I like the brief description at the top, clarifying the scale and how children should be evaluated.

Table

Description automatically generated

This demonstrates the need for my software now, as councils must choose between using BERRI at great expense, around £275 per record per year, or using spreadsheets, which provide extraordinarily little valuable data.

**A final example of the software I will base my code on is Excel and its ability to make Nightingale Charts. A Nightingale Chart is a type of visualisation used to display quantitative data in a circular format. It consists of rings, each representing a specific value or category, and lines dividing the chart into segments. The radius of each segment is proportional to the value it represents. An example of the chart can be found below.**

**Chart, radar chart

Description automatically generated**

**I like this method of representing the data better than the Berri example, as I feel the data is easier to read and understand at a glance. I also prefer the use of the circular radius, as opposed to the pentagon used in the Berri example. I prefer it as it will allow for easier maths in the following sections as circles are easier to define than pentagons.**

**From the research on similar solutions, I have concluded this list of basic requirements:**

* Free to use
* Basic UI
* Use a list of questions and numbered responses to give an overview of a child’s needs.
* Link to a SQL database.
* Give actionable advice based on children's previous outcomes such as potential housing options.
* Output visual pieces of data, in order to quickly give an overview of the child’s changes in behaviour.

This will be combined with any additional requirements for my client to give the basic outline of my code.

Limitations

One limitation of my solution is that I will not be able to give outputs on which treatment courses are most effective like BERRI does due to my current lack of data, and I am not willing to make those kinds of statements unless I am sure they are true. I can only give a partial overview of a child, as collecting all the data needed for a full assessment based purely on numbers is impractical. This means that a certain level of expertise will still be required for the social worker to interpret the results in a context the computer does not have.

Investigation and analysis

I plan to interview my clients through a questionnaire. I will briefly overview the software for the first questionnaire and what they hope it will achieve. From this, I have decided to ask these questions:

1. Have you used similar software to this before, if so, which ones?
2. What components of the software did you like?
3. What features did you dislike?
4. Are there any new features that you would like to see introduced?

I asked three different professionals in this area for their answers. These are detailed below:

First response:

1. Yes, I have used BERRI and a spreadsheet.
2. I liked that BERRI automatically produced graphs and how they separate the different domain of the child’s psychology. I also like ability to look at previous assessments and compare to the current one to find improvements. I like that the spreadsheet can be updated to reflect the current climate more accurately.
3. I dislike the cost of BERRI and the advice given is generic and gives no real help. I dislike that the spreadsheet cannot output graphs or anything useful.
4. I would like to see improved outputs, such as the pieces of advice.

Second response:

1. I have used a spreadsheet.
2. I like that it is clear and easy to use, with error checking.
3. I dislike the lack of further output, meaning everything must be done by hand, which is very time-consuming.
4. As above, add a series of outputs like a dynamic tool that is used to measure progress over time, and which provides a profile of the child’s needs that then informs their care plan and the provision of support to them.

Third response:

1. I have used BERRI
2. I like the login system, as it feels more secure and how it keeps data safe.
3. I do not like that I can’t attach this data to the child’s file, making the process feel disconnected and janky.
4. As above

Analysis

I used the first question to determine how much experience the interviewees had with similar software. This was included as it allowed me to give precedence to those with the most experience in the topic, as these people are likely to be best at describing what makes a good one good. Questions two and three were included as they allowed me to gain greater insight into the software from those more versed in it than me, as they all have real experience using this software. These questions were left intentionally open-ended, not asking about the best or worst features, as it allowed the interviewees to write freely. In turn, I received constructive answers. Question 4 was similar as it was both open-ended and allowed my interviewees to speak their minds and give honest answers.

Software and hardware requirements

The code will be written in Python with elements of SQL. The following are the minimum requirements for my code to run.

Software requirements

|  |  |
| --- | --- |
| Requirement | Justification |
| Windows 10 TH1 1507 (released July 2015) | Operating system required to run SQL |

Hardware requirements

|  |  |
| --- | --- |
| Requirement | Justification |
| 2GB of RAM | Needed to run Python 3.0 |
| X64 processor with 1.4GHz | Needed to run SQL |

Stakeholder requirements

Design

|  |  |
| --- | --- |
| Requirement | Justification |
| Lightweight design | Needs to be simple and intuitive to use. |
| Clear instructions | Include clear and obvious help, such as a scale, to make it easy to use the software with little to no training needed. |

Functionality requirements

* Free to use.
* One of the main drawbacks of BERRI was the high cost. With my services, it is right not to charge users. It would also be wrong to charge users as the high costs often price them out of using the software, leading to a worse but free alternative in the spreadsheet, meaning that due chiefly to budget constraints, the children in care cannot get the best care they deserve.
* Basic UI.

As it is a backend-focused project, creating a 'good' or complicated UI is unnecessary, as this would only make the software harder to run and less clear to the user for no tangible benefit.

* Use a list of questions and numbered responses to give an overview of a child’s needs.

This is the basic premise of my code and must be included.

* Link to a SQL database.

This must be a large factor of the software, as the most important thing is the child’s safety, and as such, any data stored about them must be kept safe and secure. Also, keeping data allows for comparisons to be made and improvements to be shown, and any warning signs can be detected early, hopefully helping to nip it in the bud.

* Give actionable advice based on children's previous outcomes, such as potential housing options.

Another of the complaints about BERRI was the lack of real suggestions to help with any potential problems it was finding. Therefore, I have decided to use machine learning to try to find patterns within children and give real, useful outputs to try to help any future problems, by placing them in a place they can thrive.

* Login system

This will add to security as it will allow only those with privileged access to view any raw data stored about a child.

Success criteria

|  |  |
| --- | --- |
| Criteria | How to evidence |
| Login page with username and password | Screenshot of the window that shows the GUI with the login page |
| Menu page with different options | A screenshot shows a simple menu with multiple buttons, such as create a new record or add to an old record. |
| A simple test screen | Screenshot that shows the questionnaire being completed, with an input that only allows integer inputs, with a back button option to correct any errors entered |
| A simple help menu is available throughout | Screenshot showing a help guide, with headers and screenshots guiding the user through the process of completing a test. |
| A lightweight design | Screenshot showing the non-cluttered and easy-to-understand menu and test screen |
| The option to search the database for previous entries to append to | screenshot showing the option to append to a previous data entry. |
| Set of graphs output | Screenshot showing completed graphs from a test. |
| Pieces of advice on what to do | Screenshot with pieces of advice, depending on the user’s score. |

I confirmed with my stakeholders that these success criteria were applicable to their requirements and they were happy.

Flow chart

Diagram

Description automatically generated

This is the flowchart on which I will base my coding.

Design

User interface design

Login screen

Graphical user interface

Description automatically generated with low confidence

This is the first screen the user will reach when they open the program. It is a simple login page with a ‘help’ button present, which will be present throughout the program’s runtime. This button, when clicked, will load up a page with basic instructions on how to use the program and get the most out of it.

This links to my success criteria of a ‘help’ button present throughout.

The use of a login system and an easy-to-use interface are also present in this image and thus show two more links to my success criteria.

Menu page

Graphical user interface, application

Description automatically generated

After logging in, this will be the next screen the user will be presented with. It has 2 or 3 options shown (depending on the level of the user). The first two options are available to both user types. The ‘complete evaluation’ button will take the user to another menu shown below. The ‘view historical data’ button will allow the user to view the information of any person currently in the database, including their data. The final button is the ‘add account’ button. This will ask for a username and password to be added to the signing-in database and will allow the user to choose whether the new user is an admin or not. Again as potentially sensitive information could be in jeopardy, this power to create accounts of any kind is strictly for admin accounts only.

Links to success criteria:

Menu page with different options

ability to view and append to previous entries.

Complete evaluation page

Graphical user interface, application, Teams

Description automatically generated

This page will load after the user clicks the ‘complete evaluation’ button. It will have two possible options for the user to select. The ‘append to record’ button will open a screen asking the user to enter the applicable person’s ID number and will then begin the evaluation. The ‘add new record’ button will load a page asking the user to enter personal details about the person on whom they are performing the test, namely their full name and date of birth. After this, the user will be presented with the given ID of the new person, which can be used the next time the person is examined.

Question page

Graphical user interface, text, application

Description automatically generated

This is the page the user will be presented with when completing an evaluation. The question will appear in the box, along with the question number and out of how many questions. The ‘back’ and ‘next’ buttons will allow the user to freely move between the questions and allows them to revisit a question if any errors have been entered.

After this section is completed, the final group of pages will load. There will be three in total: one for the line graph, the Nightingale Chart, and finally, one for the written output.

Link to success criteria:

A simple test screen.

The final screen the user will access is an output page containing both graphs and advice tailored to the specific situation.

Stakeholder feedback

I sent these designs to my stakeholders along with this email.

*I hope this email finds you well. I wanted to reach out to you regarding the UI that I have created. I would greatly appreciate any feedback on the design and whether you feel it is user-friendly. Additionally, I would be grateful for any suggestions or improvements you might have for the UI, as I want to ensure that it meets the needs of all users. Please let me know your thoughts at your earliest convenience.*

I look forward to hearing from you.

My first two stakeholders were happy with my UI choices and stated:

*‘Thank you for sending over the UI for review. After taking a look at it, I am pleased to say that I like it overall and have no real issues with it. The design is clean and easy to navigate, and the layout makes sense. I think it will be a great asset to the project. If I have any further thoughts or suggestions, I will be sure to let you know. Otherwise, I think this UI is good to go.’*

And

*‘I wanted to give you an update on the UI you've created. I've had a chance to review it and I'm very impressed with the design. It is intuitive, visually appealing, and easy to navigate. It is a great enhancement to the project. I couldn't find any issues that need to be addressed. If any concern arises in the future, I'll make sure to let you know.’*

My final stakeholder replied with this:

*‘Thank you for reaching out and asking for feedback on the UI. Overall, I like the design and think it is very user-friendly.*

*However, I do have a suggestion for improvement. I believe that the UI could be made more accessible for users who are visually impaired. For example, providing larger font options, and high-contrast colour schemes would greatly benefit this user group. I appreciate your attention to accessibility in the design process, and I look forward to seeing these improvements incorporated into the final product.’*

With this in mind, I will include a high-contrast colour scheme and provide large, dyslexia-friendly fonts to ensure the most significant number of people can access my program.

Algorithms

This is the flowchart on which I will base my coding:

Chart, box and whisker chart

Description automatically generated

Note that the ‘help’ button would be available throughout the code’s run time and is not on the algorithms tree. Also the two items denoted with an \* having nested functions and will therefore be shown in more detail below.

First, this is the display questions function in more detail.

Diagram

Description automatically generated

And this is the Nightingale Chart in more detail.

Chart, box and whisker chart

Description automatically generated

This shows the process by which I will be working to complete the project. The main challenge of the whole code will be generating the required outputs as they will need a large amount of custom code and may take a long time to develop to a high level successfully.

The two main components of the code are the completion and correct storage of the questionnaire and the outputs of the code. These can be further broken down into these components.

Completing the questionnaire:

Allowing the user to create/ append to a profile,

Giving the evaluation,

Calculating the results,

Storing these results in a logical table.

Outputs:  
creating code to generate a spider graph at the end of every assessment,

Creating code that shows progress from session to session and longer-term basis,

Using AI to suggest different outcomes for children and any preventive measures that could be useful,

The code’s most challenging part will be training an AI to suggest outcomes.

KNN

The specific algorithm I plan on using to solve this problem is the k-nearest neighbour (KNN) algorithm. The idea behind KNN is that an object can be classified based on the majority vote of its k-nearest neighbours, where k is a positive integer. The algorithm stores all available cases and classifies new ones based on Euclidean distance to previously entered data points.

To find the nearest neighbours to the new point to be added, I will take the 5 data points that will be output by the end of the questionnaire (Behaviour, Emotional response, Risk (to self and others), relationships, and Indicators, these will be explained in more detail below).

Then I will use the formula.

Text, letter

Description automatically generated

To find the Euclidean distance between every point currently stored in the database and the new point to be added will find the difference between the new point and the previous points on each of the 5 planes square that add it to the other distances and take the square root of that answer. After this, I will take the KNN and take the reciprocal of each point’s Euclidean distance to the new point, as this will make the points closest to the new input weighted the most in the final calculation. After this, I will find the outcome associated with each of the nearest neighbours and add the reciprocal values for each outcome. After this is completed, the highest value will output as the suggested outcome, and this outcome will be attributed to the new point and added to the dataset.

I have chosen the weighted version of the KNN algorithm because it allows for fewer heuristics when selecting a value for k. This is best shown by the image below. It shows that the value chosen for k is important as it can change the algorithm’s outcome. When k=3, the new point is assigned to be a triangle, but at k=5, the new point is a square.

A picture containing text, dark

Description automatically generated

This problem can be lessened by using a weighting method. As the triangles are closer, the new point will be a triangle, no matter the value of k in this example. Another advantage of using a weighting algorithm is that it reduces the impact of outliers in the dataset. Even if the outlier is a nearest neighbour, its value will be largely ignored by the algorithm due to its high distance from the new point. While creating and totalling the weight will take time and reduce the program’s speed, it is worth including in my code due to its ability to reduce the impact of noise in the output of the code.

Therefore, a weighted KNN algorithm is the best way to group my data and effectively give the most accurate outputs.

The evaluation method

As mentioned above, each time a person is evaluated, they are scored on five different factors:

Behaviour: This refers to an individual's observable actions and mannerisms, which can provide insight into their mental and emotional state. Behaviour can include verbal and nonverbal communication, as well as physical actions.

Emotional Response: This refers to the feelings and emotions an individual experiences and displays in response to various stimuli. Emotional responses can range from positive emotions, such as joy and excitement, to negative emotions, such as anger and fear.

Risk (to self and others): This refers to the potential danger or harm an individual may pose to themselves or others based on their behaviour, emotional response, and other factors. This can include physical harm, emotional harm, or harm to property.

Relationships: This refers to the connections and interactions an individual has with others, which can provide insight into their social and emotional functioning. Relationships can include familial relationships, romantic relationships, and friendships.

Indicators: This refers to signs or markers that indicate the presence or severity of certain behaviours, emotional responses, and other factors relevant to an individual's mental and emotional well-being. Indicators can include changes in behaviour, changes in emotional response, or changes in physical symptoms.

Together, these five elements can provide a comprehensive picture of an individual's mental and emotional well-being and help to identify areas of concern that may require further assessment or intervention. Understanding these elements can also help professionals to develop effective treatment plans and support systems.

Each category will be scored over a series of questions, scored either 0, 1, or 2, where two is the worst. At the end of the questions, the total of each category will be stored in the SQL database under the name of the person who was evaluated.

Subroutines

Now that I have decomposed my problem into smaller, more manageable parts, I can begin to plan how each part of my code will work.

Login

This is the first section that the user will interact with. It is a simple login system using a username and password retrieved from an SQL database.

Pseudocode algorithm

FUNCTION login(username, password)

SET conn = CONNECT TO SQL DATABASE

SET query = "SELECT \* FROM users WHERE username = 'username ' AND password = ' password + '"

SET result = EXECUTE query on connection

IF result IS NOT EMPTY

RETURN "Login successful."

ELSE

RETURN "Incorrect username or password."

END IF

DISCONNECT FROM connection

END FUNCTION

Add\_user.

The program will need to be able to add users from the program itself and not have to enter the data through the SQL database. Three pieces of data are required in order to create an account within the system. First, a unique username is checked by the code. Second, a password of at least eight characters to protect the system’s security, and finally, a Boolean option to choose whether the new account will be an admin account.

Pseudocode algorithm

FUNCTION add\_user(username, password, is\_admin)

SET conn = CONNECT TO SQL DATABASE

SET query\_check = "SELECT \* FROM users WHERE username = ‘username’”

SET result\_check = EXECUTE query\_check on connection

IF result\_check IS NOT EMPTY

RETURN "Username already in use, please choose another."

ELSE

IF LENGTH(password) < 8

RETURN "Password must be at least 8 characters long."

ELSE

IF is\_admin

SET query = "INSERT INTO users (username, password, is\_admin) VALUES ('username', ‘password', 1)"

ELSE

SET query = "INSERT INTO users (username, password) VALUES ('username', 'password',0)"

END IF

EXECUTE query on connection

END IF

END IF

DISCONNECT FROM connection

END FUNCTION

Ask\_questions.

This code will take a list of questions and output them to the tkinter window and a list of radio buttons to input the answers. These values will then be added to the relevant score in order for the outputs below to be calculated.

FUNCTION loadquestion():

SET conn = CONNECT TO DATABASE project.db

SET c = conn.cursor()

EXECUTE SELECT \* FROM questions ON c

SET rows = FETCHALL FROM c

SET questions = []

FOR row IN rows:

SET question\_text, answer\_text = row

SET answers = SPLIT answer\_text BY SEMICOLON

SET question = {"question": question\_text, "answers": answers}

APPEND question TO questions

CALL displayquestions WITH questions

FUNCTION displayquestions(questions):

SET current\_question = 0

SET user\_answers = []

FUNCTION displayquestion(index):

SET question = GET question AT index FROM questions

SET question\_text = GET question["question"]

SET answers = GET question["answers"]

SET frame = CREATE NEW FRAME

SET question\_label = CREATE NEW LABEL WITH TEXT question\_text

SET var = -1

FOR i IN RANGE LENGTH(answers):

SET button = CREATE NEW RADIO BUTTON WITH TEXT answers[i]

SET button VALUE TO i

SET button variable TO var

Pseudocode algorithm

FUNCTION nextquestion():

SET selected\_answer = var.get()

IF selected\_answer != -1:

APPEND selected\_answer TO user\_answers

DESTROY frame

IF current\_question < LENGTH(questions) - 1:

INCREMENT current\_question BY 1

CALL displayquestion WITH current\_question

ELSE:

CALL displayresults WITH user\_answers

ELSE:

DISPLAY ERROR MESSAGE "Please select an answer."

SET next\_button = CREATE NEW BUTTON WITH TEXT "Next"

SET next\_button COMMAND TO nextquestion

IF index > 0:

FUNCTION prevquestion():

POP LAST ITEM FROM user\_answers

DESTROY frame

DECREMENT current\_question BY 1

CALL displayquestion WITH current\_question

SET back\_button = CREATE NEW BUTTON WITH TEXT "Back"

SET back\_button COMMAND TO prevquestion

Nightingale Chart

This code will take the values from the answers to give five different values on which to base the outputs. For each number, it draws a line from the centre of the canvas to the point that is a certain distance away from the centre. This is based on the value of the number and at a certain angle, determined by dividing 360 by the number of values in the array and multiplying by the current iteration number.

CLASS NightingaleChart:

FUNCTION \_\_init\_\_(self, flist, width=, height=):

SET self.flist = flist

SET self.width = width

SET self.height = height

SET self.cx = width/2

SET self.cy = height/2

SET self.max\_radius =

SET self.labels to a list of 5 labels

SET self.N to the length of flist

FUNCTION draw\_sector(self, canvas, radius, start\_angle, end\_angle):

points = [cx,cy]

FOR angle in range(start\_angle,end\_angle):

CONVERT the angle to radians

CALCULATE x = self.cx + radius \* cos(angle)

CALCULATE y = self.cy + radius \* sin(angle)

APPEND [x,y] to points

CREATE a polygon object on the canvas using the points list

FUNCTION draw\_dotted\_circle(self, canvas, radius):

FOR angle in range (0,360,3):

CONVERT the angle to radians

CALCULATE x = self.cx + radius \* cos(angle)

CALCULATE y = self.cy + radius \* sin(angle)

CREATE an object with the calculated x-coordinate and y-coordinate

FUNCTION draw\_axes(self, canvas):

FOR each of the rings:

CALL draw\_dotted\_circle (canvas,radius of the circle)

FOR each of the sectors:

CALCULATE rad = (2 \* pi / N) \* j

CALCULATE x = self.cx + (i / 5) \* self.max\_radius \* cos(rad)

CALCULATE y = self.cy + (i / 5) \* self.max\_radius \* sin(rad)

CREATE text object with the calculated x-coordinate, y-coordinate, and the corresponding label

FUNCTION draw\_chart(self):

CREATE a canvas object with the given width, height

PACK the canvas onto the root window

CALL draw\_axes

FOR each index i and its corresponding value in flist:

CALCULATE radius = (value / 10) \* self.max\_radius

CALCULATE start\_angle = 72 \* i

CALCULATE end\_angle = 72 \* (i + 1)

CALL draw\_sector (given canvas, radius, start angle, end angle)

CALCULATE averrad = radians((start\_angle + end\_angle) / 2)

CALCULATE label\_x = self.cx + self.max\_radius \* cos(averrad)

CALCULATE label\_y = self.cy + self.max\_radius \* sin(averrad)

CREATE text object(label\_x,label\_y, labels[i])

Line graph

This code takes in a new value representing the latest test score. Inside the function, it connects to an SQL database and retrieves the data of previous test results using a SELECT statement. The result is stored in the variable "result" and a for loop is used to iterate through each value in the result and append it to a data array. The new value that was passed as an argument is also appended to the data array. After that, it creates a new line graph; plots the data array on the chart; creates a title, x-label, and y-label for the graph; and displays the graph.

FUNCTION linegraphload()

SET conn = CONNECT TO DATABASE project.db

SET c = conn.cursor()

EXECUTE SELECT total FROM tests WHERE id = 'id'

FETCH all the results from the SQL statement and store them in a variable called flist

CALL the linegraph function

FUNCTION linegraph()

CREATE a canvas object with a width of x and height of y

CREATE a horizontal line for the x axis

CREATE a vertical line for the y axis

SET maxvalue = 50

CALCULATE xscale = x / len(flist-1)

CALCULATE yscale = height of canvas / maxvalue

FOR i in range(0,maxvalue,5):

CALCULATE y = height of canvas - i\*yscale

CREATE a text object on the y axis with i

CREATE a dashed horizontal line across the length of the graph

FOR each number i and its corresponding value in flist:

CALCULATE x = i\* xscale

i = i +1

CREATE a text object width x on the x axis canvas with i

FOR each number i and its corresponding value in len(flist-1):

CALCULATE x1 = i\*xscale

CALCULATE y1 = height of canvas - value\*yscale

CALCULATE x2 = (i+1)\*xscale

CALCULATE y2 = height of canvas - flist[i+1]\*yscale

CREATE a line from (x1, y1) to (x2, y2)

CREATE label on x axis with "Test number"

CREATE label on y axis with 'score'

KNN

This code takes the five values given by the display results function, then compares its values to all the previous data collected by the program and finds the distance between the 2 points. It then inverses this value in order to give a ‘weighting’ to each point. After this, all the weights are grouped depending on their outcome value. Once all the weights have been grouped, the highest weight is found, and the respective outcome is outputted. As well as this, I have added a confidence score. In essence, it is the percentage of the total weight the winning weight got.

FUNCTION knn():

SET conn = CONNECT TO SQL DATABASE

EXECUTE SQL QUERY "SELECT test\_array FROM tests"

FETCH RESULTS FROM QUERY

STORE RESULTS IN nlist VARIABLE

total = 0

weightlist = []

FOR i IN nlist:

FOR j IN RANGE 0 TO 3:

temp = nlist[i][j] - tarray[j]

temp = temp \* temp

total = temp

distance = SQUARE ROOTtotal

total = 0

IF distance = 0:

INCREMENT i BY 1

EXECUTE SQL QUERY "SELECT outcome FROM tests WHERE testid = {i}"

FETCH RESULT FROM QUERY

OUTPUT RESULT

ELSE:

weight = 1 / distance

APPEND weight TO weightlist

EXECUTE SQL QUERY "SELECT outcome FROM tests"

STORE RESULTS IN rflist VARIABLE

acount = 0

bcount = 0

ccount = 0

fvalue =''

FOR I IN rflist:

IF rflist[i] = 'A':

acount += weightlist[i]

ELIF rflist[i] = 'B':

bcount += weightlist[i]

ELSE:

ccount += weightlist[i]

IF acount >= bcount AND acount >= ccount:

SET fvalue = 'A'

SET temp = 'Care home'

SET wcount = acount

ELIF bcount > acount AND bcount >= ccount:

SET fvalue = 'B'

SET temp = 'Foster care'

SET wcount = bcount

ELSE:

SET fvalue = 'C'

SET temp = 'Supervised living home'

SET wcount = ccount

conscore = wcount / (acount + bcount + ccount)

SET conscore = "Confidence level: " + conscore

OUTPUT temp

OUTPUT conscore

As I am using SQL, I need to make sure that SQL injection is not possible, as that could lead to data being leaked or deleted if the wrong person was to get access to the system. The best way to prevent SQL injection is to use parameterized queries. This method involves separating the SQL code from the input data and then using placeholders in the SQL code to represent the input data. The input data is then passed separately to the database server and is securely inserted into the placeholders. This way, even if an attacker tries to insert malicious SQL code into the input fields, the placeholders will ensure that the code is treated as data rather than as executable code. However, for my needs, this is overcomplicating what is needed, as instead, I will only allow alphanumeric characters to be used in usernames and passwords, as well as only allowing integer values to be test results. This will stop the use of special characters entering the code, meaning that SQL injection is impossible. The Python function .isalnum can achieve this ()

Solving this problem using code is well-suited for several reasons, mainly because of the concepts of abstraction and decomposition.

Abstraction allows the creation of a login system linked to an SQL database to be simplified and made more efficient through the use of code. By using code to automate the process of connecting to the database, retrieving the necessary information, and checking the login credentials entered by the user, the login system can be made more secure and user-friendly. This can be achieved by using functions, classes, and objects that encapsulate the logic of connecting to the database and authenticating the user, thus abstracting the complexity and providing a simple interface to interact with the SQL database. Additionally, abstraction can be used to prevent SQL injection by using parameterized queries, which abstracts the SQL code and ensures that the input data is properly sanitized.

Decomposition is also a critical concept that is well-suited for creating a spider graph and a line graph. By breaking down the problem into smaller, manageable tasks, such as creating a new canvas, drawing lines, and plotting data, the program can be made more efficient and easier to understand. This can be achieved by dividing the program into different functions, each responsible for a specific task, such as creating the canvas, plotting the data, and displaying the graph. This allows for a clear and visual representation of the data, making it easier for users to understand and interpret.

In conclusion, the problem of creating a login system, a spider graph, a line graph, and a question interface is well-suited for being solved using code because of the concepts of abstraction and decomposition. Through the use of code, the problem can be automated and made more efficient, secure, and user-friendly by using functions to encapsulate the logic, dividing the problem into smaller manageable tasks, and ensuring input validation and sanitization.

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Output |
| Login system | The credentials are checked against the database | Error popup/main menu page |
| Menu buttons | Tkinter will open a page with the matching screen | The page will be shown |
| Test completed | The code will create the outputs | The graphs will be output |

Database layout

Users

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Password | Email | Admin |
| String | String | String | Boolean |
| Eg, YSanogo | 7he.GoaT! | YSanogo@gmail.com | 0 (not admin) |

Persons

|  |  |  |  |
| --- | --- | --- | --- |
| Idnum | First name(s) | Surname | DOB |
| Integer | String | String | YYYY-MM-DD |
| 0001 | Matt | Ryan | 1992-04-08 |

Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tested | Idnum | Test\_Array | Test\_Num | Total | outcome |
| Integer | Integer | An array of 5 data points | Integer | An integer from 0 to 50 | One of three outcomes |
| 0002 | 0001 | [1,1,2,3,4] | 0001 | Eg, 45 | Eg foster care, care home, or supervised living home |

Questions

|  |  |  |
| --- | --- | --- |
| id | questions | Answers |
| Integer | String | String |
| 001 | 1: do you lie often | 0;1;2 |

Validation

For the code to run correctly, all inputs must be validated to ensure that inputs are correct. How I can validate this will be detailed below:

Buttons

As these are prewritten functions created by tkinter, they are known to be reliable and bug-free, meaning that any data inputted via the buttons can be trusted without needing validation.

Checkboxes

As mentioned above, they are created by tkinter, meaning that the checkbox can be trusted to give the correct output provided I have implemented them correctly.

Radio buttons

I will be using these in the evaluation portion of the code. They are a better choice than allowing any user entry as they are clear and user-friendly, improving my accessibility of the code. As the users are only given a set list of options they can choose, incorrect values can’t enter the system and cause errors. They also look better and are more space efficient. As they are also a premade function in tkinter, their implementation should be relatively simple compared to a text box.

Text box

Username

I will need to check that the username the user is trying to claim is unique and not claimed by another account.

Password

I can check that the user has passed specific requirements for the password to be accepted, such as its length, if it contains an upper-case letter, etc. This will improve the system’s overall security as if one password isn’t secure it could jeopardise the whole database.

Email

I will check if the user has entered a valid email by sending a verification email and getting the user to enter a code sent to the email the user gave.

Searching/adding a user

Testing

As I build this program, I’ll take a step-by-step approach to test each function as it's created. This way, I can ensure everything works properly before moving on to the next step. It's important to me that the program is functioning correctly and that any bugs or areas for improvement are identified early on. I'll be testing functions such as data input and output, making sure that the results are what I expect them to be. I'll also be keeping an eye on the program's console output, so I can troubleshoot any crashes or errors that may occur. And if I need to, I'll print out variables and examine their contents to help me understand what's happening. Once development is finished, I'll perform a final testing round. This will include destructive testing - intentionally entering incorrect data or trying to break the program in other ways. I'll also be testing user text input, ensuring that the program can handle various inputs. Finally, I'll show the program to stakeholders for their feedback. This is crucial to ensure that the program meets their needs and that any additional features or changes they'd like to see are considered. Throughout this process, I'll be using an iterative development approach. This means that I'll be breaking the program into smaller, more manageable chunks and then working on each one. By testing each piece as I go, I can ensure that everything works as it should and that the final product is functional.

|  |  |
| --- | --- |
| Testing checklist | Yes/no |
| Login system |  |
| Help button |  |
| Add account validation |  |
| Check unique username |  |
| Email verification |  |
| Username is only alphanumeric check |  |
| Password min length |  |
| Historical data |  |
| Add person validation |  |
| Name only contains letters and hyphens |  |
| Date of birth is a real date in the past |  |
| Add to record |  |
| Check if ID given is a real ID |  |
| Completing evaluation |  |
| Only allows the user to move on to the next question with a valid answer |  |
| Line graph |  |
| Axes are drawn correctly with correct labels |  |
| Line is drawn correctly between each point on the graph |  |
| KNN |  |
| Distance for each point is correctly calculated |  |
| Each distance is inversed to give a weighted value |  |
| Each points outcome value is loaded and its given weighted is added to the score |  |
| The final correct result is outputted to the user |  |
| Nightingale chart |  |
| Axis drawn correctly with correct labels |  |
| Segments created with correct size depending on the score of the test |  |
| Overall correct graph |  |

Test tables

Login system

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 1.1 | If the system accepts correct logins | Allows the user to login in |
| 1.2 | If the system rejects incorrect logins | Sends an error message |

Help button

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 2.1 | Help button loading correctly | Help page loads |

Add account

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 3.1 | If the system rejects duplicate usernames | Error message when a user tries to use a pre-existing username |
| 3.2 | Checks if the username has only alphanumeric characters. | If the system rejects a username with special characters |
| 3.3 | Checks the password is of a min length | Error message is password is too short |
| 3.4 | Check that email is unique | Error message if the user tries to use a pre-existing email. |
| 3.5 | Email verification using a code | Only allows the user to make an account if they have the right code. |

Add person

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 4.1 | If the system only accepts names without special characters. | Shows an error message that tells the user to correct the name |
| 4.2 | The system rejects any dob with a date that hasn’t happened | Accepts/ rejects the user based on if the dob is a real date |

Append to record

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 5.1 | If the system accepts a correct id | Shows an error message that tells the user to correct the name |
| 5.2 | The system rejects an incorrect id | Accepts/ rejects the user based on if the dob is a real date |

Complete evaluation

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 6.1 | Only allows the user to move onto the next question if they have answered the previous question | Shows an error if the user tries to move onto the next question without providing an answer. |
| 6.2 | Back button | Loads the previous question |
| 6.3 | Next button | Loads the next question |

Line graph

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 7.1 | If the axis loads correctly | If the axis and labels are loaded correctly |
| 7.2 | If the lines are drawn correctly | If the lines are correctly drawn |

KNN

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 8.1 | Calculate the distance of each point to the new point being added | A list of the Euclidean distances from the new point |
| 8.2 | Inverse all the distances to get a weight | A list of the weights of the points |
| 8.3 | Every point’s outcome is loaded | A list of the outcomes |
| 8.4 | Outcomes are compared to the weights are added to the relevant list | An output of the three scores and the correct score is determined |
| 8.5 | Confidence score | Correct confidence score is obtained |

Nightingale Chart

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 9.1 | Axes are drawn correctly | System outputs the correct axis. |
| 9.2 | Segments are drawn correctly | A completely correct graph |

Historical data

|  |  |  |
| --- | --- | --- |
| Test id | Feature tested | Expected output |
| 10.1 | Incorrect IDs are rejected | An error message stating that the id doesn’t exist. |
| 10.2 | Valid ids output correct data | A list of the correct values. |

Development and testing

Login page

The first thing that I will code is the login system. The first stage will be a basic login system that only checks if the username and password match an entry in the database.

This is the code I created to do this:

Text

Description automatically generated

I will next create the basic UI for my code. This will be as simple as possible; any cosmetic adjustments will be easy to include later in the code.

Text

Description automatically generated

After adding the tkinter code, the output window looks like this. This also shows the privacy feature of the code in that it hides the password behind asterisks, making the whole system safer.

Graphical user interface, application

Description automatically generated

Based on feedback from stakeholders, I have made changes to the program to ensure usability for as many people as possible, especially those with dyslexia and other reading difficulties, such as visual impairment. I researched online and discovered that Verdana is a highly recommended font for such individuals, and I have incorporated it throughout the program.

Additionally, I noticed a considerable amount of space in the code, so I have included a logo called 'placeholder.png' to fill it out. These two changes have been implemented in the code below.

Text

Description automatically generated

To determine whether a user attempting to log in is an admin, I will create a check examining the user's account credentials. The presence or absence of an "admin" designation in the associated SQL database will indicate whether the user has administrative privileges and, therefore, access to certain system features on the subsequent menu screen. This check will be a relatively simple process involving examining a Boolean value in the SQL database and should be straightforward to implement.Text

Description automatically generated

After creating the required column in the database, the login system will check the admin value associated with each user account, and the outcome will be added to the system. To improve the organisation of the code and facilitate easier error detection and correction, I have decided to move the creation of the tkinter window out of the login system and into a dedicated subroutine. There are two primary reasons for this decision. First, it allows for better code organisation, making it easier to locate and fix any issues that may arise. Additionally, by separating this functionality into its own subroutine, the code can be reused for future use cases. I have created a generic subroutine called 'basescreen' that makes a primary screen in tkinter that can be modified for each specific use case. This modular approach is more memory-efficient and quicker to debug, as there are fewer lines of code to check.

Text

Description automatically generated

This is the basescreen subroutine as described above. This is the basis for every window in my code.

After completing this section of the code and the testing for it, I decided that the current system was clunky as the user had to click out of the welcome message, and the only reason I checked if the user was an admin or not was to check if my system was working as intending. Therefore, I changed the code a final time to give this resultant login system.

Text

Description automatically generated

This has the same logic as the solution above, but instead of showing a welcome message, it simply moves on to the menu screen, only showing an error if the username or password doesn’t match.

This produces the final output shown below.

Graphical user interface, application, Word

Description automatically generated

Note for this testing, ‘u1’ and ‘u2’ are the only accounts to exist, with passwords ‘p1’ and ‘p2’, respectively.

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 1.1 | 04/01/2023 | A correct username and password combination ‘u1’ and ‘p1’ is entered | Pass- the user is correctly allowed to enter the system |
| 1.2.1 | 04/01/2023 | An incorrect username and password, ‘u3’ and ‘p3’ is entered | Pass- the system correctly showed an error message and didn’t allow the user into the system |
| 1.2.2 | 04/01/2023 | A correct username but incorrect password, ‘u1’ and ‘p3’ is entered | Pass – the user wasn’t allowed to log in |
| 1.2.3 | 04/01/2023 | An incorrect username but correct password, ‘u2’ and ‘p3’ is entered | Pass – the user was not logged in |
| 1.2.4 | 04/01/2023 | A correct username and correct password was entered but not the same pair, ‘u1’ and ‘p2’ | Pass – the user was not logged in |

Now that the login system has been created, I can move to the code’s main menu. Although making the main menu primarily involves using tkinter, several considerations must be considered to ensure the menu is user-friendly and practical. This includes carefully planning the menu's functionality and attention to its design and aesthetics. This is the code I created.

Text

Description automatically generated

This code produces the following output.

Graphical user interface, application

Description automatically generated

Next, I will complete each button individually, starting with the complete evaluation button.

Complete evaluation

The first screen that will load will show two buttons, asking the user if they want to append the test they are about to complete to a current record or if they want to add a new one.

Text

Description automatically generated

This is the code to complete this. With the button the user clicks loading the next screen depending on their choice.

Graphical user interface, application

Description automatically generated

This is the outputted tkinter window.

Append to record.

If the choice append to record is clicked, then the user should be sent to a page that allows them to enter the user number of the person they are completing the evaluation for. This is a simple case of tkinter and SQL and data validation to see if the value entered matches up to a real person stored in the system.

The above of this system is quite like that for the previous parts and creates this output.Graphical user interface, application, Word

Description automatically generated

Next, I will take the value the user inputted and see if it matches a value stored in the database. This is the code that achieves this.

Text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 5.1 | 08/01/2023 | A correct id is entered | Pass – the correct item is loaded |
| 5.2 | 08/01/2023 | An incorrect id is entered | Pass- the code gives an error message |

Add new record

To add a new record, the user must give three pieces of information, first name, last name, and date of birth. I will first code the tkinter for this.

Text

Description automatically generated

This creates the following output.

Graphical user interface, application, Word

Description automatically generated

Next, I will start the validation for each of the inputs. This will be straightforward for the first name and surname, as it is just checking that the characters entered are those found in a name, e.g., letters, - and spaces. This can be achieved by using a dictionary to check that every value in the string that the user entered is contained within the allowed characters.

Text

Description automatically generated

This code takes a name as an input and outputs true if all the characters are in the set, and false if any character is not contained within the set.

Next, I will create the checker for the date of birth. This can be split into two parts. First, check if the date entered is in the correct form (YYYY-MM-DD), and then check if the date is real and has happened.

Text

Description automatically generated

This is the code that shows the description above. It will only allow the date to pass if it is in the correct form and if the date is a real one in the past.

All this validation together combines to create this code.

Text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 4.1.1 | 14/01/2023 | Leaving the first name box empty | Pass- an error message was output saying to correct the first name. |
| 4.1.2 | 14/01/2023 | Leaving the surname box empty | Pass- an error message output saying to correct surname. |
| 4.1.3 | 14/01/2023 | Entering a banned character in the first name box. | Pass – an error message output saying to correct |
| 4.1.4 | 14/01/2023 | Entering a banned character in the surname box. | Pass – an error message stating to correct surname. |
| 4.2.1 | 14/01/2023 | Leaving the date of birth box empty | Pass- an error message stating that the date given does not exist. |
| 4.2.2 | 14/01/2023 | Entering a date not in the correct form, 12292004 | Pass – an error message stating the date doesn’t exist. |
| 4.2.3 | 14/01/2023 | Entering a date that doesn’t exist, 30/02/2000 | Pass – an error message stating the date doesn’t exist. |
| 4.2.4 | 14/01/2023 | Entering a date in the future, 29/07/2025 | Pass – an error message stating that the date is in the future. |
| 4.2.5 | 14/01/2023 | Entering a current, real date,12/10/2006 | Pass – the system allowed the user to pass into the system |

Next, I will move on to the load question function.

Load questions

The loadquestion function will connect to the database and retrieve a list of questions from the questions table. Each question's text and answers will be stored in a dictionary and appended to the questions list.

Text

Description automatically generated

This code is then run on the following table in the database.

Table

Description automatically generated

The questions presented here have been gathered from various sources, such as a spreadsheet provided by a stakeholder and from conversations with other stakeholders. This code section is crucial, as low-quality questions would render the results ineffective. Consequently, it was vital to consult with professional stakeholders to ensure the questions’ quality.

I reached out to my stakeholders with the following email:

I hope this email finds you well. I have been working on a significant part of the code I am developing, and I would like you to review the list of questions I have attached below. Ensuring the highest quality of these questions is essential, as subpar inputs could render the overall performance of the code ineffective. As this aspect of the code is critical to the success of our project, it is vital to seek your professional opinion on the questions we plan to use. Your valuable input and suggestions for improvements or modifications would be greatly appreciated.

Thanks, Kyle.

My first stakeholder replied:

Thank you for reaching out and providing the opportunity to review the questions for this crucial part of the code. I appreciate your diligent approach to ensuring the success of the project.

After thoroughly reviewing the document you provided, I am pleased to inform you that the questions appear well-structured and comprehensive. In my opinion, the questions are well-suited for generating high-quality outputs, and I believe they will contribute to the effectiveness of the code.

As of now, I have no further suggestions or modifications to make. I am confident that the current set of questions will serve our project's needs well. Wishing you the best of luck with the project, and I look forward to seeing the results.

My second and third stakeholders were also satisfied with the questions I chose to use. The questions are designed to be asked to the child; if the worker disagrees, they can discuss before concluding their score. The questions are also grouped in sets of 5 in the BERRI order. This is done to make it easier to create the outputs later as they are already grouped.

As my stakeholders have approved these questions, I can move on to displaying and getting answers.

Display questions

First, I will get the code to display the first question.

Text

Description automatically generated

This outputs the first question to the tkinter window. Next, I will add the text box that allows the user to enter their response.

Text

Description automatically generated

This results in the following output.

Graphical user interface, application, Word

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 6.1.1 | 20/01/2023 | A correct value,1, is entered. | Pass- the value is accepted. |
| 6.1.2 | 20/01/2023 | An invalid answer is entered. | Fail – the code accepts the value. |
| 6.1.3 | 20/01/2023 | The text box is left empty | Pass – the code sends an error message. |

After completing the testing, the code doesn’t operate correctly, not allowing the correct values to pass. I have decided to change my input system to use radio buttons instead. The reason for this is two-fold: first, I will have to do less error checking, as the only values the user can enter the system are the ones I provide and the initial value. This will lead to better performance in the code, as well as no way for incorrect values to enter the system. Second, it means that the user can perform a test without using a keyboard, which could be helpful. I will now edit the code to include the radio buttons.

Text

Description automatically generated

The changes I have made include setting the initial value of the variable that the user changes to -1; this makes it easy to check whether the user has entered a value. R4 is also present as it seemed to solve a cosmetic problem where the 0-button appeared differently from the other 2 two buttons, shown below.

Without r4:

A picture containing text

Description automatically generated

With the r4 line included:

A picture containing table

Description automatically generated

As this line has no impact on the code’s functionality and does improve the appearance of the code, although it appears to do nothing, it has a use, and I will therefore include it.

Finally, before I begin testing my solution, I need to add the next and back buttons to my code to reach past the first question. I will need to ensure that the buttons are only visible at the correct times, e.g., the back button shouldn’t be present on the first question as the user has nowhere to go back to. With this all in mind, I will start the code for the buttons.

Text

Description automatically generated

This is the code for the next button. When the button is pressed, it checks the value of what the user entered; if -1, the user entered nothing. If the user has answered the question, the value of the answer is appended to the list of the answers, and the next question is loaded. If no response is provided, the code will output an error message telling the user to enter an answer.

Text

Description automatically generated

This is the code for the back button; it only loads after the first question for the reason described above. Then if the user presses the button, the previous answer is removed to make way for the new answer that the user chooses. Then the previous question is loaded for the user to answer again.

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 6.1.4 | 23/01/2023 | A radio button is selected. | Pass – the code loads the next question and appends the answer to the list. |
| 6.1.5 | 23/01/2023 | No option is selected. | Pass - an error message is shown. |
| 6.2.1 | 23/01/2023 | Pressing the next button before an answer has been selected. | Pass – an error message is output asking the user to enter an answer. |
| 6.2.2 | 23/01/2023 | Pressing the next button after the answer was selected. | Pass – the next question is loaded |
| 6.2.3 | 23/01/2023 | Pressing the previous button. | Pass – the previous question is loaded and the last answer is removed from the list |

Now that the question’s function has been completed, I can move on to the outputs produced by the code.

The first thing that needs to be done is to collect the data in a form that can be output in a meaningful way. I need to collect the data in 2 different formats.

First the value of the five different parameters (behaviour, emotional well-being, risk, relationships, indicators), each of which has five questions for a value of 0-10; this will be used for both the KNN algorithm and the Nightingale Chart. Second, I will add all the values formed in the first part and add them all together. This is needed for the line graph.

Text

Description automatically generated

This function takes the answers to the questions and groups them into the relevant group, then finds the total test score.

With the data I need to complete the outputs, I will begin creating them, starting with the KNN algorithm.

First, I need to load every point that has been completed into the system.

Text

Description automatically generated

As this loads the data in a form that I don’t want, e.g.:



When I need a 2d array, e.g.

[[10,5,0,5,10],[10,10,10,10,10] etc



This line strips the code of the round brackets and gives the output.



Finally, to remove the speech marks, I used the JSON function to remove them as shown below.



This gives the final output of the 2d array I was looking for.



Next, I need to find the Euclidean or shortest distance, between every point and the new point being added. As shown in the design section, I can use a modified version of the Pythagorean theorem to calculate this distance, so this shouldn’t be too hard to code with the math function.

Text

Description automatically generated

First, the total and weight list are created; these will be needed later to store the value of the weights calculated. Then the code enters 2 for loops; this is done so that each value within each array is correctly given the right weight, and the difference between the points is calculated on each plane. This value is then squared and added to the total, this is then repeated until each plane is calculated. Then the code takes the square root of the sum to give the distance; the total is then reset so the next point doesn’t have the last point’s values included, making the calculations wrong.

Next, I need to work out what to do with the distance value that comes from the code. There are two possible outcomes, either the distance is zero or not. If the distance is 0, then the rest of the code can be skipped. This is because the correct outcome has already been found. If the length is non-zero, the code will take the reciprocal and append that to the weight list.

Text

Description automatically generated

Now the code should get a list of values from 0 to 1. This represents how much impact each point will have on the overall total.

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 8.1.1 | 30/01/2023 | A new and unique point is compared to all previous points in the database. | Pass – a correct list of all the distances from the previously added points to the new one. |
| 8.1.2 | 30/01/2023 | A duplicate point is added to the system. | Fail – the code finds the correct outcome but continues to run the rest of the code and then runs out of data, causing a runtime error. |
| 8.2 | 30/01/2023 | A unique point is added to the system. | A correct weightlist is the output |

With every point’s weight, I need to load each point’s corresponding outcomes and add all the points with the same outcome weights together. After adding these values, the outcome with the most significant total weight will be output and stored in the SQL database to improve the system’s accuracy over time.

Text

Description automatically generated

This code loads all the previous outcomes and converts the output into a form the system can understand. Then each point’s outcome is sorted into three options, and the corresponding weight is added to that outcome’s total weight. Finally, after all weights are added, there is a check to see which outcome is the highest value, with any equal values decided by an arbitrary choice.

The final part of the code will be outputting the outcome of the KNN algorithm to the user; I have also decided to include a confidence score on how sure the algorithm is that the outcome it chose is correct. To calculate the score, I will add the total weight together and use the formula conscore = winning weight/total weight. This will give a value from 0 to 1, the higher the more confident the system is in its prediction.

Text

Description automatically generated

This code should correctly output both the top outcome and the system’s confidence in it being the correct outcome.

Now that the rest of the code has been created, I can try to fix the issue that arose when the distance is equal to zero. After some thought, I think that I have decided on a good temporary solution, and that is to make the distance between the two points a very small number. This will mean that the weight of this point will be high and will, therefore, overpower the rest of the points.

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 8.1.3 | 05/2023 | A duplicate point is again entered into the system. | Pass – the correct outcome is produced with a high confidence score. |
| 8.3 | 05/02/2023 | N/A | Pass - The correct list of outcomes is loaded to be used later. |
| 8.4 | 05/02/2023 | N/A | Pass - Each point is successfully sorted into the relevant outcomes. |
| 8.5 | 05/02/2023 | N/A | Pass - The correct confidence score is found, and all data is correctly output to the user. |

Now that the KNN algorithm works effectively, I can move on to the line graph and Nightingale Chart.

As the KNN algorithm works, all the data that the test needs to be added to the database has been created; this means that the new test can be inserted into the table and makes the line graph much easier to code, as all the data points I need are now stored in one place.

To start creating the line graph, the first thing that needs to be completed is the axes for which the chart will lay. I want the y-axis to have dotted lines coming off it in order to make the graph easier to read.

However, before I can make the axes, I need to import the data from the database. This is important because the x-axis, which represents the number of the test, will need to change its scale based on how many tests have been completed.

Text

Description automatically generated

This code finds all the total values stored in the database, including the test that has just been completed and converts them into a list that the rest of the code can read and use.

Now with this ‘list’ created, I can make the axis for the graph.

Text

Description automatically generated

This is the first part of the axes creating the two lines off which the rest of the graph will be built.

Graphical user interface

Description automatically generated with medium confidence

Next, I will add the y-axis labels with the dotted line included.

A screenshot of a computer

Description automatically generated with medium confidence

This is the solution I found after some trial and error, with the values to use in reference to the yscale to use etc. It takes the total height of the graph and divides it by the highest possible value that the test can output. It then enters a loop, first finding the relevant y value to place the labels and lines. It then puts the I value, which is every fifth number, on the graph. Finally, to draw the bar across the chart, it takes the calculated y value and draws the line across the graph at that height.

Table

Description automatically generated

This is the output given by this code. I like the dotted line. However, I don’t like the fact that the code doesn’t create the final line, ideally, there would be a final line at the top of the y-axis with 50 next to it.

After some research, I believe the issue is that the code is not iterating the final time it needs to in order to create the last line. This is a simple fix that requires the final value to be increased, as shown below.



This change gives the corrected output.

Table

Description automatically generated

Next, I will add the labels on the x and y axis (score on the x and test number on the y).



Table

Description automatically generated

The final element of the axes is the correct labels on the x-axis. This will involve finding the number of elements of flist to find the accurate scale and the correct number of items to put on the bottom.

Text

Description automatically generated

Table

Description automatically generated

This is the final part of the axis completed: note that I am using an example list for this testing with values [50,30,0,27]. This is to test both the higher and lower extremes of the graph and some values in the middle.

Now the final part of the graph is the actual creation and joining of the points, and this can be done by calculating the current point and the next point to be plotted and simply joining them.

A screenshot of a computer

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 7.1 | 15/03/2023 | N/A | Pass – the axis are correctly formed and labelled. |
| 7.2.1 | 15/03/2023 | A maximum test value is added | Pass – the line is correctly placed at 50. |
| 7.2.2 | 15/03/2023 | A minimum test value is added | Pass – the line is correctly placed at 0. |
| 7.2.3 | 15/03/2023 | A value of 25 is added | Pass – the line falls at the correct place. |

Now with the line graph completed, I can move into the final element of the outputs, the Nightingale Chart. As described above, the first element of the graph that needs to be created is the axes on which the data will lie. However, as the Nightingale Chart has no traditional x and y axes, it will be harder to create the axis. The first part of the axes to develop is the dotted concentric circles to help the user read the graph’s values.

To do this, I first need to construct my class and the class variables I will need to create the graph.

Text

Description automatically generated

This code takes flist (the list of the five different totals found by the test just completed). The width and height of the canvas on which the graph will be drawn and works out the x and y values of the centre of the canvas and the chart.

Next, I need to create the actual circles for the graph to rest on. This shouldn’t be too difficult, just creating six different circles.

Text

Description automatically generated

This code draws six rings going out from the centre, It works out the position of the dot using the properties of a right-angle triangle and some basic trigonometry to convert the angle and direction of the line into the cartesian form, which Python can then process and output the plot I am looking for. This can be best understood using the diagram below.

Diagram

Description automatically generated

So, imagine that point one is the centre of the circle and point 2 is the edge, this then gives the formula to get from the origin to the new point:

X = centrex+ (Rcosn)

Y = centrey+ (Rsinn)

I then add one to both values to have a line that the user can see, creating the following.

A picture containing text, white

Description automatically generated

This is an excellent basis to work on the rest of the graph, as the lines are clear enough to be seen but easy to look past when the user needs to know where the actual segments are. Next, I will add the labels for each ring, which will have a 0,2,4,6,8,10 at each ring every 72 degrees, so when the graph is completed, it will be between each segment.

Text

Description automatically generated

This is the code I created. It is a nested loop in the draw axes function and uses similar logic as shown above but with a ‘multiplier’ in the (i/5) to place the values correctly and results in the following output, note that self.size = 5.

Chart, radar chart

Description automatically generated

Now with the axes complete, the final two components of the graph are the BERRI labels and the segments representing the data collected.

To correctly draw the segments, I need to accurately find the segment's radius and the angles at which it should be visible.

The radius is quite simple to find. It is simply the value of each of the elements in the array divided by the max value and then timed by the max radius.

The starting and ending angle is also quite simple: 72\*the sector number and 72\*(sector number+1)

I next added two more class variables, a list of colours that the segments will be shown as. This is to make it easier for the user to read the graph. I also added the BERRI labels, this is to make it clear what segment refers to which area.

Text

Description automatically generated

Text

Description automatically generated

This code finds the outline of the segments to be drawn. It goes through every angle from the starting and ending angle found before and calculates the positional coordinates at each angle that falls on the edge of the segment. This is then added to the list of points, and finally, the polygon is created.

Chart

Description automatically generated

As this is all correct, I can move on to the final element of the graph: the labels. This only requires finding each segment's average angle, calculating those points’ x and y values, and finally placing the label at that point.

Text

Description automatically generated

Chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 9.1 | 27/02/2023 | N/A | Pass – the axes are completely correct and correctly labeled. |
| 8.2 | 27/02/2023 | A test with outcome [0,2,4,8,10] is entered in order to test all the extremes of the graph. | Pass – the graph is correctly displayed to the user. |

Now that the complete evaluation button is finished, I can move on to the other options on the main screen: the ‘help’, ‘view historical data’ and ‘add account’ buttons. I have decided to start with the ‘add account’ button, as this should be very similar to the ‘add new record’ button created before.

Add account

I will first need to create a page with three text boxes for email, username and password, and a tick box to determine if the new user is an admin.

Text

Description automatically generated

This is the code for the add account function. Note that the password will be shown on this page because the password cannot be changed later, so it is crucial that the user picks the correct password they want.

Next, I need to check that the user entered a unique username, which is also alphanumeric. I also need to check that their email is unique.

Text

Description automatically generated

Finally, I can move on to verifying the user’s email. This will send a code to the email that the user puts into the sign-up page and only allows the user to create their account when they have entered the correct code. This can be done most efficiently using the smtplib library installed in Python.

Text

Description automatically generated

Finally, I will run a check in the verify function, and if the code is correct, the email, username and password will be inserted into the SQL database.

Text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 3.1.1 | 15/03/2023 | A unique username is entered. | Pass – the user is allowed to create an account |
| 3.1.2 | 15/03/2023 | A username that is already in use is used | Pass – the code outputs the correct error massage. |
| 3.1.3 | 15/03/2023 | The username field is left empty | Partial pass – an error message is shown, however, it states that the username contains a non-alphanumeric character. |
| 3.2.1 | 15/03/2023 | A username containing only alphanumeric characters is entered | Pass – the user is able to create the account |
| 3.2.2 | 15/03/2023 | A username containing a non-alphanumeric character is entered | Pass – the appropriate error message is shown |
| 3.3.1 | 15/03/2023 | A password of below 8 characters is entered | Pass – the system sends an error message saying to increase the length |
| 3.3.2 | 15/03/2023 | A password greater than or equal to 8 characters is entered | Pass – the system allows the user to create their account. |
| 3.4.1 | 15/03/2023 | A unique email is entered | Pass – the user is allowed to continue |
| 3.4.2 | 15/03/2023 | An email already in use is used | Pass – the user is shown an error message |
| 3.4.3 | 15/03/2023 | Email text box is left blank | Pass – the user is shown an error message |
| 3.5.1 | 15/03/2023 | The user inputs the correct code sent to their email | Pass – the account is created and details are correctly stored |
| 3.5.2 | 15/03/2023 | The user enters an incorrect code | Pass – an error message is sent |
| 3.5.3 | 15/03/2023 | The user leaves the enter code text box blank | Pass – the user is shown and error message. |

Now that the ‘add account’ button is completed and correctly functional, I can move on to the final two elements of the code.

Help button

As I have already written the tkinter code for the button, it is just a case of writing a basic description of how the code works and how to use it effectively. This is not worth covering as it has all been detailed at great length in the previous pages.

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 2.1 | 17/03/2023 | Help button clicked | Pass – help button loaded correctly |

Historical data

The final part of my coded element is the view ‘historical data’ button. This is quite a simple part of the code as it just involves loading all previous entries from the SQL database and showing them on a tkinter window.

Text

Description automatically generated

This code asks the user to enter the id of the person whose results they want to find out, then the code loads a screen with all previous results laid out, as shown below.

Text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Test id | Date | Input | outcome |
| 10.1.1 | 22/03/2023 | An incorrect id is entered | Pass – an error message was shown |
| 10.1.2 | 22/03/2023 . | The input box was left blank | Partial pass – an error message saying ‘this id has no record’ is shown despite no id being given. |
| 10.2 | 22/03/2023 | A correct id is entered | Pass – the correct test scores are loaded. |

Checklist

|  |  |
| --- | --- |
| Testing checklist | Yes/no |
| Login system | Y |
| Help button | Y |
| Add account validation | Y |
| Check unique username | Y |
| Email verification | Y |
| Username is only alphanumeric check | Y |
| Password min length | Y |
| Historical data | Y |
| Add person validation | Y |
| Name only contains letters and hyphens | Y |
| Date of birth is a real date in the past | Y |
| Add to record | Y |
| Check if id given is a real id | Y |
| Completing evaluation | Y |
| Only allows the user to move on to the next question with a valid answer | Y |
| Line graph | Y |
| Axis are drawn correctly with correct labels | Y |
| Line is drawn correctly between each point on the graph | Y |
| Knn | Y |
| Distance for each point is correctly calculated | Y |
| Each distance is inversed to give a weighted value | Y |
| Each points outcome value is loaded and its given weighted is added to the score | Y |
| The final correct result is outputted to the user | Y |
| Nightingale Chart | Y |
| axis drawn correctly with correct labels | Y |
| Segments created with correct size depending on the score of the test | Y |
| Overall correct graph | Y |

Final testing

As all my destructive testing has been completed, I believe that the best way to test my code now is through ordinary testing of the whole system together. In order to show this testing, I have created a series of videos which can be found here: <https://www.youtube.com/@CESSSOFTWARE/featured>. This is a list of videos which shows all my different functions working together cohesively to create my overall code.

Through both my testing while developing the features and while testing the code, I now believe that my code is as bug-free as possible, and I will now pass it off to my stakeholders from them to test it.

I asked my stakeholders to try to test as much of the code as they could and asked the:

Was the code easy to use on first use? And if not, why?

Did the help menu sufficiently help you use the program?

Did the code give accurate recommendations?

Overall, is this code good enough for you to use in the future?

To these questions, I got the following responses:

Was the code easy to use on the first use? And if not, why?

Stakeholder 1: ‘Yes, I found the code to be clear and quite easy to understand.’

Stakeholder 2: ‘While I found the code to be mostly easy to use, I thought the add new record button within the complete evaluation button to be quite confusing as it took me a while to find.’

Stakeholder 3: ‘When first looking at the code, I found the load historical data button to be a bit redundant, in my opinion, and make the menu screen look cramped.’

Did the help menu sufficiently help you use the program?

Stakeholder 1: ‘As I found the code quite easy to understand, I did not need to use the help button much, although I can still see its use.’

Stakeholder 2: ‘In my opinion, the help button could have been improved by adding images to make it clear to the person using the program what screen you are referring to, for example.’

Stakeholder 3: ‘I thought the help button was quite useful as it helped me understand what the load historical data button does, and I can now see its use.’

Did the code give accurate recommendations?

Stakeholder 1: ‘Yes, I thought the code was quite accurate, and I often agreed with its opinions.’

Stakeholder 2: ‘The recommendations I found to be pretty correct, and I liked the confidence score, as it helped to give more context to the answer than just what it said on the screen.’

Stakeholder 3: ‘I think the recommendations could be improved upon; however, I understand that the code will continue to learn and hopefully start to improve on its answers in the future.’

Overall, is this code good enough for you to use in the future?

Stakeholder 1: ‘Yes, I believe this is good enough for future use.’

Stakeholder 2: ‘With some changes to the help button, I believe this program could be very beneficial to future work.’

Stakeholder 3: ‘After some time and changes to the code’s recommendations, I think this code may be useful in the future.’

Analysis:

Overall, my stakeholders seem to be happy with how my code turned out. After some use, they liked my interface as it was quite simple to read and understand for most people. I also think that they liked the graph outputs that I created.

The main complaints were in the help button and the Knn algorithm. The help button is quite bland and could be hard to understand if the user hasn’t used programs like this one before, so this is something that could be easily improved upon in the future. The Knn algorithm would be harder to improve, and I believe that as time goes on and the algorithm becomes more refined, its outcomes will only become more accurate.

Evaluation

|  |  |  |
| --- | --- | --- |
| Criteria | How to evidence | Met? |
| Login page with username and password | Screenshot of the window that shows the GUI with the login page | yes |
| Menu page with different options | A screenshot shows a simple menu with multiple buttons, such as creating a new record or adding to an old record. | Yes |
| A simple test screen | Screenshot that shows the questionnaire being completed, with an input that only allows integer inputs, with a back button option to correct any errors entered | Yes |
| A simple help menu is available throughout | Screenshot showing a help guide, with headers and screenshots guiding the user through the process of completing a test. | Partial |
| A lightweight design | Screenshot showing the non-cluttered and easy-to-understand menu and test screen | Yes |
| The option to search the database for previous entries to append to | screenshot showing the option to append to a previous data entry. | Yes |
| Set of graphs output | Screenshot showing completed graphs from a test. | Yes |
| Pieces of advice on what to do | Screenshot with pieces of advice, depending on the user’s score. | Yes |

Evidence

Login page

Graphical user interface, application, Word

Description automatically generated

Overall, I am happy with how the login page turned out. It is a basic screen but shows all the functionality it needs and is quite easy to read for those who may have vision impairment due to the high contrast of the background and text, as well as the use of a font that is easier for those with dyslexia to read.

Menu page

Graphical user interface, application

Description automatically generated

This is the main menu page, I think that it looks exactly how I thought it should look when I first started, not overly complicated but still has all the options I hoped it would.

A simple test screen

Graphical user interface, application, Word

Description automatically generated Graphical user interface, application, Word

Description automatically generated

These two screenshots show everything I wanted to achieve with the test screen: it only allows the correct inputs to be entered and has the next and back buttons working correctly.

Help page

A screenshot of a computer

Description automatically generated with medium confidence

I decided not to include any images in my final help guide as I felt they weren’t going to be of much use and opted for the text-based approach instead. I feel this is still just as valuable to the user as if I included images.

A lightweight design

Graphical user interface, application

Description automatically generatedGraphical user interface, application, Word

Description automatically generated

I think that this is as simple of a design as possible. This is important as it will allow people who may be less experienced with software similar to this one to still use it effectively and get the most out of the code.

The option to search the database for previous entries to append to.

Graphical user interface, application, Word

Description automatically generated

Set of graphs output

Pieces of advice on what to do

Graphical user interface, text

Description automatically generated

All of my features have now been tested thoroughly multiple times, both independently and while working together with the other functions in the system. This this means that I am quite sure that my code is as bug-free as it will get. I have also been sure to check the Gui for any text errors in the code, e.g., spelling and grammar, so I believe the code to be completely correct. Every text box has been given a series of correct and incorrect inputs to test it. To my knowledge, all the validation has worked perfectly in not allowing an invalid piece of data to enter the system. As well as this, all the buttons in the code have been tested at various points to see that they operate correctly and only allow the code to continue if the correct prerequisites are provided.

Limitations

The biggest limitation of the code is its current menu page and the Knn algorithm not being fine-tuned. This is due to the lack of clarity provided by the menu page, as I was told by my stakeholders. I believe this issue can be avoided by giving some basic training to those using the code. This would be quite easy to do as the code is not too difficult to explain when the code is being used in a real-world application.

The other issue is the Knn algorithm. This this is due chiefly to the current lack of data from which the algorithm can teach itself. This just means that the user must be careful with what the code outputs. However, this should not be a problem, as only those who are trained social workers will have access to the code and should therefore be able to make their own judgements on the person’s outcome not just relying on the currently young and inexperienced Knn algorithm. However, even when the algorithm has enough data to work with and give more accurate results, it should be considered a complementary tool to those trained in the profession and not replace any of their skills.

Overall, the impacts of the limitations on my code will be very manageable to work around and will not impact the efficiency or usability of my code much at all.

Maintenance

In the future, I would like to change a few things about the code. I think that the biggest issue, other than the ones mentioned above, is the fact that the code only accepts 0,1,2 in the test phase of the program. This could mean that some detail and nuance is lost that could be valuable to the outcome chosen. This would require quite a large overhaul of the system and is impractical to add now.

If, in the future, the stakeholders tell me they would like to see something added or changed, it would be quite easy to do, as my design is modular and would allow for additional outputs for example to be added or edited without impacting the rest of the code. It would also be very simple for the stakeholders to edit any of the questions they don’t like in the future, just requiring them to change them in the SQL database.

Future versions of this software could include a longer and more detailed evaluation of the person and the ability to have the outputs of the code emailed to the user as opposed to them just being on the tkinter window.

Final code

MAIN.py

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

‘Project. dB’

‘Persons table’

Table

Description automatically generated

‘Questions table’

Table

Description automatically generated with medium confidence

‘Tests table’

Table

Description automatically generated

‘Users table’

Table

Description automatically generated

References

BERRI - About BERRI. <https://berri.org.uk/about-berri>

<https://planetradio.co.uk/metro/local/news/children-services-cut-by-more-than-a-third-since-2010/>

<https://www.bbc.co.uk/news/uk-england-lancashire-58512743>