



Computer Science
A Level NEA

Henna Design Application



Thaker, Preya

Computer Science A Level NEA

Contents

| | |
|---|----|
| Analysis | 5 |
| What is the Problem? | 5 |
| Why is it suited to a computational solution? | 5 |
| What are the characteristics and needs of the stakeholders? | 7 |
| Initial Conceptions | 7 |
| Google Form Survey..... | 7 |
| How have my assumptions about my stakeholder characteristics changed? | 17 |
| Selected Stakeholders..... | 20 |
| What solutions already exist to similar problems?..... | 22 |
| In person teaching | 23 |
| YouTube tutorials..... | 25 |
| Applications/ Pinterest/ Google Images | 28 |
| What are the main features of the project? | 29 |
| 1. Henna Design Creator | 29 |
| 2. Recommendations | 29 |
| 3. Linked to YouTube videos | 30 |
| Other features my stakeholder wanted (from the survey):..... | 30 |
| What are the limitations of the Project? | 31 |
| What hardware and software are required? | 32 |
| Summary of requirements..... | 33 |
| How can the project's success be measured? | 35 |
| Success Criteria | 35 |
| Design | 37 |
| Decomposition..... | 37 |
| Decomposition Diagram | 37 |
| Explanation for decomposition diagram..... | 37 |
| Decomposition in greater detail | 38 |
| User Interface | 40 |
| Screen 1 – Loading Screen | 40 |
| Screen 2 – Homepage | 42 |
| Screen 3 – Settings..... | 44 |
| Links to Success Criteria..... | 45 |

| | |
|--------------------------------------|-----|
| Accessibility..... | 46 |
| Algorithms..... | 47 |
| Generating the Design | 47 |
| If Draw function | 57 |
| Henna Tutorials Via YouTube..... | 64 |
| Data Structures | 66 |
| Linked List | 66 |
| Database | 69 |
| 2D array..... | 71 |
| Data Types | 72 |
| User Defined Record Types..... | 72 |
| Key Variables..... | 74 |
| Inputs and outputs..... | 76 |
| Table | 76 |
| Validation..... | 77 |
| Testing..... | 78 |
| Development and Testing Method | 78 |
| Collecting Evidence | 80 |
| Functionality Checklist | 81 |
| Development and Testing..... | 82 |
| Stage 1- Loading Screen..... | 82 |
| Setting up the User Interface..... | 82 |
| Adding Interactivity..... | 85 |
| Stage 1 Review | 86 |
| Stage 2 - Homepage Interface..... | 89 |
| Setting up the components..... | 89 |
| Like and dislike buttons | 90 |
| New Design and Save Buttons | 94 |
| Henna Canvas | 95 |
| Stage 2 Review | 97 |
| Stage 3 - Elements | 99 |
| Plan | 99 |
| Elements | 99 |
| Stage 3 Review | 113 |

| | |
|---|-----|
| Stage 4 – Basic Templates..... | 114 |
| Plan | 114 |
| Template 1 | 114 |
| Template 2 | 126 |
| Stage 4 Review | 127 |
| Stage 5 – Linked List..... | 128 |
| Creating and Testing the Linked List | 128 |
| Stage 5 Review | 134 |
| Stage 6 - Database Setup | 135 |
| Creating the file | 135 |
| Populating the Tables..... | 135 |
| Stage 6 Review | 137 |
| Stage 7 – Displaying the Elements | 138 |
| Labels | 138 |
| Testing Label Population :..... | 141 |
| Testing Label Population Fix : | 142 |
| Stage 7 Review | 144 |
| Stage 8 - Linking database with code..... | 145 |
| Stage 8 Review | 149 |
| Stage 9 - Youtube API..... | 150 |
| Stage 9 Review | 157 |
| Stage 10 - saving canvas | 158 |
| User Testing: | 158 |
| Modified Code | 159 |
| User Testing 2: | 162 |
| Stage 10 Review | 163 |
| Stage 11 - Liking and disliking designs | 164 |
| Stage 11 Review | 171 |
| Stage 12 - New Design | 173 |
| Stage 12 Review | 177 |
| Stage 13 - Tidying up the code..... | 178 |
| Stage 13 Review | 179 |
| Stage 14 – Randomisation and Complex Templates..... | 180 |
| Finger block..... | 180 |

| | |
|--|-----|
| Stroke Randomisation..... | 183 |
| Adding finger to template 1..... | 186 |
| Changing filled circle | 190 |
| Adding Template 3..... | 191 |
| Stage 14 Review..... | 200 |
| Stage 15 – Adding More menu Bar Options | 202 |
| Application Manual..... | 202 |
| Henna Application Instructions..... | 203 |
| Stage 15 Review..... | 205 |
| Stage 16 – Final Testing | 207 |
| Functionality Checklist | 210 |
| Stage 17 – Final Stakeholder Feedback..... | 214 |
| Questions | 214 |
| Responses | 214 |
| Evaluation | 216 |
| Success Criteria Met | 216 |
| Unmet Criteria or Tests..... | 224 |
| Success Criteria | 224 |
| Functionality Testing..... | 224 |
| Module Tests | 224 |
| Usability | 225 |
| Limitations | 228 |
| Maintenance | 229 |
| Additional Changes | 229 |
| How is the application maintainable in the long term?..... | 229 |
| Final Forms and interfaces..... | 230 |
| Load Screen..... | 230 |
| Homepage..... | 231 |
| Final code | 232 |
| Bibliography..... | 260 |

Analysis

What is the Problem?

"Henna is a paste that is culturally associated with positive spirits and good luck" ¹ in many South Asian, Middle Eastern and North African cultures. Its primary use is for decorative body art, containing symbols and patterns which represent different blessings. For example, traditional Hindu weddings have a "Mehndi Ki Raat", which translates to "The Night of Henna". In this event an intricate henna design containing an image of a married couple and her husband's name is applied to the bride. This event is used to wish the bride good luck, health, and prosperity in her married life.

It is an oral art form, where the knowledge is passed down through generations. Even in our modern, technologically advanced world there is no easy way to learn henna online, if it is not taught in the family most people need to find a tutor. However, this has high monetary costs associated with it, and is risky given the current pandemic. The only methods which currently exist to self-teach henna are from online pictures, which often contain a lot of background noise, or books, which are not easily accessible and often expensive.

The solution I intend to code is a henna application which will help people to learn this artform. It will use different templates to create unique henna designs. The program will also learn from the user and adapt the designs according to the user's preferences. Additionally, it will link each element to a YouTube tutorial on how to draw it, hence aiding the user in learning.

Why is it suited to a computational solution?

This solution tends to computational methods as by its very nature as henna is composed of simple repeated elements. These are combined in different ways to form intricate, aesthetic designs. Computers are very good at this due to their ability to perform simple repeated actions (iteration) in a precise and effective manner, which is not possible to replicate by hand.

A computer would allow the necessary data for machine learning to be collected and analysed to customise designs to each user's preference. This algorithmic learning requires the processing capacity of a computer to achieve meaningful results.

Using API's I can automatically link the elements to YouTube tutorials, allowing the users to effortlessly find them.

Computers are also widely available, with most people having an individual or family device (especially in the first world and parts of developing countries), hence it would allow me to make henna learning easily accessible.

The application can also be easily updated with newer henna designs and trends, the only cost being maintaining and updating the software. There is no cost for printing physical goods, such as with books and magazines.

Problem Recognition:

The overall problem is generating henna designs in a format which aids the user in learning henna.

Problem Decomposition

The problem can be decomposed into smaller parts. Here is the initial idea of these steps:

1. Create a henna design
 2. Teach the user how to do specific parts of that design
 3. Ensure the design is one the user would like
-

Abstraction

Abstraction is essential to the success of this program. There are many parts of creating real life henna designs which can be ignored and simplified in this application, as its primary purpose is to teach people the basics of henna. For example the texture and size of the hand many make a difference. We can ignore these as they do not influence the generation of training patterns.

What are the characteristics and needs of the stakeholders?

Initial Conceptions

Due to its cultural importance (Section 1.1 for more information), I believe the application's main stakeholders will be of South Asian, Middle Eastern and North African ethnic origin. Traditionally in these cultures, henna is applied to females by females. Yet in our modern society with male beauty influencers such as James Charles and Lewy Ball, the beauty standards are being remoulded, hence I believe that gender is not an important stakeholder quality.

One of the main characteristics I believe my stakeholders will all share is a low-level artistic ability. Once you attain a high level of expertise in henna, many artists prefer to create their own unique designs and may not find tutorials helpful. Hence, I believe my users will be of mostly beginner and intermediate level.

Google Form Survey

I used a google form survey to find out more about my stakeholders and challenge the pre-existing conceptions I had about them. Using a survey over other methods of sampling, such as questionnaires, gave me access to a wider audience, as they are quick and easy to fill out.

Why did I use Google Forms to conduct my surveys?

- Google forms is an easy to access service, all you need to provide is a link
- It is standardized and easy to use due to its simple interface and common usage
- Data can be viewed in charts and a spreadsheet can be compiled, which makes it easy to analyse and spot trends.

Below I have added screenshots of the survey I sent out. It is followed by the spreadsheet of the data collected by my survey.

Exhibit 1: Google Survey Form

Henna Application - Google Form Henna Application

docs.google.com/forms/d/e/1FAIpQLSfCxHqMxkDxmjnWqdTjQXWsJfDousY8ae_HZKklBCZdwsEdbg/viewform



Henna Application

Creates unique henna designs and makes learning henna art easier. The application would also show you henna recommendations according to your likes and dislikes.

All responses are anonymous.

* Required

Please select your ethnicity *

White
 Asian/Asian British
 Black/ African/Caribbean/Black British
 Mixed/Multiple ethnic groups
 Other: _____

Do you believe that henna makes up an important part of your culture? *

Yes
 No

Henna Application - Google Form Henna Application

docs.google.com/forms/d/e/1FAIpQLSfCxHqMxkDxmjnWqdTjQXWsJfDousY8ae_HZKklBCZdwsEdbg/viewform

Would learning henna make you feel closer to your culture? *

Yes
 No

If henna is not considered a part of your culture - would you like to learn it for fun?

Yes
 No

How old are you? *

Under 18
 18-24
 25-34
 35-44
 45-54
 55-64
 65 and over

Are you : *

Female

Henna Application - Google Form Henna Application

docs.google.com/forms/d/e/1FAIpQLScxHqMxkDxmnlWqdTjQXWsUfDousYBae_HZKklBCZdwsEDbg/viewform

Are you : *

Female
 Male
 Prefer not to say
 Other: _____

Would you feel comfortable going to a henna artist during Covid? *

Yes
 No
 Maybe

What level is your henna ability? *

Expert (You do henna in professional environments)
 Advanced (You can do henna for other people)
 Beginner (Your can do henna but not good enough for functions or events)
 NA (You've never done henna before)

Would you like to learn/ improve your henna ability? *

Yes
 ..

Henna Application - Google Form Henna Application

docs.google.com/forms/d/e/1FAIpQLScxHqMxkDxmnlWqdTjQXWsUfDousYBae_HZKklBCZdwsEDbg/viewform

Would you like to learn/ improve your henna ability? *

Yes
 No

What features would you find useful in a henna application? *

Generation of random henna designs
 Designs being adapted to your taste
 The ability to pick the henna design style (e.g. wedding, Arabic)
 Links to Youtube tutorials
 Different henna design complexity levels
 The ability to pick and mix your henna design
 The ability to pick henna for specific occasions
 Henna designs being specialise to body parts
 Henna designs generated for henna artwork

Any other features you would like to have/find useful in a henna application? If yes, please elaborate.

Your answer _____

Which devices would you like the application to be available on? *

Mobile

Henna Application - Google Form Henna Application

docs.google.com/forms/d/e/1FAIpQLScxHqMkkDxmnlWqdTjQXWsUfDousY8ae_HZKkl8CZdysEDbg/viewform

Which devices would you like the application to be available on? *

Mobile
 Desktop
 Laptop
 Other: _____

What methods do you currently use to look for henna designs / learn them?

Henna books
 Youtube tutorials
 Google images
 Pinterest
 Other: _____

Are there any applications you currently use for henna designs/ learning?

Your answer _____

Do you have any disabilities?

Your answer _____

Submit

!

Exhibit 2: Google Survey Responses

| Timestamp | Please select your ethnicity | Do you believe that items makes up an important part of your culture or career? | Would learning about items make you feel more confident in your culture or career? | If items is not considered part of your culture or career, how old are you? Are you: Under 18, 18-24, 18-35, 35-49, 50+. | Would you feel comfortable doing a task involving items? | What level is your items ability? | Would you like to improve your skills? | What feature would you find useful in a items application? | Any other features you would like to benefit from in a items application? If yes, please state all. | Which devices would you like to use for items application? | What methods do you currently use to pick items designs? | Are there any applicable circumstances? Do you have any disabilities? |
|------------------|------------------------------|---|--|--|--|---|--|---|---|--|--|---|
| 14/12/2020 19:13 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | Confident | No | Advanced (You can do items for one people) | Yes | Mobile, Laptop | YouTube tutorials, Google Images | |
| 14/12/2020 19:22 | Asian/Asian British | Yes | Yes | 18-24 Female No | Yes | NA (You've never done items before) | Yes | NA (You've never done items before) Yes | Mobile, Laptop | YouTube tutorials | No | No |
| 14/12/2020 19:23 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | Beginner (You can do items but not good enough for functions or events) | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:23 | Asian/Asian British | Yes | Yes | 18-24 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Laptop | YouTube tutorials | No | No |
| 14/12/2020 19:24 | White | No | No | Under 18 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Laptop | YouTube tutorials | No | No |
| 14/12/2020 19:24 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | Beginner (You can do items but not good enough for functions or events) | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:24 | Asian/Asian British | Yes | Yes | 18-24 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Laptop | YouTube tutorials | No | No |
| 14/12/2020 19:25 | Asian/Asian British | Yes | No | 18-24 Female No | Yes | NA (You've never done items before) No | Yes | NA (You've never done items before) No | Mobile, Laptop | YouTube tutorials | No | No |
| 14/12/2020 19:27 | Asian/Asian British | Yes | Yes | Under 18 Male No | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:29 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:30 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Desktop, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:31 | Asian/Asian British | No | No | Under 18 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Desktop, Laptop | YouTube tutorials, Google Images | No | No |
| 14/12/2020 19:31 | Asian/Asian British | Yes | Yes | Under 18 Female No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Laptop | Google Images | No | No |
| 14/12/2020 19:33 | Asian/Asian British | Yes | Yes | Under 18 Male No | Yes | NA (You've never done items before) Yes | Yes | NA (You've never done items before) Yes | Mobile, Laptop | Google Images | No | No |
| 14/12/2020 19:33 | Asian/Asian British | No | Yes | Under 18 Female No | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop | Google Images | No | No |

| | | | | | | | | | | |
|------------------------------------|-----|-----|----------|--------|--|--|--|-------------------------|---------------------------------|---------------------------------|
| 1/1/2020 19:34 Asian/Asian British | Yes | No | Under 18 | Male | Na (You've never done items before) | Na (You've never done items before) No | Under 18 Male | Desktop, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 19:36 Asian/Asian British | Yes | No | Under 18 | Male | Brainer (You can do items but not good enough for functions or events) | Brainer (You can do items but not good enough for functions or events) Yes | Under 18 Male | Mobile, Laptop | YouTube tutorial | |
| 1/1/2020 19:37 Asian/Asian British | Yes | Yes | Under 18 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile | YouTube tutorial, Google Images | |
| 1/1/2020 19:44 Asian/Asian British | Yes | Yes | 18-24 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | 18-24 Female | Mobile, Laptop | YouTube tutorial | |
| 1/1/2020 19:44 Asian/Asian British | Yes | Yes | Under 18 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile | YouTube tutorial | |
| 1/1/2020 19:45 Chinese | No | No | Under 18 | Female | Na (You've never done items before) | Na (You've never done items before) No | Under 18 Female | Mobile | Mobile | |
| 1/1/2020 19:48 Asian/Asian British | Yes | Yes | Under 18 | Female | Brainer (You can do items but not good enough for functions or events) | Brainer (You can do items but not good enough for functions or events) Yes | Under 18 Female | Mobile, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 19:48 Asian/Asian British | Yes | Yes | 18-24 | Female | Na (You've never done items before) Yes | Brainer (You can do items but not good enough for functions or events) Yes | 18-24 Female | Mobile, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 19:52 White | No | No | Under 18 | Female | No | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 19:55 Asian/Asian British | Yes | Yes | Under 18 | Female | No | Brainer (You can do items but not good enough for functions or events) | Brainer (You can do items but not good enough for functions or events) Yes | Under 18 Female | Mobile, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 19:58 Asian/Asian British | No | Yes | 18-24 | Male | No | Na (You've never done items before) | Na (You've never done items before) Yes | 18-24 Male | Mobile, Desktop, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 19:58 Asian/Asian British | No | No | Under 18 | Male | No | Brainer (You can do items but not good enough for functions or events) | Brainer (You can do items but not good enough for functions or events) Yes | Under 18 Male | Mobile, Desktop, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 20:08 Asian/Asian British | Yes | No | Under 18 | Female | No | Na (You've never done items before) | Na (You've never done items before) No | Under 18 Female | Mobile, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 20:08 Asian/Asian British | Yes | Yes | Under 18 | Female | No | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 20:14 Asian/Asian British | Yes | Yes | 18-24 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | 18-24 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 20:25 Asian/Asian British | Yes | Yes | 18-24 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | 18-24 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 20:26 Asian/Asian British | Yes | Yes | Under 18 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 20:26 Asian/Asian British | Yes | Yes | Under 18 | Female | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images | |
| 1/1/2020 20:42 Asian/Asian British | Yes | Yes | Under 18 | Female | No | Na (You've never done items before) | Na (You've never done items before) Yes | Under 18 Female | Mobile, Desktop, Laptop | YouTube tutorial, Google Images |
| 1/1/2020 20:44 Asian/Asian British | No | No | Under 18 | Female | No | Brainer (You can do items but not good enough for functions or events) | Brainer (You can do items but not good enough for functions or events) Yes | Under 18 Female | Mobile, Laptop | Google Images, Print |

| | | | | | | | | |
|--------------------------------------|-----|-----|-----|------------------------|---|---|-------------------------|---|
| 1/1/2020 20:50 Asian/Asian British | Yes | No | No | Under 18 Female Yes | N/A (You've never done items before) No N/A (You've never done items before) Yes | Design items affected by your skills. The ability to pick the items design style (e.g. welding, Arctic). Generation of random items designs. Links to YouTube channel. The ability to pick and mix design components. The ability to pick and mix specific tools/parts. | Mobile, Desktop, Laptop | YouTube tutorials |
| 1/1/2020 20:52 Asian/Asian British | Yes | Yes | Yes | Under 18 Male No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Laptop |
| 1/1/2020 21:08 Asian/Asian British | Yes | Yes | Yes | Under 18 Male Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:10 Asian/Asian British | Yes | Yes | Yes | Under 18 Male Native | 45-54 Male Native | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Handbooks |
| 1/1/2020 21:12 Asian/Asian British | No | No | No | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Handbooks |
| 1/1/2020 21:13 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Handbooks |
| 1/1/2020 21:23 Asian/Asian British | Yes | Yes | Yes | 18-24 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:23 Asian/Asian British | Yes | Yes | Yes | 18-24 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:34 Asian/Asian British | Yes | Yes | Yes | 25-34 Male Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:35 Asian/Asian British | Yes | Yes | Yes | 18-24 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:37 Asian/Asian British | Yes | Yes | Yes | 18-24 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 21:44 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 22:02 Asian/Asian British | Yes | Yes | Yes | Under 18 Female Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 22:06 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | Google Images |
| 1/1/2020 22:39 Asian/Asian British | Yes | Yes | Yes | Under 18 Male Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | My imagination |
| 1/1/2020 23:49 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | YouTube tutorials, Project |
| 1/1/2020 0:33:35 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | YouTube tutorials, Project |
| 1/1/2020 0:33:36 Asian/Asian British | Yes | Yes | Yes | Under 18 Male Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | Google Images, Project |
| 1/1/2020 2:20:49 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | Google Images, Project |
| 1/1/2020 2:20:51 Asian/Asian British | Yes | Yes | Yes | Under 18 Female No | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | Google Images, Project |
| 1/1/2020 2:24:12 Asian/Asian British | Yes | No | Yes | 18-24 Female Native | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop | YouTube tutorials, Google Images, Project |
| 1/1/2020 8:04:45 Asian/Asian British | No | No | Yes | Under 18 Female Yes | N/A (You've never done items before) Yes | Beginner (You can do items but not good enough for functions or events) Yes | Mobile | No |

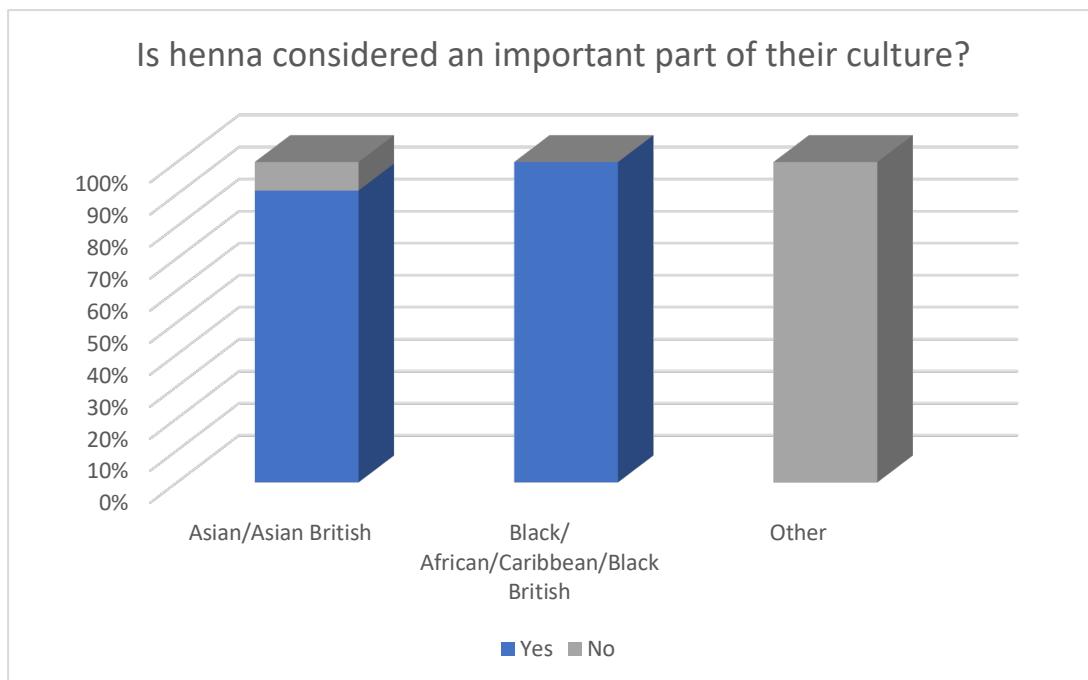
| | | | | | | | | | | | | | | |
|---|-----|-----|----------|--------|----------|--------|---|--|--|---|-------------------------|---|----|----|
| r13/2020 10:46:52 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs for other people) Yes Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Designs items all to fit your taste. The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. Different items design completely events. Items designs are very specific to each part.</p> | Mobile, Laptop | Patient, YouTube, Google Images, Print | No | No |
| r13/2020 10:53:20 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | Google Images, Print | No | No |
| r13/2020 10:54:43 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | Yes | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Mobile | Google Images, Print | No | No |
| r13/2020 11:00:08 Asan/Han/Korean/British | Yes | Yes | No | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | Yes | <p>Links to YouTube tutorials. The ability to pick items for specific occasions.</p> | Mobile | Google Images, Print | No | No | |
| r13/2020 11:04:23 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | Yes | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Mobile | Google Images, Print | No | No |
| r13/2020 11:05:07 Asan/Han/Korean/British | No | No | No | Yes | Under 18 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) No</p> | No | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Laptop | YouTube Tutorials, Google Images, Print | No | No |
| r13/2020 11:13:48 British | Yes | Yes | Yes | Yes | Under 18 | Female | Male | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile, Laptop | YouTube Tutorials, Google Images, Print | No | No |
| r13/2020 11:17:50 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 45-54 | Female | Male | <p>Beginner (You can do designs but not good enough for pictures or events) No</p> | No | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Mobile | Google Images, Print | No | No |
| r13/2020 11:18:27 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile, Laptop | YouTube Tutorials, Google Images, Print | No | No |
| r13/2020 11:19:13 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 25-34 | Female | Yes | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Mobile | Google Images, Print | No | No |
| r13/2020 11:23:55 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 45-54 | Female | Male | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile, Laptop | YouTube Tutorials, Google Images, Print | No | No |
| r13/2020 12:23:34 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | Male | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>The ability to pick the items design style e.g. wedding, birthday, name using a logo generator, items to YouTube tutorials. The ability to pick items for specific occasions. Items designs are very specific to each part.</p> | Mobile | Google Images, Social media App | No | No |
| r13/2020 12:51:47 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 12:53:43 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 12:59:30 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 45-54 | Female | Male | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 13:20:36 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 13:30:02 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 13:35:49 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile, Desktop, Laptop | YouTube Tutorials, Google Images | No | No |
| r13/2020 13:40:47 Asan/Han/Korean/British | Yes | No | Under 18 | Female | 45-54 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile, Laptop | YouTube Tutorials, Google Images | No | No |
| r13/2020 14:14:11 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 45-54 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |
| r13/2020 14:15:28 Asan/Han/Korean/British | Yes | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do designs but not good enough for pictures or events) Yes</p> | No | <p>Links to YouTube tutorials</p> | Mobile | YouTube Tutorials, Google Images | No | No |

| | | | | | | | | |
|--|-----|-----|-----|----------|--------|-------|--|---|
| r13/2020 14:22:30 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Generation of random items designs |
| r13/2020 14:34:41 Asim/Han/British | Yes | Yes | Yes | 15-24 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Generation of random items designs |
| r13/2020 15:26:57 British | Yes | Yes | Yes | 35-44 | Male | No | <p>NA (You never done items before) Yes</p> | Generation of random items designs |
| r13/2020 16:00:27 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Generation of random items designs |
| r13/2020 16:30:08 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | Maybe | <p>Beginner (You can do items but not good enough for functions or events) No</p> | Generation of random items designs |
| r13/2020 20:27:25 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Generation of random items designs |
| r13/2020 22:40:14 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Generation of random items designs |
| r13/2020 23:05:55 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) No</p> | Generation of random items designs |
| r14/2020 3:43:34 Asim/Han/British | Yes | Yes | Yes | 15-24 | Female | Yes | <p>NA (You never done items before) Yes</p> | Ability to pick the items design style (e.g. wedding, Arabic) |
| r14/2020 16:01:19 Asim/Han/British | Yes | Yes | Yes | 35-44 | Male | Maybe | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick the items design style (e.g. wedding, Arabic) |
| r14/2020 16:45:23 Asim/Han/British | Yes | No | No | 35-44 | Male | No | <p>NA (You never done items before) No</p> | Ability to pick and mix your items design style (e.g. wedding, Arabic). The ability to pick mix your items design. The ability to pick items for specific occasions. Items designs being good enough for functions or events) Yes |
| r14/2020 18:02:29 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick the items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |
| r14/2020 18:05:52 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | Maybe | <p>Beginner (You can do items but not good enough for functions or events) No</p> | Ability to pick items for specific occasions |
| r14/2020 19:18:47 Asim/Han/British | Yes | Yes | Yes | 35-44 | Male | Maybe | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |
| r14/2020 21:03:55 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |
| r15/2020 4:13:13 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |
| r15/2020 16:11:59 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |
| r15/2020 16:10:40 Blank Arabic/Cambodian/British | Yes | Yes | Yes | Under 18 | Female | No | <p>Beginner (You can do items but not good enough for functions or events) Yes</p> | Ability to pick items design style (e.g. wedding, Arabic). Click to YouTube links. Different items design company links |

| | | | | | | | | |
|--|-----|-----|-----|----------|--------|-------|---|--|
| r13/2020 14:22:30 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Generation of random items designs |
| r13/2020 14:34:41 Asim/Han/British | Yes | Yes | Yes | 15-24 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Generation of random items designs |
| r13/2020 15:26:57 British | Yes | Yes | Yes | 35-44 | Male | No | NA (You never done items before) Yes | Generation of random items designs |
| r13/2020 16:07:27 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Generation of random items designs |
| r13/2020 16:30:08 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | Beginner (You can do items but not good enough for functions or events) No | Generation of random items designs |
| r13/2020 20:27:25 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Generation of random items designs |
| r14/2020 22:40:14 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | The ability to pick the items design style (e.g. wedding, Arabic) and mix them |
| r14/2020 23:05:55 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | Beginner (You can do items but not good enough for functions or events) No | The ability to pick the items design style (e.g. wedding, Arabic). The ability to pick mix your items design. The ability to pick items for specific occasions. Items designs being good enough for functions or events. Yes |
| r14/2020 23:43:34 Asim/Han/British | Yes | Yes | Yes | 15-24 | Female | Yes | NA (You never done items before) Yes | Designs being aligned to your taste |
| r14/2020 16:01:19 Asim/Han/British | Yes | Yes | Yes | 35-44 | Male | No | Beginner (You can do items but not good enough for functions or events) Yes | Items designs being specialised to body parts |
| r14/2020 16:45:23 Asim/Han/British | Yes | No | No | 35-44 | Male | No | Beginner (You can do items but not good enough for functions or events) No | Mobile, Desktop, Laptop |
| r14/2020 18:02:29 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r14/2020 18:05:52 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | Maybe | Beginner (You can do items but not good enough for functions or events) No | Mobile, Desktop, Laptop |
| r14/2020 19:05:52 Asim/Han/British | Yes | Yes | Yes | 35-44 | Male | Maybe | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r14/2020 20:18:47 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r14/2020 21:03:55 Asim/Han/British | Yes | Yes | Yes | 45-54 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r15/2020 4:13:13 Asim/Han/British | Yes | Yes | Yes | 35-44 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r15/2020 16:11:59 Asim/Han/British | Yes | Yes | Yes | Under 18 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Desktop, Laptop |
| r15/2020 16:10:40 Blank Arabic/Cambodian/British | Yes | Yes | Yes | Under 18 | Female | No | Beginner (You can do items but not good enough for functions or events) Yes | Mobile, Laptop |

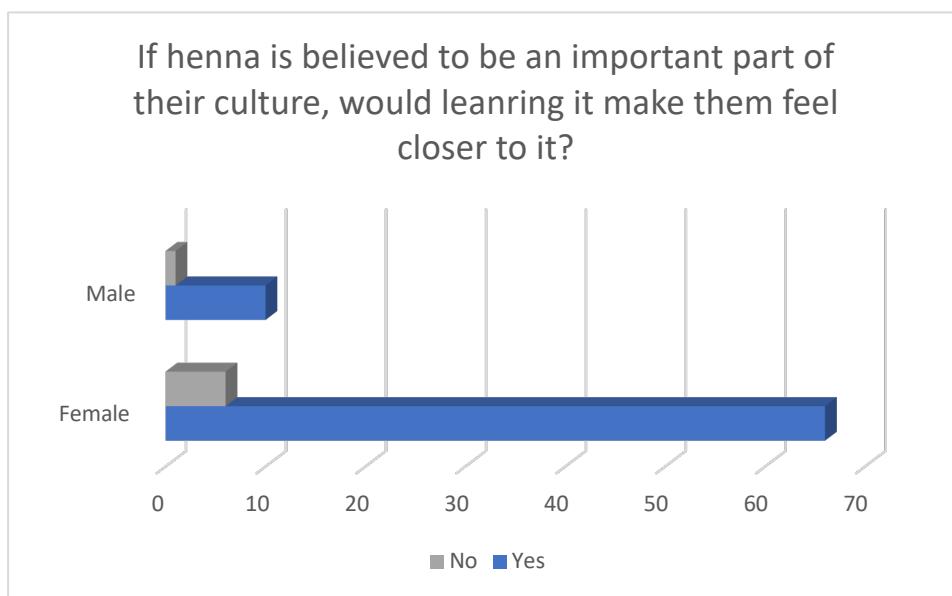
How have my assumptions about my stakeholder characteristics changed?

Exhibit 3: Google Survey Response



From this graph it is apparent that my initial conceptions about which ethnic groups considering henna culturally important is true; Asian and Black/African/Caribbean surveys strongly felt henna was an integral part of their ancestral traditions. Thus, my stakeholders will primarily be of this origin, allowing me to focus on designs and elements which are specific to them and have cultural meanings.

Exhibit 4: Google Survey Response



This question highlights the fact that regardless of gender, henna does have strong cultural links. In both males and females, the majority would feel closer to their culture by learning henna. Hence my application will help many people learn the art of their heritage and bond with their ancestors. My target market is both males and females, I need to take a gender-neutral approach when designing my application, in order to make men feel accepted into an industry which breaks traditional stereotypes.

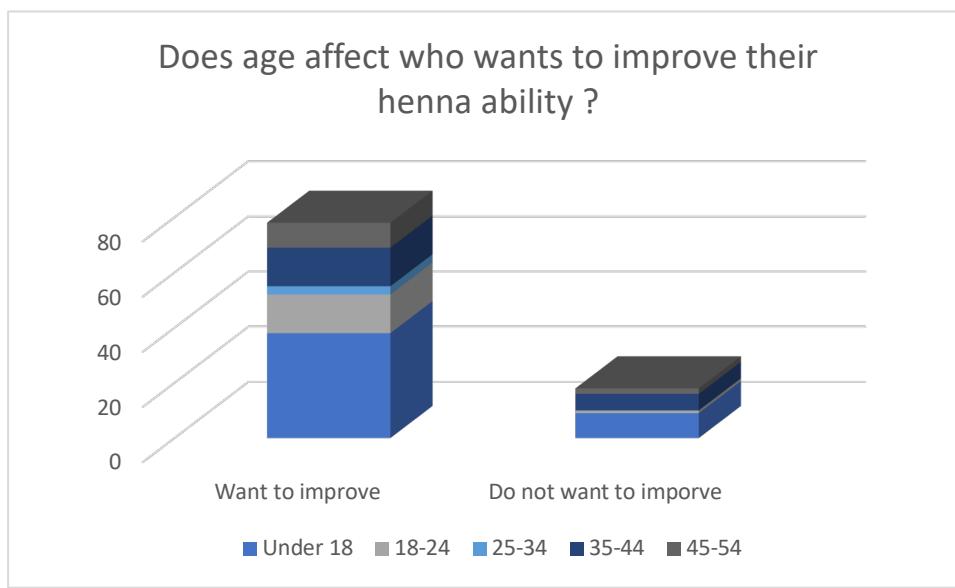
If henna is not considered an important part of their culture, would they still like to learn it?

Exhibit 5: Google Survey Response



In this question a strong gender bias was present. 100% of females who were not of Asian or Black/African/Caribbean cultures said they would like to learn henna for fun, whereas 100% of men said they would not. This may be due to the lack of gender and stereotypically cultural links the men may feel. Hence this once again emphasis that a gender-neutral application is needed to make men feel welcome to use the application. It is also vital that some of my test users/stakeholders are female, as they are more likely to use the application.

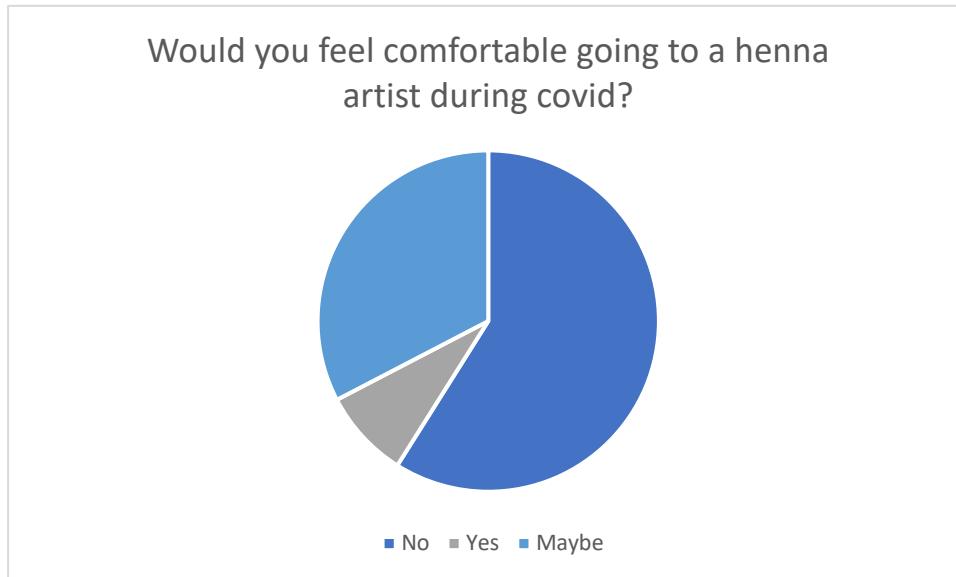
Exhibit 6: Google Survey Response



From the graph it is apparent there is no correlation between age and a desire to improve henna ability. Due to

this my application needs to be made suitable for all age groups and take into consideration the factor older people may not be as technologically advanced. To cater for this my application will need to have a simple and easy navigation system and have minimal design be intuitive for people to use.¹

Exhibit 7: Google Survey Response



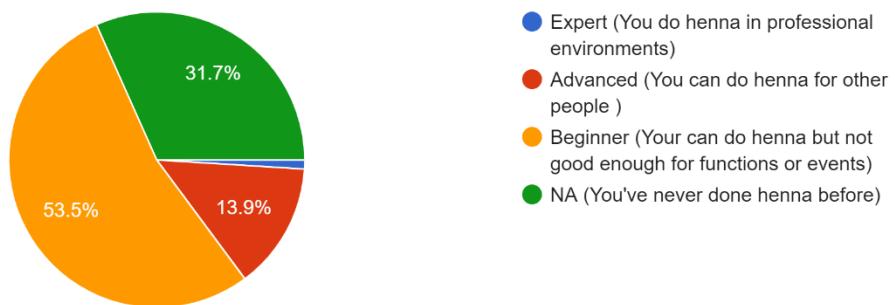
As my predictions suggested, the pandemic has increased the need for this application. Over 50% of people would feel uncomfortable going to a henna artist during COVID-19. As highlighted above henna has a strong cultural association, and despite festivals, such as Diwali and Eid being celebrated in isolation, this application could enable them to have some festive spirit. It would also be significant in events such as weddings, as in many cultures henna is essential before getting married, as it serves as protection and well wishes.

Exhibit 8: Google Survey Response

¹ <https://www.businessofapps.com/news/things-to-be-taken-care-of-when-designing-apps-for-the-elderly/>

What level is your henna ability?

101 responses



This chart shows that around 85% of my users will be beginners or never have done henna before. Hence my application will be focused on teaching people the basic steps of henna, such as how to draw common elements, as otherwise large henna designs may seem like an impossible or daunting task. This could be done via the inclusion of tutorials or instructions. However, these need to be optional as a small proportion of my users will be advanced/ experts hence will find these an unnecessary hindrance.

Selected Stakeholders

As my project will heavily have an iterative approach and change will regularly be made based on stakeholder feedback, I asked a list of people if they would be willing to take this role. Below are listed some basic details about those who replied and were happy to fill in this role.

Tanvi

Age: 18

Henna Ability: Beginner - has tried henna a few times but still requires assistance (e.g. tutorials, books) to do her henna designs

Ethnicity: South Asian

Neha

Age: 45

Henna Ability: Expert - has been doing henna for weddings and events for many years

Ethnicity: South Asian

Sarah

Age: 62

Henna Ability: Never tried henna before

Ethnicity: White

Other: Visual Impairment

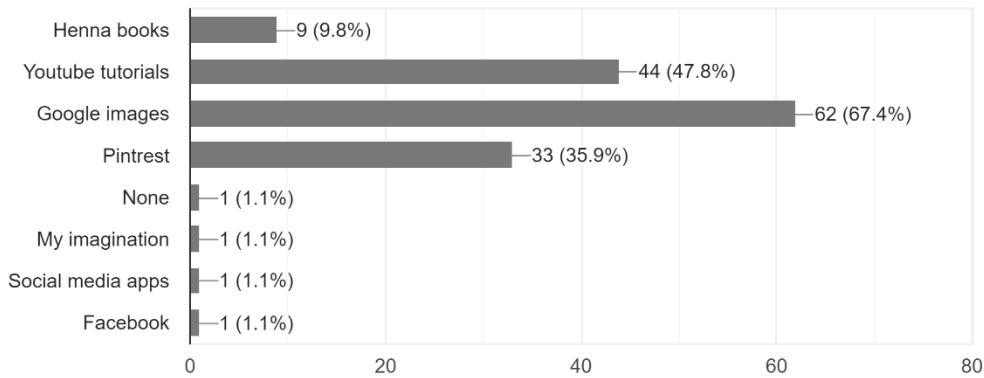
As can be seen I have tried to select a range of users, who have varying ages, henna abilities and ethnicities. This will allow me to gain as diverse feedback as possible, and make my application optimised for the widest possible audience.

What solutions already exist to similar problems?

Exhibit 9: Google Survey Response

What methods do you currently use to look for henna designs / learn them?

92 responses



To find out what methods are commonly used to learn and look for henna designs, I included the above question in my survey. From the results we can see that the most popular methods appear to be henna books, YouTube tutorials, google images and Pinterest. Hence, I have focused on the analysing of these methods in the following section. I have also analysed other solutions which exist and studied why these are not popular to learn where their flaws are and improve my application.

In person teaching

Exhibit 10: Henna tutoring reviews

The screenshot displays two profiles from the Superprof platform:

Sasha (Henna/Mehndi art lessons here!)

- Tutor profile verified
- Response Time: 24h
- Lessons offered by Sasha: Individual - In group
- The lessons will be held: At her home - At your home
- By webcast
- Teaching subjects: Other subjects
- Levels: All Levels

Pam Sangha (Henna Artist & Tutor)

- Total students: 222
- Ratings: 58
- About me:
I am a professional henna artist for 15 years providing my services at fairs and events, corporate events and even teaching Henna in schools and colleges. If you want to learn from a legit henna experienced working artist who have come to the right place will show you exactly what do and share all my Henna Artist Tips and Tricks.
- I love creating new Henna patterns and creating new designs for my clients. It is never to late for anybody to learn new hobbies and skills and with this course it will help develop your creative side and hopefully you will have fun learning.
- My courses (1)
- Learn The Art Of Henna From The Comfort Of Your Home
More Details
£14.99 - £20
1 Total User - 7 Ratings - All Levels
£15.99 (£14.99)

Course Breakdown:

- Tricks of the trade to perfect henna cone making and learning the basic shapes and fillers to build the right grounding foundation
- Learn SV signature veil and bridal designs
- Learn a variety of different henna styles including Bridal Henna, Indian Henna, Arabic Henna, Mandala Henna, Shading Henna, Inverse Henna, 3D Henna, festival Henna and henna tattoos.
- Learn to create your own henna designs and various secret tricks of the trade
- Portfolio building and presentation

Course Time: 9am - 5pm
Price: £549 (incl. VAT)
Location: London

Source: <https://www.superprof.co.uk>

Overview:

Due to henna as an art form, in person teaching is one of the most effective ways to learn it. Eventually a person must develop the required instinct to correct mistakes, as they are often to do with aesthetic issues, something which is impossible to learn via reading about it. Hence when interviewing some stakeholders these were their thoughts:

Stakeholders' comments:

Tanvi:

'I have always wanted to try a henna course to improve my ability as it would be helpful to have someone in real life there to help me. However the price is a massive put off. A digital tutor service in the application maybe a nice idea.'

Neha:

'Having done henna for several years, in person teaching would no longer be useful as I have my own style of doing henna.'

Sarah:

"For me, I would only every learn henna for fun, hence it is something I would want to do in my own time. Taking lessons from someone else would not fit around be schedule and take up too much time"

What I can apply to my application:

Due to restrictions on the manual labour, I have available, it would not be plausible to offer a digital tutorial service in real time. Furthermore considering the factor of pricing and flexibility, this would not be something which would meet my stakeholder needs on a majority. However, a suitable alternative could be linking to appropriate resources which can replicate a tutor, such as web pages and YouTube tutorials enabling people to self-teach and the app to be self-sufficient.

YouTube tutorials

Overview:

From personal experience I have found YouTube tutorials as one of the most financially affordable, flexible, and easy ways to learn henna. They enable you to find specific elements of whole designs and learn how to replicate them. Additionally videos combine both auditory and visual learning and combined with attempting to do henna yourself (kinaesthetic learning). Therefore this covers the three main learning style preferences.

YouTube also has a well-developed and advanced user interface, compared to other existing video applications. For example it has the ability to slow down and speed up videos, allowing people to go at their own pace. It is also a large platform hence has a wide range of tutorials available.

Stakeholders' comments:

Tanvi

'Henna video tutorials are a good way for me to learn, especially the ones which talk you through what they are doing. I find them a lot more useful over webpages as you can see the whole process. However, the ones which teach you how to do the whole design are often less helpful as I don't really learn myself, I just copy them. Therefore, I like looking at how to do the bits of the design I'm stuck on'

Neha

'I like henna tutorials as they give me knew inspiration. They also teach me ways which I can speed up my henna or improve my designs.'

Sarah

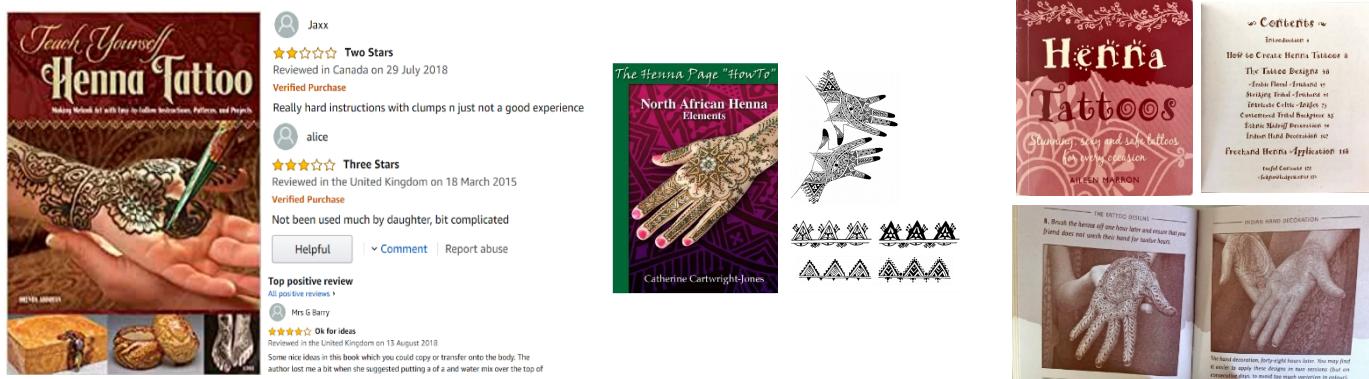
'Tutorials are a really nice way to learn as there are so many especially on YouTube. They allow me to learn in my own time as most of them are short. However, I often struggle when looking for specific element ideas as I don't really know what they are called.'

What I can apply to my application:

It would not be practical to replicate the wide variety and quality of the henna tutorials available on YouTube. So I have decided to integrate YouTube into my application. I intend to use a YouTube API to link each element in my design to a relevant YouTube tutorial, keeping in mind Tanvi's feedback about linking it to whole designs. This would allow users to pick when they need assistance and pick the video best suited to their likes. It would also include the functionality of YouTube (e.g. speeding up and slowing down tutorials).

Books

Exhibit 11: Henna Books



Source: <https://www.amazon.co.uk>

Overview:

Henna books all generally follow the same format. They provide basic instructions on how to make a henna cone, how to use the henna cone and the after care. They then follow with different design and pattern ideas. Sometimes they provide instructions on how to create the specific design elements.

Review analysis:

From the reviews it is apparent the form of written instructions often cause confusion and appear complicated. This may be due to the nature of an instruction book: it gives you snapshots of the process “frozen in time”, as opposed to being able to see the whole process fluidly. For further opinions I interviewed my stakeholders.

Stakeholders' comments:

Tanvi

“The only one of these books I really liked the look of was the North African Henna one due to its format as simple black and white images, which makes the patterns clear to see. The Henna Tattoos book has monochromatic photos; however, the background clutter can make be distracting. Also, the angles mean I can't see the whole design and have to use guess work which is hard and irritating for a beginner”

Neha

“As an experienced henna artist, I never use books. Originally when I started learning they provided some useful instruction on how to learn, however in our modern era you would probably find more eloquent and useful information online. Books also get very repetitive, once you've done a design its useless, so I would prefer something which generated unique designs.”

Sarah

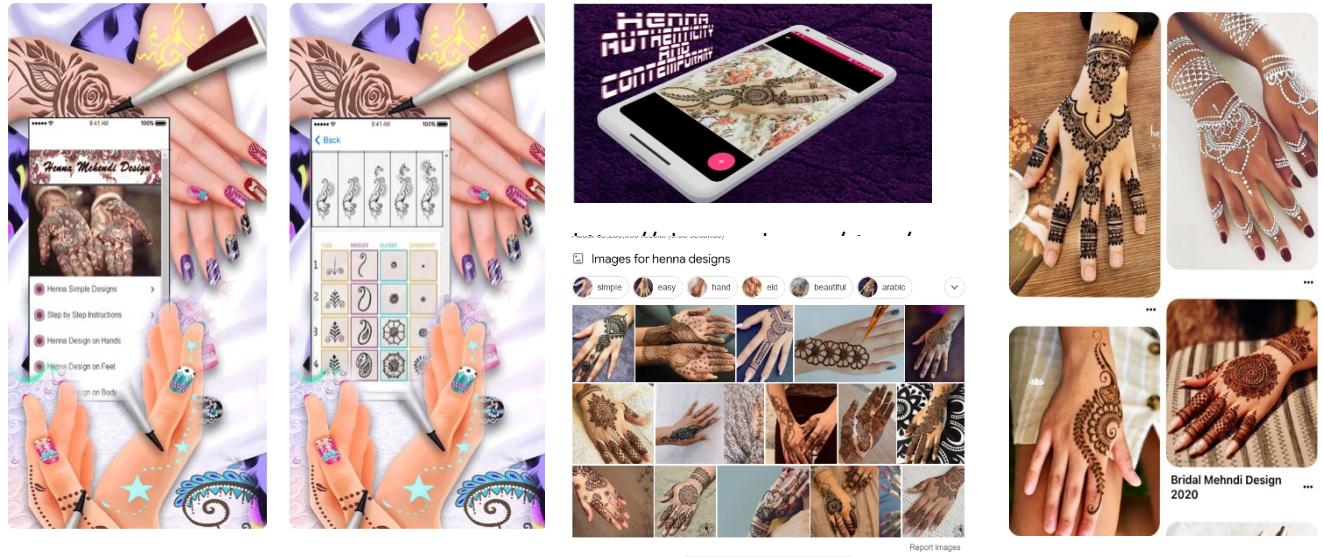
"I actually like books quite a lot. They provide information which is available offline without the hassle of having to download or google anything."

What I can apply to my application:

From my stakeholders I can see that they prefer a simple interface, both application wise and for the presentation of the henna design. Thus, it may be a good idea to stick to a minimalistic colour pallet (e.g. monochromatic) when presenting the designs and a simple user interface.

Applications/ Pinterest/ Google Images

Exhibit 12: Henna Applications



Source: <https://apps.apple.com/us/app/henna-mehendi-design/id144010000>

Overview and what I can apply to my application

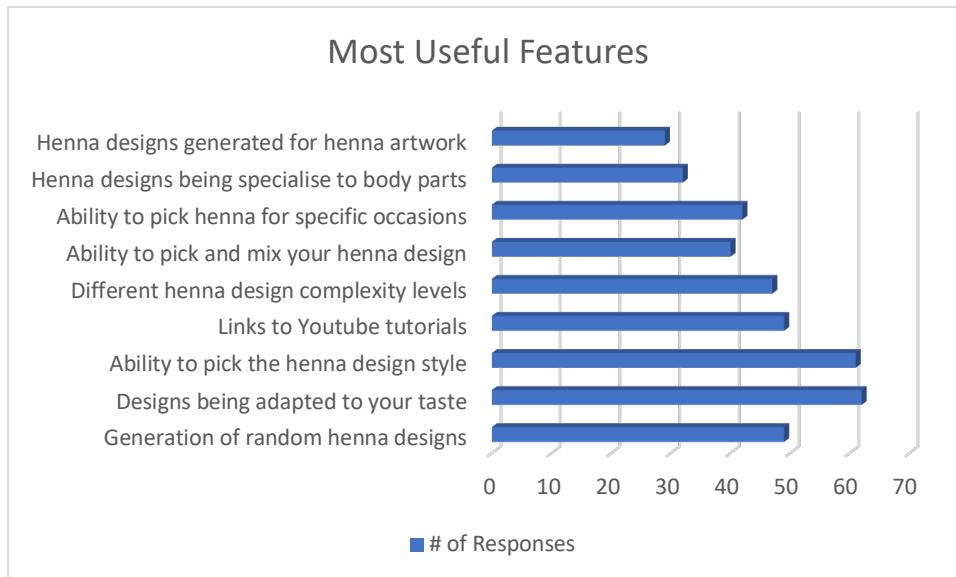
Current henna applications which exist mostly consist of a collection of images and different design ideas. Whilst not solely for henna designs, Pinterest is probably one of the most renowned places to search for henna, due to its ability to save images to a board and look back on them later. Google Images is also used in a similar manner to Pinterest, except them images must be saved locally on the computer. My application will use this feature by downloading the images in a photo format (such as JPEG).

Pinterest and Google Images also have a large variety of images which can be made more refined by selecting different suggestions once you have entered a search. This allows users to select designs which are suited to their taste. My application will automate this process, by machine learning.

Only one application appears which is specific to henna and not just a collection of images exists, it provides a series of four snapshots to look at how to construct a single element. This could be a particularly useful feature as it would break up a daunting complex design into several smaller achievable parts. However, this application currently only does this for a small design, my application will provide help on every element in a whole design, by using a YouTube API to link them to YouTube tutorials.

What are the main features of the project?

Exhibit 13: Google Survey Response



1. Henna Design Creator

The main purpose of this application is to provide the user with henna designs. The users will then be able to download these designs in a photo format (such as JPEG).

The program could be coded to draw different elements. These elements could be called in different combinations and follow different templates to create a variety of designs. Properties about these elements could be stored in a database.

2. Recommendations

The application will have the ability to provide designs which are most suited to a particular user's tastes. This was the most popular feature which came up in the survey.

The application will allow the user to achieve designs which are customised to their taste, which considering the fact henna is semi-permanent is an important factor. Henna comes in many categories and forms, yet currently it is hard to find patterns which appeal to people's specific taste.

This could be implemented using a recommendation algorithm in a similar way to Spotify recommending songs or Amazon recommending products. Initially the user could be asked to pick which images appeal to them, giving the code an initial idea of their preferences. After each design will have the option to like or dislike a design, providing more data.

3. Linked to YouTube videos

The user will be able to click on each element of the design and be taken to a corresponding YouTube video which contains a tutorial on how to do that pattern.

Despite its importance, henna art is never officially taught, usually it is orally passed down through generations. The link to YouTube videos will allow people to easily self-learn and participate in this tradition despite having a limited artistic ability. Furthermore, without the background knowledge, it can be very hard to find associated tutorials for the specific element as no official names are given to them.

This could be done by using the YouTube API, which would be free for the scale of this project.

Other features my stakeholder wanted (from the survey):

- “common mistakes and how to avoid them... recipes for making henna at home...”

This can be achieved via links to YouTube tutorials or webpages in a different section of the application. However mistakes are often particular to a scenario and cannot be generalised, thus it may be better to leave the users to find their own solutions via tools such as Google.

- “Step by step written tutorials with pictures”

From research (mentioned in the analysis of YouTube), it is apparent that video tutorials combined with practice are more helpful to most of the population, and as both are close substitutes, I have decided to provide the YouTube tutorial instead.

- Uploading henna designs to Pinterest

My application will have the ability to save the produced designs to their device, thus people can then upload this onto Pinterest if they desire.

- Sourcing nearby suppliers

Many supplier often add chemicals to for colour enhancing properties. Hence if not sourced from a suitable supplier these chemicals maybe toxic or irritants for the skin and have negative impacts. Therefore, it would be used to generate a suggested list computationally as there would be no human safety nets. Thus, I have decided not to implement this.

What are the limitations of the Project?

Exhibit 14: Google Cloud Platform Pricing Structure

Pricing and quotas

The price for Cloud Endpoints depends on the number of calls to your API, as described in the following table:

| API calls per month per billing account | Cost per million API calls |
|---|----------------------------|
| 0-2M | \$0.00 |
| 2M-1B | \$3.00 |
| 1B+ | \$1.50 |

If you pay in a currency other than USD, the prices listed in your currency on [Cloud Platform SKUs](#) apply.

Endpoints charges by its calls to [Service Control](#). Each API call processed by the Extensible Service Proxy (ESP) or the Cloud Endpoints Frameworks are reported as a tracked operation by the Service Control API and is listed as line item for Service Control on your bill.

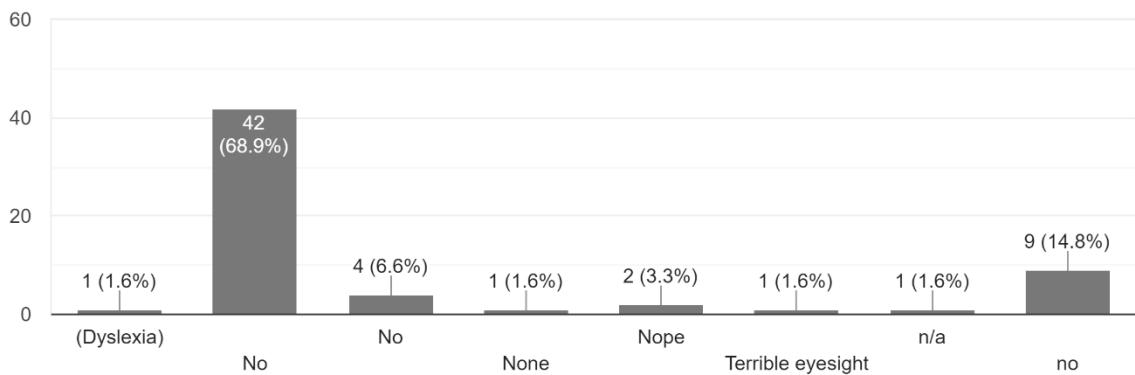
Source: <https://cloud.google.com/endpoints/pricing-and-quotas>

The YouTube API is free for 0 to 2 Million API calls per month, which is approximately 70000 calls per day. My application is for a niche target market, and henna is not used daily, thus does not need to be designed for more than 70000 users daily. I expect the API calls to be in low double digits on a regular basis and maybe triple during festivals.

Exhibit 15: Google Survey Response

Do you have any disabilities?

61 responses

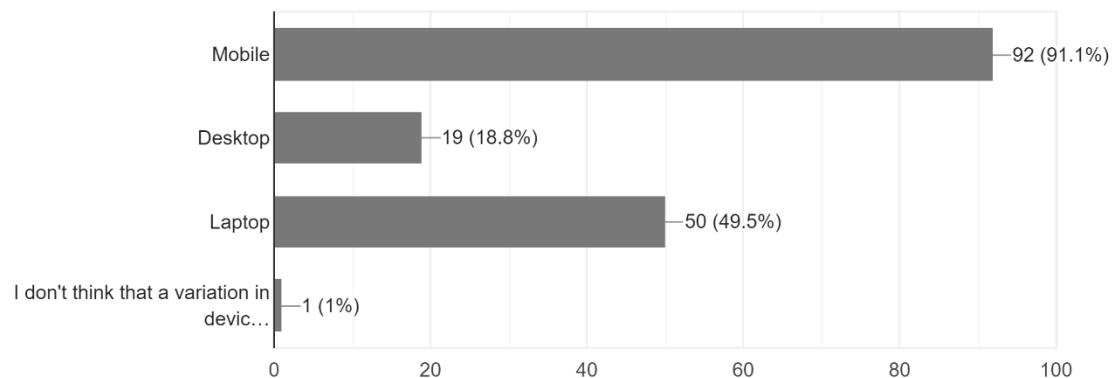


From my survey I can also see that some of the users may have disabilities, for example dyslexia and terrible eyesight. For this reason, my application needs to have a relatively simple and decluttered interface. I will also choose button options or dropdown over text to make it easier for them. My application will also have high contrast in the design to make it accessible to people with bad eyesight.

Exhibit 16: Google Survey Response

Which devices would you like the application to be available on?

101 responses



What hardware and software are required?

From my survey it is apparent the stakeholders would prefer this application to be available on mobile. However due to its functionality and nature as a graphical application with machine learning, it will require a higher processing power meaning it would operate very slowly on a mobile device. Furthermore, screen size is important in this application, as a bigger screen would allow the user to look at the design in more detail and for it to appear less clustered. Hence, I intend on making my application desktop based.

When asking Tanvi, Neha and Sarah they said that it may be inconvenient whilst doing henna, as you want to have the design easily accessible. I came up with the idea of being able to download the image in a picture format, this would then enable you to send it to a mobile device or print it out. Both thought this would be a useful practical solution. Hence to use my application the user will require a desktop or laptop device.

To make full use of my application the users will also require a Wi-Fi connection to access YouTube pages.

Summary of requirements

Hardware and software

Exhibit 17: Hardware Requirement

Hardware

| | |
|--|--|
| A computer capable of running the software | The computer needs sufficient processing power to run the application - which most modern-day computers do |
| Peripherals <ul style="list-style-type: none">● Monitor● Keyboard● Mouse | This will allow the user to interact with the application |
| An internet connection (and the ability to connect to it) | This will allow the user to access the full potential of the application by viewing YouTube tutorials |

Exhibit 18: Software Requirement

Software

| | |
|---|---|
| One of the specified OSs on https://www.componentsource.com/product/delphi-professional/compatibilities | These operating systems are supported by Delphi (the language the application will be written in) |
| Delphi Compiler | As it is compatible with all Delphi code (the language the application will be written in) |

Stakeholder requirements

Design:

Exhibit 19: Stakeholder Requirements

| Requirement | Explanation |
|--|--|
| An easy-to-use interface with limited typing | This will make my application intuitive and easy to use, which is especially important for older users who may already struggle with understanding new technology. |
| Menu bar | A menu bar should be at the top of the page containing the essential links to navigate the application. This will allow a streamlined user journey and make the application simple to use. |

| Requirement | Explanation |
|---|--|
| My application has a gender-neutral interface | The colour pallet and design elements chosen should be relatively gender neutral in order to make my application feel welcoming to everyone. |
| Lightweight Design | The design should be simple ensuring it is not too complicated to use the application |
| Colour pallet chosen is easy on the eye | The colour palettes chosen should not be too busy or bold, and high in contrast. This will make my application more suitable for people with certain visual impairments. |

FUNCTIONALITY:

Exhibit 20: Functionality Checklist

| | |
|---|--|
| The program generates henna designs | This is the main purpose of the program - to generate henna designs which the user can draw out. |
| A clear instruction menu on how to use the application | This will serve as a guide for any users who are unsure how to use the application. |
| Designs being specialised to a user | This will adapt the designs to the user's preferences, increasing the likelihood of the design being to the user's liking. |
| The ability to like or dislike each design | This is needed to collect data about the user's preferences which will be used to customised designs to a user. |
| Option to reset preferences | Allows the user to "reset" the application and get more neutral designs which are not as specific to the user |
| Designs being randomised each time | This will allow the user to generate new designs till they find one which matches their taste. |
| Ability to save the designs in an image format | This will allow the user to edit the design if they desire. They will also be able to send the design to a mobile. |
| My application should have designs which are inclusive of all the major cultures (which consider henna important) | This will make my application more inclusive of all cultures, allowing me to gain a wider user base. |
| Elements being linked to YouTube tutorials | This will allow the user to access tutorials specific to the design and learn how to draw the design. |
| Option to exit the application | Will allow the user to close the application |

How can the project's success be measured?

Success Criteria

Exhibit 21: Summary of Success Criteria

| Success Criteria | How to Evidence |
|---|---|
| Loading screen | Screenshot |
| An easy-to-use interface with limited typing | Screenshots of all input boxes (inc. buttons/textboxes/menus). Options should not require typing where possible (e.g. drop-down box instead of textbox) |
| Menu bar | Screenshot of the menu bar and all of its options |
| My application has a gender-neutral interface | Screenshot and justification of the colour pallet used (with the precise colour hex codes) |
| Lightweight Design | Screenshot of application pages which are not cluttered |
| Colour's pallet chosen in easy on the eye | Screenshot of colour pallet used (with the precise colour hex codes) with light background colour and dark font colour |
| The program generates henna designs | Screenshot of a generated henna design |
| A clear instruction menu on how to use the application | Screenshot of the button to access the menu and the instructions |
| Designs being specialised to a user | Screenshot of database values changing according to like and dislike buttons, code and testing |
| The ability to like or dislike each design | Screenshots of the buttons |
| Option to reset preferences | Screenshots of the reset button |
| Designs being randomised each time | 10 screenshots of different designs being produced consecutively and code |
| Ability to save the designs in an image format | Screenshot of the design open in a different application |
| My application should have designs which are inclusive of all the major cultures (which consider henna important) | Screenshot of design from each culture |

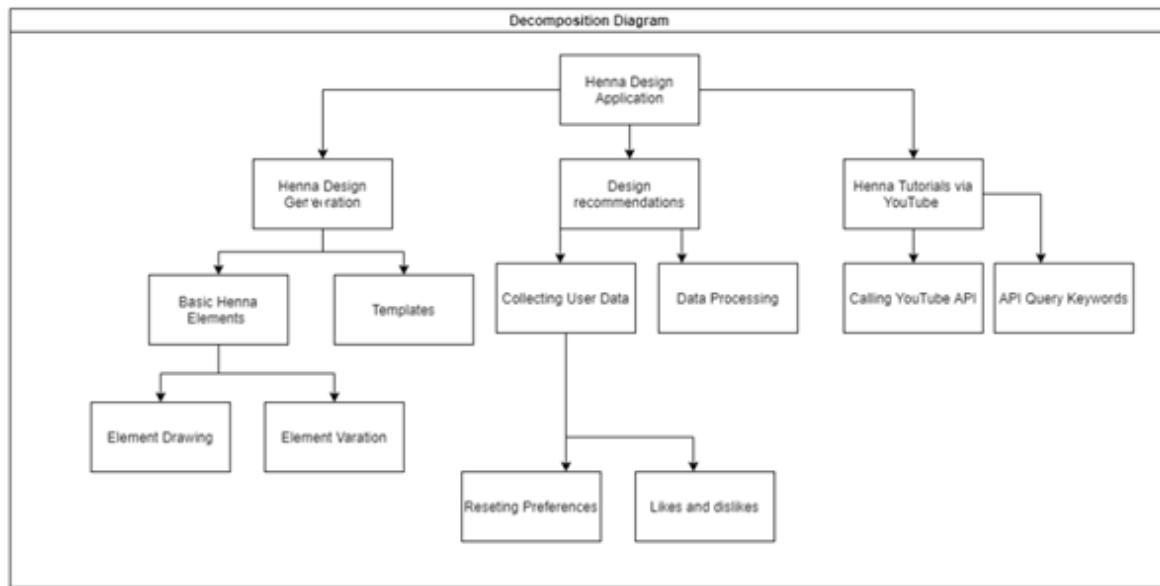
| Success Criteria | How to Evidence |
|--|--|
| Elements being linked to YouTube tutorials | Screenshot of buttons which open YouTube tutorials, screenshot of corresponding tutorial and code |
| Option to exit the application | Screenshot of button and code |

Design

Decomposition

Decomposition Diagram

Exhibit 22: Project Decomposition Diagram



Explanation for decomposition diagram

Initially the project seemed one huge, daunting task. Therefore, I decomposed the functionality of my program, as can be seen in the diagram above. This provides a visual overview of the project.

Each node now contains a smaller module, all of which complete a specific action. The modularity of it, means each task can be dealt with in isolation. This makes the initially huge project seem more manageable. It also provides a better, more defined structure for planning and structuring my project.

Decomposition in greater detail

Using my decomposition diagram the project can be broken down into the following sections:

Henna Design Generation

- Basic Henna Elements
 - Elements Drawing
 - Hand outline
 - Loading the hand into canvas on:
 - Project loading
 - New design generation
 - Henna elements
 - Create a visual representation of the core henna elements
 - Draw them computationally
 - Elements Variation
 - Add randomisation and variations to the basic henna elements
 - Templates
 - Plan out henna design templates
 - Plan where elements should be placed to achieve certain designs
 - Randomisation procedures to pick different combinations in order to increase combinations of henna patterns produced
 - Saving templates
 - Generating a new template

Design Recommendations

- Collecting user data
 - Resetting Preferences
 - Option to reset database
 - Likes and dislikes
 - When a design is liked/disliked in the table this needs to be registered
 - Appropriate data processing needs to be triggered
- Data Processing
 - Initial preferences
 - Design a database containing all elements
 - Fill in standard, constant database data (e.g. element name)
 - Likes
 - Weighting of all elements in design need to be increased
 - All similar templates weighting needs to be increases
 - Dislikes
 - Weighting of all elements in design need to be reduces
 - All similar templates weighting needs to be decrease

YouTube Videos

- API Query Keywords

- Extract the keywords from the database
 - Put these into a format the YouTube API can understand and will produce accurate results
- Calling the YouTube API
 - Dynamically load henna tutorials based on query for that element

User Interface

Below I have inserted screen shots of the intended layout of my application. This provides a graphical idea of what my interface should look like. Below each screen I have explained the functions of the elements on that page.

Screen 1 – Loading Screen

Exhibit 23: Loading Screen Prototype



This is the loading screen. It will appear when the user opens the application and will disappear when they click on it. This provides a chance for the application to fetch and load the appropriate data and work efficiently, preventing future lag.

It also creates a gentler user journey, as they are not thrust into the busier main screen instantly, instead they are eased into the application.

Setting up components (not code)

1. Border Elements
2. Download image
3. Create a copy and rotate this 180 degrees
4. Create TImage components in Delphi

5. Upload the border image files
6. Place these in opposite corners of the page

Text

1. Create two labels and type text into it
2. Set the title to larger font size and custom font (via its property)
3. Set the caption to a smaller font size and to the main font of the application (via its property)

Setting up colours (do directly from properties not via code)

1. Change labels font colour property to custom dark brown
2. Change form background colour property to custom light brown
3. Photoshop side border elements to custom dark brown

Loading screen automatically disappearing

PROCEDURE onTimerTriggered

```
loadScreen.Hide
```

```
homepage.Show
```

END PROCEDURE

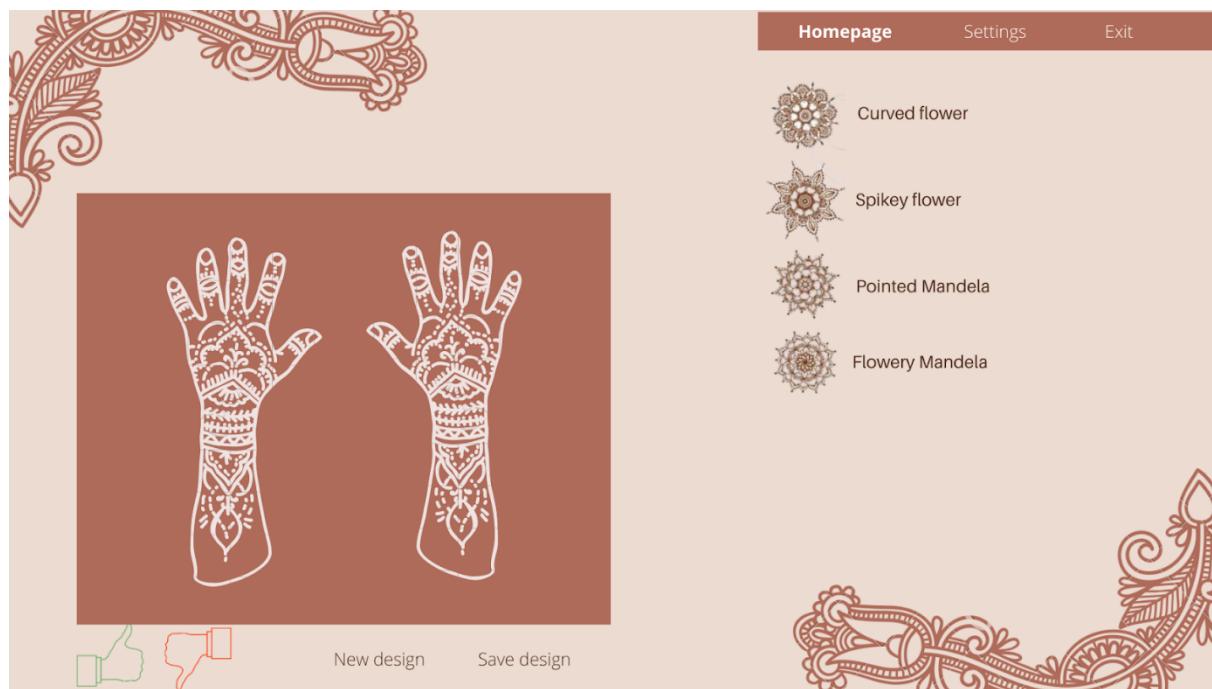
1 second after application started = call(onTimerTriggered)

User Feedback:

When my test users were shown these prototypes, they generally complimented the design and had no criticisms.

Screen 2 – Homepage

Exhibit 24: Homepage Prototype



This is the homepage of my application. It contains most of the features of my application.

Canvas

The canvas displays the henna designs. It has a rust orange background, and the design will be drawn on it in an off-white shade – inverted colours to henna on the skin. This provided a high contrast design making it easy to see.

The canvas is also a simple block. This minimises the distractions around the henna, making it easy to focus on. This is well suited to my target market of beginners.

Like Dislike Buttons

These buttons will allow the users to like or dislike the generated design. This feature is used to give more accurate patterns according to the users taste and is discussed later in the section of the recommendation algorithm.

Element Breakdown

The element breakdown on the right-hand side of the page provides the user with the chance to click on each element used in the henna design and be led to the YouTube tutorial teaching them how to draw that element.

New Design

This button will generate a new design

Save Design

This button will save an image of the henna design to their computer.

Menu Bar

- Setting – This option will take the user to the settings page
- Exit
Link to success criteria: Option to exit the application

```
PROCEDURE menuBar.exit.onClick
    application.exit
END PROCEDURE
```

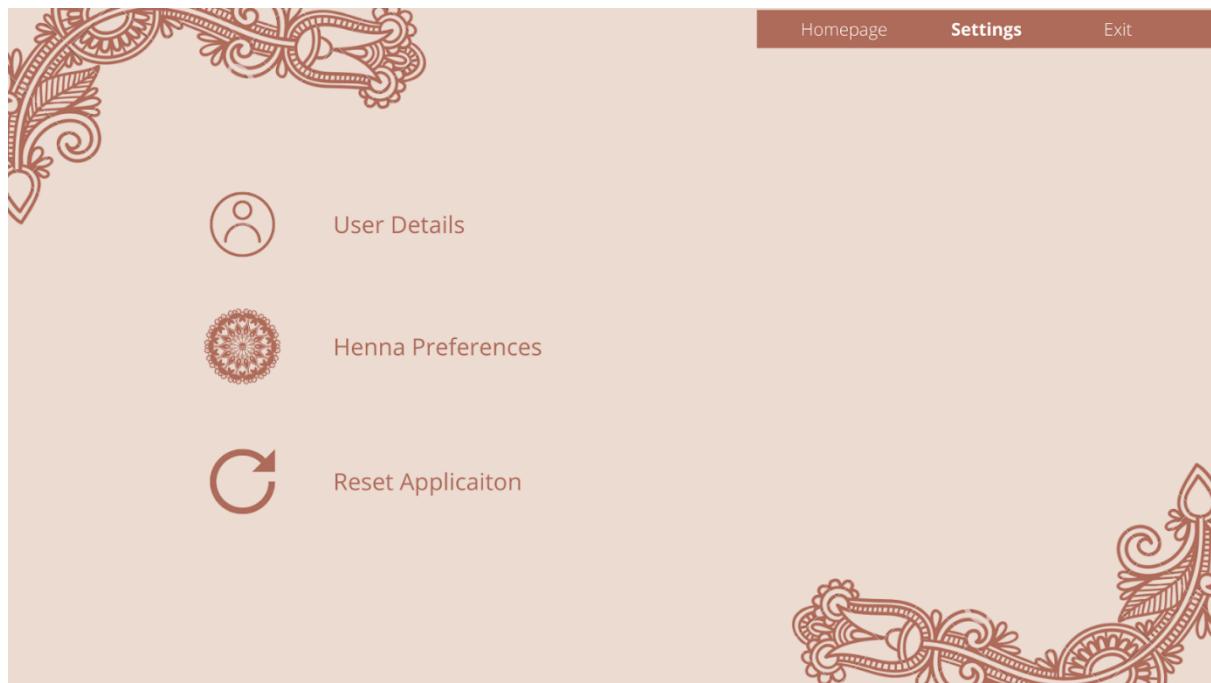
User Feedback:

“The two hands seem unnecessary, especially as they’re likely to show the same design. It would make it easier to focus if you only showed one hand and let us pick which one.”

After considering this input I decided my canvas would only display one default hand, this makes more sense as users have the ability to download the image and then flip it for the other hand. When I explained this to my users, they seemed happy with this idea.

Screen 3 – Settings

Exhibit 25: Settings Prototype



This is the setting screen. Here the user is able to set certain preferences and further customise the application to them.

User Feedback:

"There are not many things here, it might just be easier to include all of these in the homepage making it easier to navigate the app"

After hearing this user feedback, I decided it was a better idea for the setting to be included in the menu bar. This would reduce the application to a single screen, making it easier for those less technologically aware to use and navigate.

Links to Success Criteria

- An easy-to-use interface with limited typing
- Menu bar
- My application has a gender-neutral interface
- Lightweight Design
- Colour's pallet chosen in easy on the eye

Accessibility

As my application needs to be suitable for people of all ages (excluding kids), it must be easy to use. This is especially for older users, as they often have deteriorated vision.

Colour Scheme :

In my success criteria I have stated that the “colour pallet chosen [should be] easy on the eye”, = Hence I picked the following colours :

- Rust Orange
- Pastel Brown
- White

These colours all have a high contrast allowing text and symbols to be easily visible.

The colour scheme also has a close connection to my app and clearly represents its purpose. The shades of brown combined with a neutral white have a close resemblance to henna on the skin, I believe it resonates well with the theme my application.

The colours chosen are not stereotypically linked to a particular gender, making my application inclusive for everyone to use.

Layout:

I have opted for symbols wherever possible as these are intuitive and easier to use. They also lead to a less clutter interface, creating a more minimalist and aesthetic design. This is especially helpful for people with certain disorders, such as OCD and dyslexia, which was apparent in my market research.

There are also consistent border henna themed elements and font formatting to give the application a uniform feel.

Menu :

My application will have a menu bar at the top to easily navigate between pages. This will improve and streamline the user journey .

Success Criteria Fulfilled

- Colour pallet chosen in easy on the eye
- My application has a gender-neutral interface
- My application should be age inclusive

Algorithms

Generating the Design

hand outline

Loading the Hand

External:

1. Store the hand outline as a bitmap image in program folder (more maintainable)
2. Photoshop colours to match colour scheme

In Delphi:

```
PROCEDURE loadBitmap  
    hennaCanvas.image = hand.bmp  
END PROCEDURE
```

Reloading the hand

```
PROCEDURE clearCanvas  
    hennaCanvas.clear  
    loadBitmap  
END PROCEDURE
```

Henna elements

The elements chosen below are the ones which Neha, a professional henna artist, claims to be the most essential to the art for the following reasons:

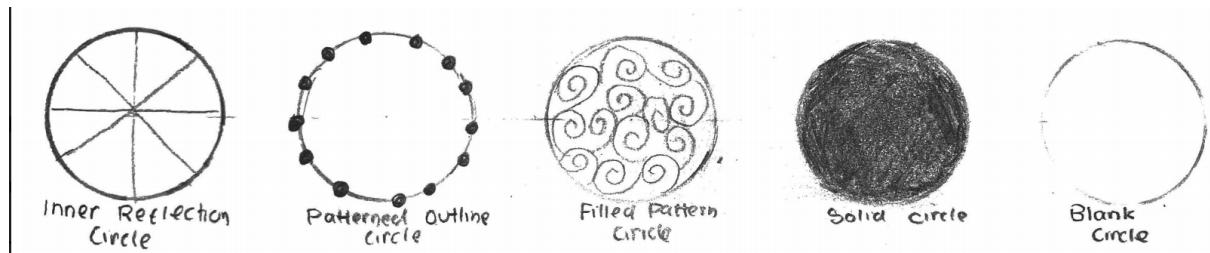
- They are present in the majority of designs
- They form the basis of most complex elements

Therefore using this experience I decided my application should generate designs primarily consisting of the elements listed before.

Lines

| | Line Type | Pseudocode |
|-----------------|--|--|
| Simple Line | Simple Line | <pre>FUNCTION simpleLine(startX, startY, endX, endY) PenMoveTo(startX, startY) drawLineTo(endX, endY) END FUNCTION</pre> |
| Leaf Line | Leaf Line / Flower / Dot Line / Spiral line/ Bump / Petal Line | <pre>FUNCTION ElementLine(startX, startY, endX, endY) SimpleLine(startX, startY, endX, endY) selectElement x = startX y = startY WHILE x < endX AND y < endY drawElement at x and y x = startX + xInterval y = startY + yInterval END WHILE END FUNCTION</pre> |
| Spiral Line | | |
| Bump/Petal Line | | |

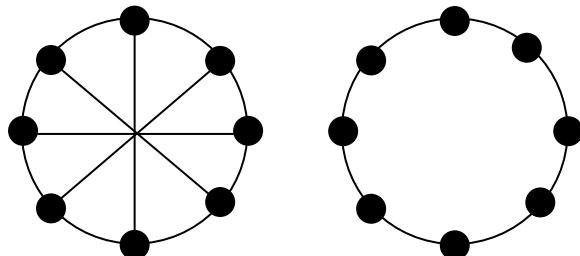
Circles



| Circle Type | Pseudocode |
|---------------|---|
| | <p>$x1,y1$ are the coordinates of the top left corner and $x2,y2$ are the coordinates of the bottom right corner of a square drawn around the circle</p> |
| Blank Circle | <pre>FUNCTION blankCircle(x1,y1,x2,y2) Ellipse(int X1, int Y1, int X2, int Y2) END FUNCTION</pre> |
| Dotted Circle | <pre>FUNCTION innerReflectionCircle(x1,y1,x2,y2) blankCircle(x1,y1,x2,y2) circleCentres[0..7] = eightCirclePoints(x1,y1,x2,y2) FOR i = 0 to (circleCentres.length)-1 //will still work in case the number of points is changed tempCentre.x = circleCentres[i].xCord tempCentre.y= circleCentres[i].yCord solidCircle((tempCentre.x-2),(tempCentre.y+2),(tempCentre.x + 2),(tempCentre.y - 2)) //gives a circle with centre at one of the previously calculated 8 points and a radius of $2\sqrt{2}$ NEXT i END FUNCTION</pre> |
| Filled Circle | <pre>FUNCTION filledCircle(x1,y1,x2,y2) blankCircle(x1,y1,x2,y2)</pre> |
| Solid Circle | <pre>FUNCTION solidCircle(x1,y1,x2,y2) blankCircle(x1,y1,x2,y2) Floodfill END FUNCTION</pre> |

Inner reflection circle

```
FUNCTION innerReflectionCircle(x1,y1,x2,y2)
    blankCircle(x1,y1,x2,y2)
    circleCentres[0..7] = eightCirclePoints(x1,y1,x2,y2)
    FOR i = 0 to (circleCentres.length/2)-1 //will still work in case the number of points is changed, gives half of the points
        line(circleCentres[i].xCord,circleCentres[i].yCord,circleCentres[i+3].xCord,circleCentres[i+3].yCord)
        //has parameters (startX,startY,endX,endY)
        //want line drawn between a centre and 3 centres after (e.g. 0 connects to 4) as they will always be opposite
    NEXT i
END FUNCTION
```



Both the circle dots and inner reflection lines require 8 coordinates on the border symmetrically on the circle border, hence it makes sense to write a common procedure to calculate these coordinates.

```
FUNCTION     eightCirclePoints(topLeft.XCord,topLeft.YCord,      bottomRight.XCord,      bottomRight.YCord,
byRef:array[0..7] of typeCoordinates)

//Pass in the coordinates of the top left and bottom right coordinate of an imaginary square drawn around the circle

//The 0 indexed item is directly above the centre, then they are counted clockwise from there
centre.XCord =( topLeft.XCord + bottomRight.XCord)/2
centre.YCord =( topLeft.YCord + bottomRight.YCord)/2
array[0].XCord =centre.XCord //same as center as it needs to be in line
```

```

array[0].yCord = topLeft.yCord //same as top of rectangle

array[1].xCord =(centre.xCord + bottomRight.xCord)/2 //halfway between centre and right most x coordinates

array[1].yCord = (centre.yCord + topLeft.YCord)/2 //halfway between centre and top y coordinates

array[2].xCord = bottomRight.xCord //same as most right x coordinate

array[2].yCord = centre.yCord //same as center as it needs to be in line

array[3].xCord =(centre.xCord + bottomRight.xCord)/2 //half way between centre and right most x coordinates

array[3].yCord = (centre.yCord +bottomRight.YCord)/2 //half way between centre and bottom y coordinates

array[4].xCord = centre.xCord //same as center as it needs to be in line

array[4].yCord = bottomRight.yCord //needs to be at bottom

array[5].xCord =(centre.xCord + topLeft.xCord)/2 //half way between centre and left most x coordinates

array[5].yCord = (centre.yCord +bottomRight.YCord)/2 //half way between centre and bottom y coordinates

array[6].xCord = topLeft.xCord //same as most left x coordinate

array[6].yCord = bottomRight.yCord //needs to be at bottom

array[7].xCord =(centre.xCord + topLeft.xCord)/2 //half way between centre and left most x coordinates

array[7].yCord = (centre.yCord +topLeft.YCord)/2 //half way between centre and top y coordinates

RETURN array[]

END FUNCTION

```

Triangle

```

PROCEDURE drawTriangle(x1,y1,x2,y2,x3,y3)

drawLine(x1,y1,x2,y2)

drawLine(x1,y1,x3,y3)

drawLine(x2,y2,x3,y3)

```

```
END PROCEDURE
```

Stroke

```
FUNCTION selectStroke  
selectedNum = randomNumber(1,numbOfStyles)  
  
Case selectedNum of  
1:selectedStyle = solidStroke  
2:selectedStyle = dashedStroke  
...  
return selectedStyle  
  
END FUNCTION
```

Fill

```
FUNCTION selectFill  
selectedNum = randomNumber(1,numbOfFills)  
  
Case selectedNum of  
1:selectedStyle = solidFill  
2:selectedStyle = hashedFill  
...  
return selectedStyle  
  
END FUNCTION
```

Templates

To decide what templates my designs should be based around, I once again talked to Neha. She said the following were the most important principles in henna:

- Symmetry
- Geometry
- Sectioning of the hand
- Placement

I also asked Neha to help with the style each template fit into, as this requires many years of experience.

Template 1

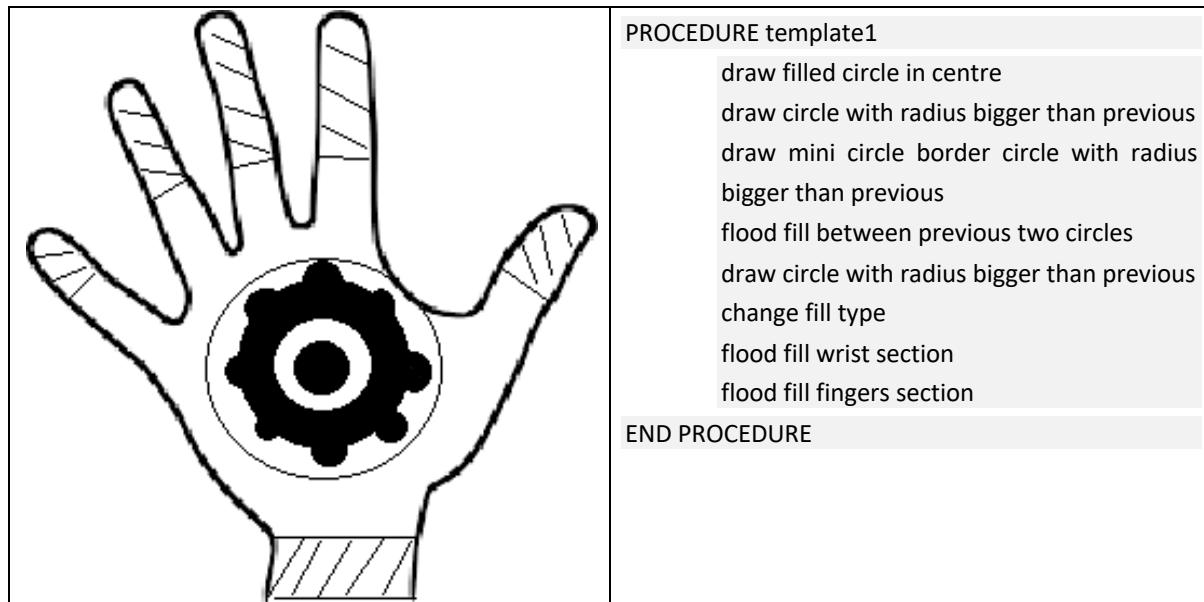
Link(s) to success criteria:

- My application should have designs which are inclusive of all the major cultures (which consider henna important)

Purpose

This template will focus on the principles of symmetry and section of the hand.

Algorithm



Styles

- South Asian
- Middle Eastern

Randomisation

| Randomisation | Pseudocode/Plan |
|-----------------------------------|---|
| Select elements to be draw | Add ifDraw = true around each section (all individual circles, wrist block, finger block) |
| Wrist Fill | Randomise the brush stroke before flood filling the wrist |
| Finger Fill | Randomise the brush stroke before flood filling the finger section |
| Circles | Randomise pen type before drawing |

Template 2

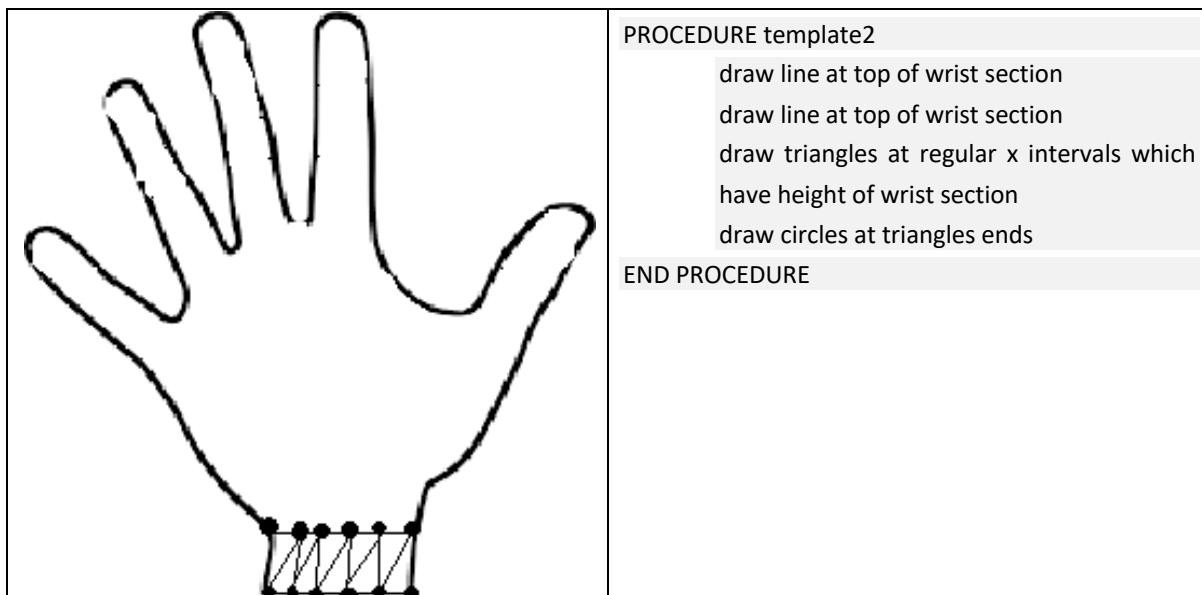
Link(s) to success criteria:

- My application should have designs which are inclusive of all the major cultures (which consider henna important)

Purpose

This template will focus on the principle of geometry.

Algorithm



Styles

- South Asian
- Middle Eastern

Randomisation

| Randomisation | Pseudocode/Plan |
|----------------------------|--|
| Select elements to be draw | Add ifDraw = true around each section (wrist borders, circles , triangles) |
| Circles | Randomise filled or solid |

Template 3

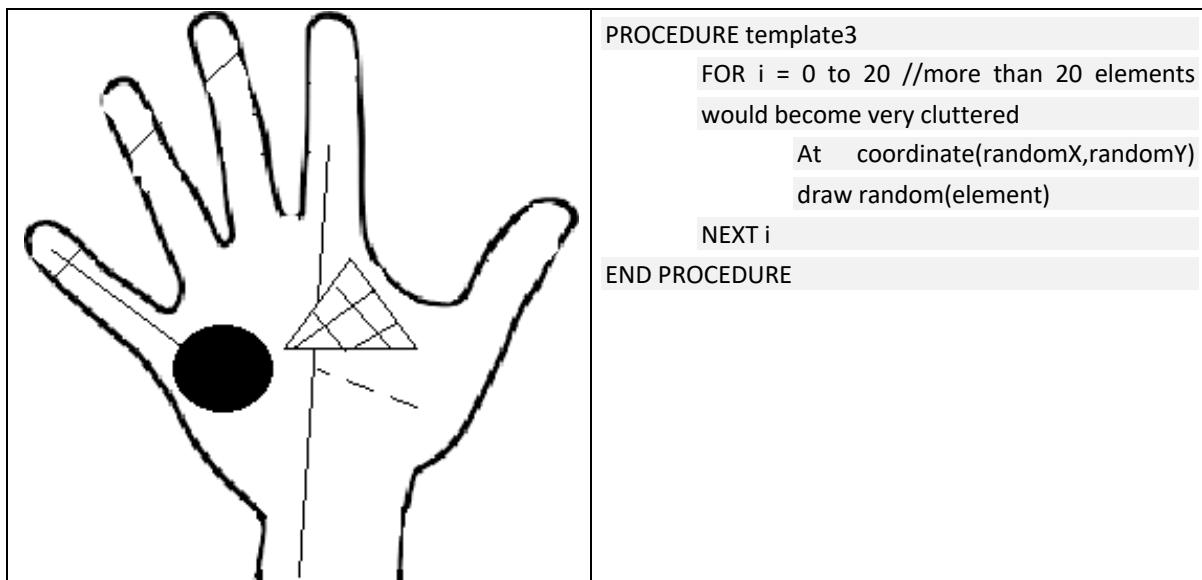
Link(s) to success criteria:

- My application should have designs which are inclusive of all the major cultures (which consider henna important)

Purpose

This template will focus on the principle of placement.

Algorithm



Styles

- North African
- Middle Eastern

Randomisation

Note : This algorithm already contains element and placement randomisation.

| Randomisation | Pseudocode/Plan |
|-------------------|--|
| Select pen type | Randomise the pen stroke before drawing each element – provides a different shape border |
| Select brush type | Randomise the brush stroke before drawing each element – provides a different shape fill |

If Draw function

ifDraw is a subroutine I will include in my program. It will randomly pick whether or not to draw an element, factoring in the elements weighting when deciding if it should be drawn or not. The algorithm for it is show below:

```
FUNCTION ifDraw : Boolean
    randomValue = randomNumber(0,100+weighting)
    IF randomValue <100 then
        RETURN false
    ELSE IF
        RETURN true
    END IF
END FUNCTION
```

Generating a New Template

Link to success criteria:

- The program generates henna designs
- Designs being randomised each time

Overview of steps needed to generate a new template:

1. Clear Canvas (procedure in loading hand section)
2. Reset Variables
3. Pick new template and draw
4. Display Elements used

```
PROCEDURE resetVariables
```

```
    elementsLISp = -1;  
  
    elementsTop = -1;  
  
    for i = 0 to 30  
  
        elementsLI[index].pointer = -1;  
  
        elementsLI[index].item = "";
```

```
END PROCEDURE
```

```
PROCEDURE pickNewTemplate
```

```
for i = 1 to 3  
  
    stylesTable.recno = i  
  
    styleData[i].lowerBound = totalWeight  
  
    totalWeight = totalWeight + stylesTable["fieldValue = weighting"]  
  
    styleData[i].upperBound = totalWeight  
  
next i  
  
tempNum = randomNumber(0,totalWeight)  
  
If tempNum <= styleData[1].upperBound then  
  
    stylesTable.recno = 1  
  
ELSE IF tempNum <= styleData[2].upperBound
```

```

styleTable.recno = 2

ELSE

    styleTable.recno = 3

END IF

chooseStyle = stylesTable["fieldValue = weighting"]

templateFound = false

WHILE templateFound = false

    recNo = randomNum(1,templatesTable.numOfRecords)

    IF templatesTable["fieldValue = chooseStyle "] = true THEN

        templateFound = true

    END IF

END WHILE

callProcedure("Template"+recNo)

END PROCEDURE

PROCEDURE displayElements

    createLabels

    populateLabels

    createButtons

    populateButtons

    createImageComponents

    populateImageComponents

END PROCEDURE

```

Saving templates

Link(s) to success criteria : Ability to save the designs in an image format

ON save button.click

Call windows in built dialogue to choose location to save
save canvas.image to this location

END ON

Recommendations

Link(s) to success criteria :

- Designs being specialised to a user
- The ability to like or dislike each design
- Option to reset preferences

Collecting user data

- Initial preferences
 - Design a database containing all elements
 - Fill in standard, constant database data (e.g. element name)
- Settings
 - Option to reset database

```
PROCEDURE onResetButtonClick  
    Set all weighting to initial values  
END PROCEDURE
```

- Likes and dislikes
 - When a design is liked/disliked in the table this needs to be registered
 - Appropriate data processing needs to be triggered

In order to improve the designs the user receives, it is important that the program keeps track of which patterns the user has liked/disliked, analyse this and use it to increase/decrease the frequency of the elements and pattern styles present.

Overview of Algorithm to achieve this:

1. User likes/dislikes a pattern
2. If the weighting for that element does not equal zero
3. The weighting for every element presents in that pattern increased/decreased by one
4. If the score for that category does not equal zero
5. Total score for that design category changed
6. The elements and style of the design are more /less likely to be used (as the weighting is considered when picking the design)
7. The user will see certain elements and templates more often hence making it specific to their taste

Subroutines needed:

```
PROCEDURE onThumbsUpClick
```

```

FOR all elements in table

    IF currentElement.weight ≠ maxWeight //needs to be read in from the
        database

        Call procedure changeWeighting(currentElement, 1.1)

    END IF

    NEXT element

    FOR all categories in table

        IF currentCategory.weight ≠ maxWeight //needs to be read in from the
            database

            Call procedure changeWeighting(currentCategory ,1.1)

        END IF

        NEXT category

    END PROCEDURE

PROCEDURE onThumbsDownClick

    FOR all elements in table

        IF currentElement.weight ≠ 0 //needs to be read in from the database

            Call procedure changeWeighting(currentElement , 1.1)

        END IF

        NEXT element

        FOR all categories in table

            IF currentCategory.weight ≠ 0 //needs to be read in from the database

                Call procedure changeWeighting(currentCategory, 1.1)

            END IF

            NEXT category

        END PROCEDURE

```

```
PROCEDURE changeWeighting(attribute, changingValue)

    do with tableContaining(attribute)

        for i: 1 to numbRecords do

            begin

                recno:=i;

                if fieldvalues[fieldContaining(attribute)] = attribute then

                    attribute.value = attribute.value*changingValue

                end if

            end

    END PROCEDURE
```

Henna Tutorials Via YouTube

Link(s) to success criteria: Elements being linked to YouTube tutorials

Overview:

1. Elements used in the design are displayed on the page
2. The element is clicked
3. Query information retrieved from database
4. YouTube API called and the search page with all the video tutorials appears on screen

Displaying used elements

WHEN designIsBeingGenerated

 FOR every element drawn

 Add element to linkedList

 NEXT element

END WHEN

PROCEDURE displaySetElements(byVal:set,xCord,yCord) //xCord gives xCord of where the label should start from, yCord gives the yCord of where the first label and canvas should be places

 For i = 1 to nextItemInlinkedList

 labelArray.label.caption = linkedList[i].item

 End For

END PROCEDURE

PROCEDURE apiCallButtons

 INITIALIZE buttonsArray[0..set.length-1] of type labels

 FOR i = 0 to [set.length-1]

 create button with name (set[0].lbl) at (xCord,yCord)

```
buttonsArray[0..set.length-1] = set[0]

button.click = APICall

create canvas with name (set[0]_canvas) at (xCord + 20,yCord)

get necessary procedures from database

call procedures in order to draw element on canvas called (set[0]_canvas)
```

```
NEXT i
```

```
END PROCEDURE
```

On button click

```
IMPORT library ShellAPI
```

```
PROCEDURE APICall
```

```
Call Youtube API("henna + Tutorials + " + corresponding label ) in browser
```

```
END PROCEDURE
```

Data Structures

Linked List

A linked list is a dynamic data structure, used to hold an ordered sequence of data. I will implement one in my application in order to keep alphabetical track of the elements used in the template created. A linked list is suitable for this as it is designed to have items added to the middle of it, allowing it to stay in order.

To implement this I will use an array of **TLinkedListItem**, a custom user defined type consisting of pointer and data item. Whilst this could be done as a 2D array, I believe this method makes more sense, as it will allow pointer and data item to be accessed by this name, thus not relying on having to remember which column stores which piece of information. Also you could only store data of a single data in a 2D array, hence it would make the process more complicated as the record number of the item would have to be stored and obtained from the database.

The array storing items will have a size of 31 items, indexes from [0..30]. This is a reasonable size as when I talked to Neha she said beginner patterns usually have 15 elements as maximum.

An Example is shown beneath for clarity:

| Index | TLinkedListItem | |
|-------|------------------------|---------------------------|
| | Item as String | Pointer as Integer |
| 0 | basicLine | 2 |
| 1 | Spiral | -1 |
| 2 | filledCircle | 1 |
| ... | ... | ... |

Note: -1 is used as a null pointer, it will tell us we are at the end of our linked list

```
PROCEDURE addToLinkedList(item, by ref linkedList, by ref lastIndexUsed, by ref SP) //data related to LL
pasted by reference as this actual data may need to be modified

IF SP = -1 then

    lastIndexUsed += 1 // += increments current variable value by one

    linkedList[lastIndexUsed].item = item

    SP = lastIndexUsed

    linkedList[lastIndexUsed].pointer = -1

ELSE IF linkedList[SP].item = item then

    addItem = false;
```

```

ELSE IF linkedList[SP].item > item then

    lastIndexUsed = lastIndexUsed +1

    linkedList[lastIndexUsed].item = item

    tempIndex = SP

    SP = lastIndexUsed

    linkedList[lastIndexUsed].pointer = tempIndex

ELSE

    placeFound = false

    addItem = true

    p = SP


WHILE (placeFound = false) AND (linkedList[p].pointer <> -1) AND(addItem = true) do

    IF item = linkedList[linkedList[p].pointer].item then

        addItem = false

    ELSE IF item > linkedList[linkedList[p].pointer].item then

        p = linkedList[p].pointer

    ELSE

        placeFound = true

    END IF

    IF addItem = true then

        lastIndexUsed = lastIndexUsed +1

        linkedList[lastIndexUsed].item = item

        tempIndex = linkedList[p].pointer

        linkedList[p].pointer = lastIndexUsed

        linkedList[lastIndexUsed].pointer = tempIndex

    END IF

```

```
END PROCEDURE
```

```
PROCEDURE clearLinkedList(length , by ref LL)
```

```
    elementsLISP = -1
```

```
    FOR index = 0 to length do
```

```
        elementsLL[index].pointer =-1
```

```
        elementsLL[index].item = null
```

```
    NEXT index
```

```
END PROCEDURE
```

Database

In order to store the details about the elements, templates and style preferences about the user I will set up a database in Access as it is compatible with Delphi. It is also a widely used software, hence is well tested and has many advanced functionalities which may be useful in creating my program.

Database Design

Elements

| Element name* | API Query | Type | User Rating | Image |
|---------------|-----------|------|-------------|-------|
| | | | | |
| | | | | |

- The Element name - String
 - This is the primary key as it allows a unique identifier for the recognition of every element
- API Query – String
 - This contains the data which needs to be used in the YouTube API call
- Type – String
 - This will be used in randomising the elements
 - Categorises the elements
- User rating – Integer
 - Gives you quantitative data about the user's opinion on that element
- Image
 - Stores what the element looks like

Templates

| Template ID* | Template name | South Asian Weight | North African Weight | Middle Eastern Weight |
|--------------|---------------|--------------------|----------------------|-----------------------|
| | | | | |
| | | | | |

- Template Id – Integer
 - Primary key
 - Allows a unique identifier for the recognition of every template
- Template Name - String
 - Make it recognizable to the person entering the data into the database
- Weightings – Integer
 - Shows how closely related each element is to the particular category

Style weighting

| Category | User_Weight |
|----------|-------------|
| | |

| | |
|----------------|--|
| Middle Eastern | |
| North African | |
| South Asian | |

- User_Weight - Integer
 - Stored the individual preference of the user's style types
- Altered when a user likes or dislikes a pattern

Linking the Database to Delphi

The following Delphi components will be needed to link the program with the database:

- TADOConnection
 - This will link the file to the database
- TADOTable
 - Will get the data from a particular database table
- TADOQuery
 - Will allow SQL to be written to query the database

2D array

styleInfo as an array[1..3,0..2] containing integers

This array is needed when selecting the next template to be drawn to store the style name and range of values

Diagram

| styleName | lowerBound | upperBound |
|------------------|-------------------|-------------------|
| | | |
| | | |
| | | |

Data Types

User Defined Record Types

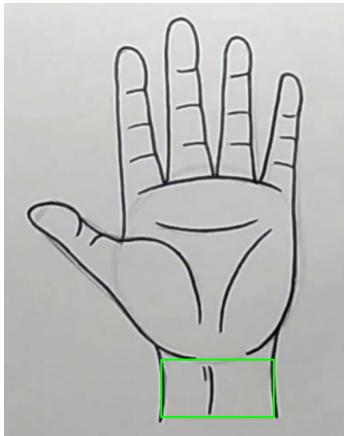
Coordinates

This will be used to store pairs of coordinates.

```
RECORD coordinates
  xCord : Integer
  yCord : Integer
END RECORD
```

TCooridantesBlock

To store “blocks” of coordinates. For example for the wrist henna section as illustrated below



```
RECORD TCooridantesBlock
  topLeft : TCooridantes
  topRight : TCooridantes
  bottomLeft : TCooridantes
  bottomRight: TCooridante;
END RECORD
```

TLinkedListItem

To store items in the linked list,

```
RECORD TLinkedListItem
  item : string ;
  pointer : integer;
END RECORD
```

TElementsDisplay

To display the items used in the template to the user on the page.

```
RECORD TElementsDisplay
  labels : TLabel
  button : TButton
  image : TImage
END RECORD
```

Key Variables

| Variable Name | Type | Why | Initial Value |
|-------------------|----------------------------------|---|--|
| handCentre | TCooridantes | Stores the coordinates of the centre of the hand - which is commonly used for many templates | handCentre.xCord = 325 handCentre.yCord = 550 |
| wristMaxPoints | TCooridantesBlock | Stores the coordinates of the section of the wrist which henna is usually done on | wristMaxPoints.topLeft.xCord = 212 wristMaxPoints.topLeft.yCord = 800 wristMaxPoints.bottomLeft.xCord = 212 wristMaxPoints.bottomLeft.yCord = 900 wristMaxPoints.topRight.xCord = 423 wristMaxPoints.topRight.yCord = 800 wristMaxPoints.bottomRight.xCord = 420 wristMaxPoints.bottomRight.yCord = 900 |
| fingerMaxPoints | Array[0..4] of TCooridantesBlock | Stores the coordinates of the sections of the fingers which henna is usually done on | fill in |
| defaultBrushStyle | TBrushStyle | Stored the default brush style, which the brush style will need to be changed back to after any changed to it | bsClear |
| defaultPenStyle | TPenStyle | Stored the default brush style, which the pen style will need to be changed back to after any changed to it | psSolid |
| elementsLabels | Array[0..30] of TLabel | An array of 31 display labels to display elements used in the template | N/A |

| | | | |
|-----------------|---------------------------------|---|---|
| elementsLI | Array[0..30] of TLinkedListItem | An linked list used to store the elements used alphabetically , implemented as an array | elementsLI[all].item = “ elementsLI[all].pointer =-1 |
| elementsLISp | Integer | Stored the start pointer of the linked list | -1 |
| elementsTop | Integer | Stores the last index used of the elementsLI | -1 |
| currentTemplate | Integer | Stored the number of the current template used | N/A |

Inputs and outputs

Table

The following table shows all of the inputs into the application, the type of input, the process that should occur and the expected output.

| Input | Type | Process | Output |
|-----------------------|-----------------|--|---|
| Like Design | Button | Changing the element and style weighting algorithms with a multiplier of 1.1 | The weights of the elements used are multiplied by 1.1 and the new values are stored in the database The weight of the style used is multiplied by 1.1 and the new value is stored in the database |
| Dislike Design Button | Button | Changing the element and style weighting algorithms with a multiplier of 0.9 | The weights of the elements used are multiplied by 0.9 and the new values are stored in the database The weight of the style used is multiplied by 0.9 and the new value is stored in the database |
| New Design | Button | Canvas clear algorithm Variable reset algorithm New template algorithm Elements display algorithm | A new template is drawn , the elements used in it are displayed in labels and corresponding buttons which take the user to the appropriate Youtube API are created |
| Save Design | Button | Call in built pop up which allows user to select file location in which to save canvas Save canvas contents to user selected location | The template is saved to the computer in the location specified by the user |
| YouTube tutorial | Button | Call YouTube API with “henna + tutorials +’ element name” | Youtube search page is opened with element name and henna tutorial being queried |
| Reset Preferences | Menu Bar Option | Go through all record in database and reset all weighting in database to initial values | All database weighting values set to initial values |

| | | | |
|------|-----------------|---|-------------------------------------|
| Exit | Menu Bar Option | Call Delphi in built procedure to close the application | Application shut on user's computer |
|------|-----------------|---|-------------------------------------|

Validation

As there is very limited user input in my program, there does not need to be excessive validation.

There are two typed of use inputs in my program:

- Buttons
- Menu bar selection

Both of these inputs act in the same way, when they are clicked/selected, the corresponding procedure is triggered. Hence as they are commonly used inbuilt components into Delphi, they are pretested and should be error free (or at least have no major error). As they cannot pass data to any variables, there is no risk of the users sending erroneous data into the program, thus no validation is required.

Testing

Development and Testing Method

Development

My development will take an iterative approach. As outlined earlier in the design section of this document, I have decomposed the problem down into smaller parts and planned these. Thus my program will consist of the following stages:

- Stage 1 – Loading Screen
- Stage 2- Homepage interface
- Stage 3 – Elements
- Stage 4 – Basic Templates
- Stage 5 – Linked List
- Stage 6 – Database Setup
- Stage 7 – Displaying Elements
- Stage 8 – Linking dataset with code
- Stage 9 – YouTube API
- Stage 10 – Saving Canvas
- Stage 11 – Linking and disliking designs
- Stage 12 – New Design
- Stage 13 – Tidying up the code
- Stage 14 – Randomisation and Complex Templates
- Stage 15 – More menu bar options
- Stage 16 – Final Testing
- Stage 17 – Final Stakeholder Feedback

After each stage I will have a review. The review will consist of user testing, feedback and improvements.

Some stages may still rely on heuristic approximations, which are improved by trial and error due to the visual nature of this application. For example when trying to find the centre co-ordinate of the hand, it maybe easiest to get a dot and place it at a random location in the canvas. Then iterate through with a better approximation of the location, until you find a satisfactory one. This maybe assessed on a visual basis or using stakeholder feedback.

Testing

My testing will take a mix of approaches. Due to the visual nature of my application, unlike most data driven application, it is easier and more accurate to test the algorithms during implementation. Hence I have laid out my testing plan below:

Module tests

Every procedure coded should be module tested, to ensure it works in isolation. Hence if any errors occur later on in the program it will be easier to debug them, as it is likely to be a logic error in calling the procedure one after another.

Black Box testing

The inputs listed earlier in development should be compared to the expected outputs. Evidence should be provided of the results produced.

Functionality Testing

My application should match the test criteria, with evidence provided, to ensure it is fully functional and working.

User / Stakeholder Testing

User testing and feedback is critical in this project as it is a design based application ; whilst something may work on a technical basis it may not appear aesthetic or correct. Thus after each stage of development I should consult my stakeholders or other users and consider their feedback.

Collecting Evidence

This applied mainly for module tests

Henna Design Generation

| To be tested | How to collect evidence |
|--------------------------------------|---|
| Loading hand onto the canvas | Screenshot and code evidence |
| Henna elements | Comparison of visual representation and produced element |
| Element's variation | Ensure each element is at least displayed once |
| Templates working | Comparison of visual representation and produced template |
| Templates acceptable as henna design | Ask stakeholders and record answers |
| Saving templates | Screenshots and code evidence |

Design Recommendations

| To be tested | How to collect evidence |
|--------------------|---|
| Resting database | Screenshot of database values and code evidence |
| Likes and disliked | Screenshot of database values compared to expected values |

YouTube Videos

| To be tested | How to collect evidence |
|-------------------------|--|
| API Query Keywords | Query written in YouTube page matches expected from database |
| Calling the YouTube API | Screenshot of YouTube page |
| Elements displayed | Screenshot of displayed vs expected |

Linked List

| To be tested | How to collect evidence |
|--------------------|-----------------------------|
| Items stored in LL | Debugger point and data tip |
| LL traversal | Output to a caption |

User Interface

| To be tested | How to collect evidence |
|--------------|-------------------------|
| | |

| | |
|----------------------------------|-------------------------------------|
| Technically correct | Compare result produced with design |
| Aesthetical and visually correct | User feedback |

Functionality Checklist

My code should meet the following checklist to ensure it works properly and has full functionality :

| Action to test | Working? |
|---|----------|
| Does the loading screen automatically appear? | |
| Does the loading screen automatically disappear after a set time? | |
| Does the homepage automatically appear? | |
| Does the new design button produce a new design? | |
| Can you save designs? | |
| Are the elements used displayed? | |
| Can you go to the relevant YouTube pages for the elements? | |
| Are pictures of the isolated elements displayed? | |
| Does the like button increase weightings of styles? | |
| Does the like button increase the weighting of style? | |
| Does the dislike button decrease weightings of styles? | |
| Does the dislike button decrease the weighting of style? | |
| Does the menu bar have a settings option? | |
| Does the menu bar have an exit option? | |
| Is there an option to reset the application? | |

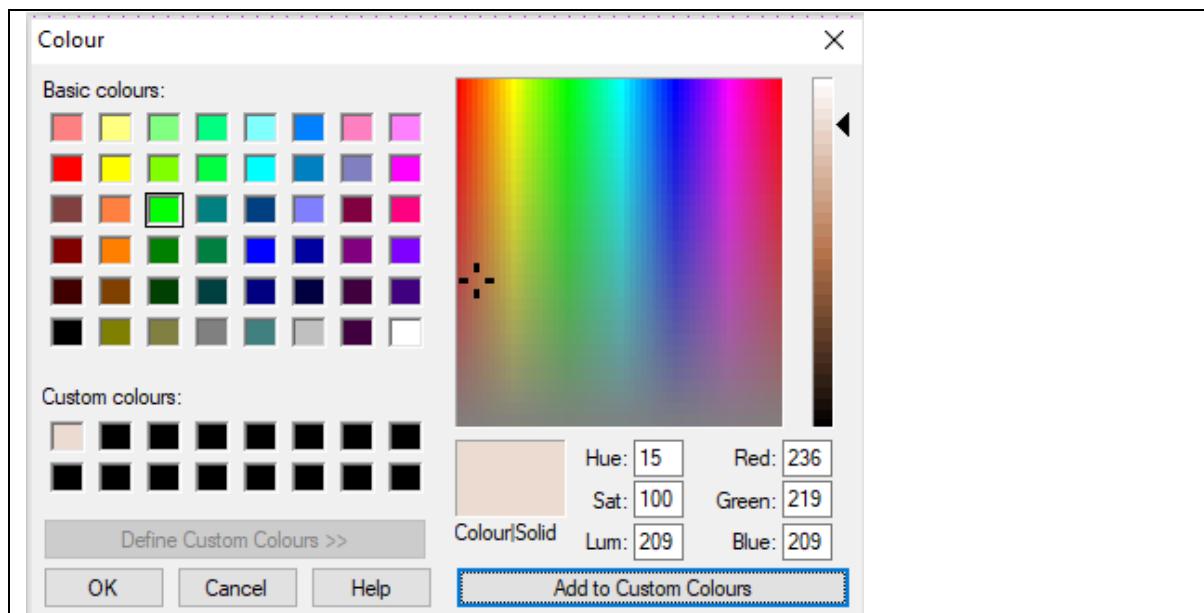
Development and Testing

Stage 1- Loading Screen

Setting up the User Interface

Colours

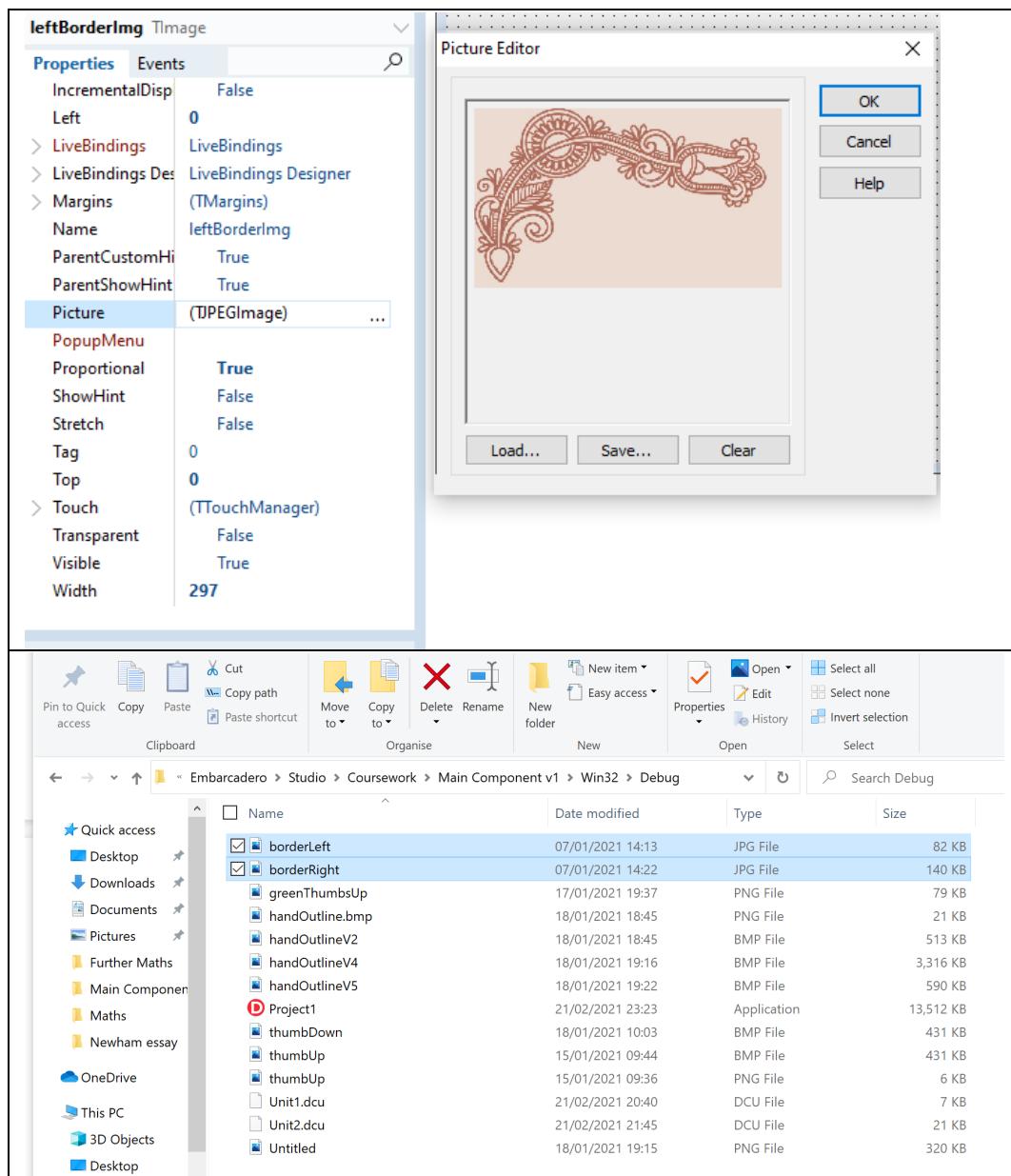
To make my design optimised for the elderly and those with visual impairments, my design uses custom colours which are gentle on the eye and high in contrast. Thus, to find the exact RGB values used in the background of my design, I uploaded the design into an online colour picker (<https://imagecolorpicker.com/>). I then created a custom colour in Delphi by entering these values and adding to custom colours, making it available for use through the program.



Images

According to my design the plan was to load the images for the border of my loading screen though the code on form load. However, this would be unnecessarily complex decreasing its long-term manageability .

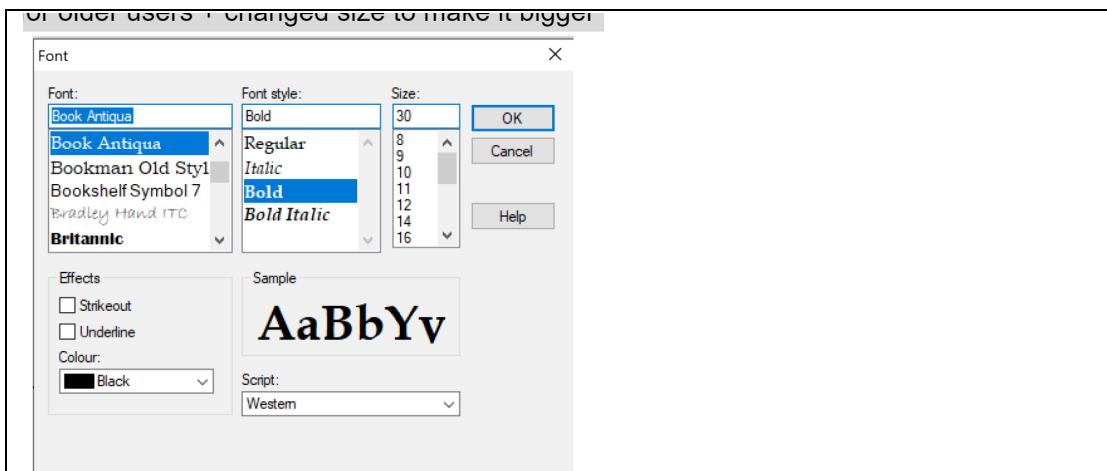
1. I downloaded the border element from the internet and converted them into JPG format.
2. I then duplicated this file, rotated the copy 180 degrees and renamed it.
3. I moved these files into the project folder in order to keep everything together in case I needed to change/edit this in the future.
4. I created two image objects and positioned them in the corners of my form.
5. I changed the picture property of the imaged objects to the appropriate files.



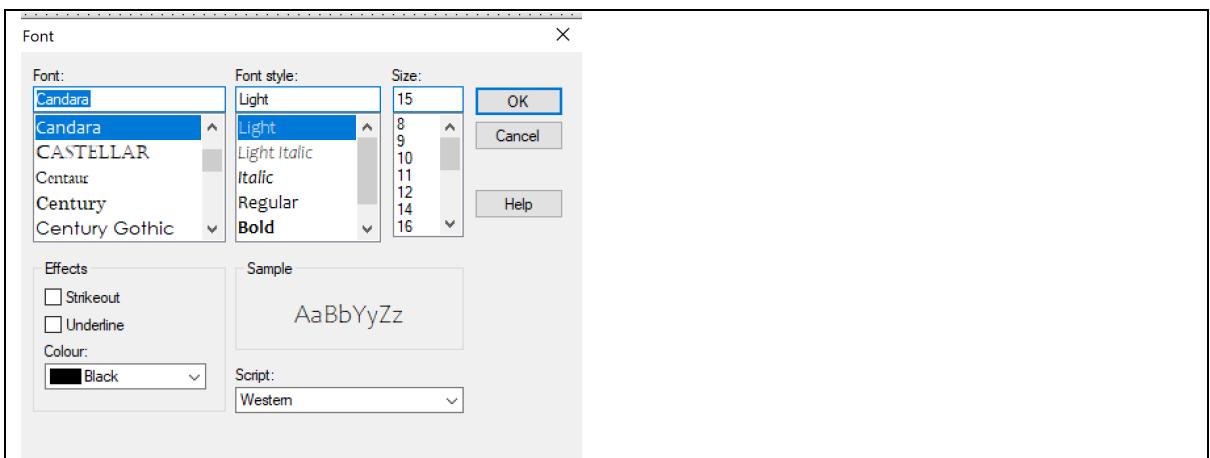
Text

To add the planned text to the loading screen I added label in the appropriate locations and typed the planned text into the corresponding labels. However when I tried to change the fonts to the same type as used in Canva during my design, I realised that Delphi did not support them. Hence I decided to use similar fonts which fulfilled my accessibility criteria which were supported over placing in images as it would make the program more adaptable and flow better throughout, as pictures would not be practical for later parts of my program.

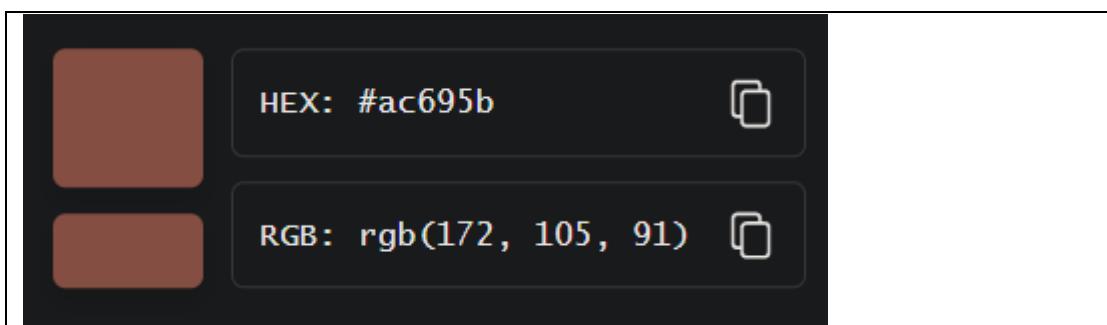
1. I changed the titleLbl (camel case has been used to name the variables) font to Book Antiqua as it was large, thick and easy to read, with more options to emphasise it. I also changed the size to make it bigger and applied bold to make it stand out.



2. I altered the same things in captionLbl, choosing Cassandra this time as it is a lighter font to contain the less important information, hence avoiding my page being cluttered.

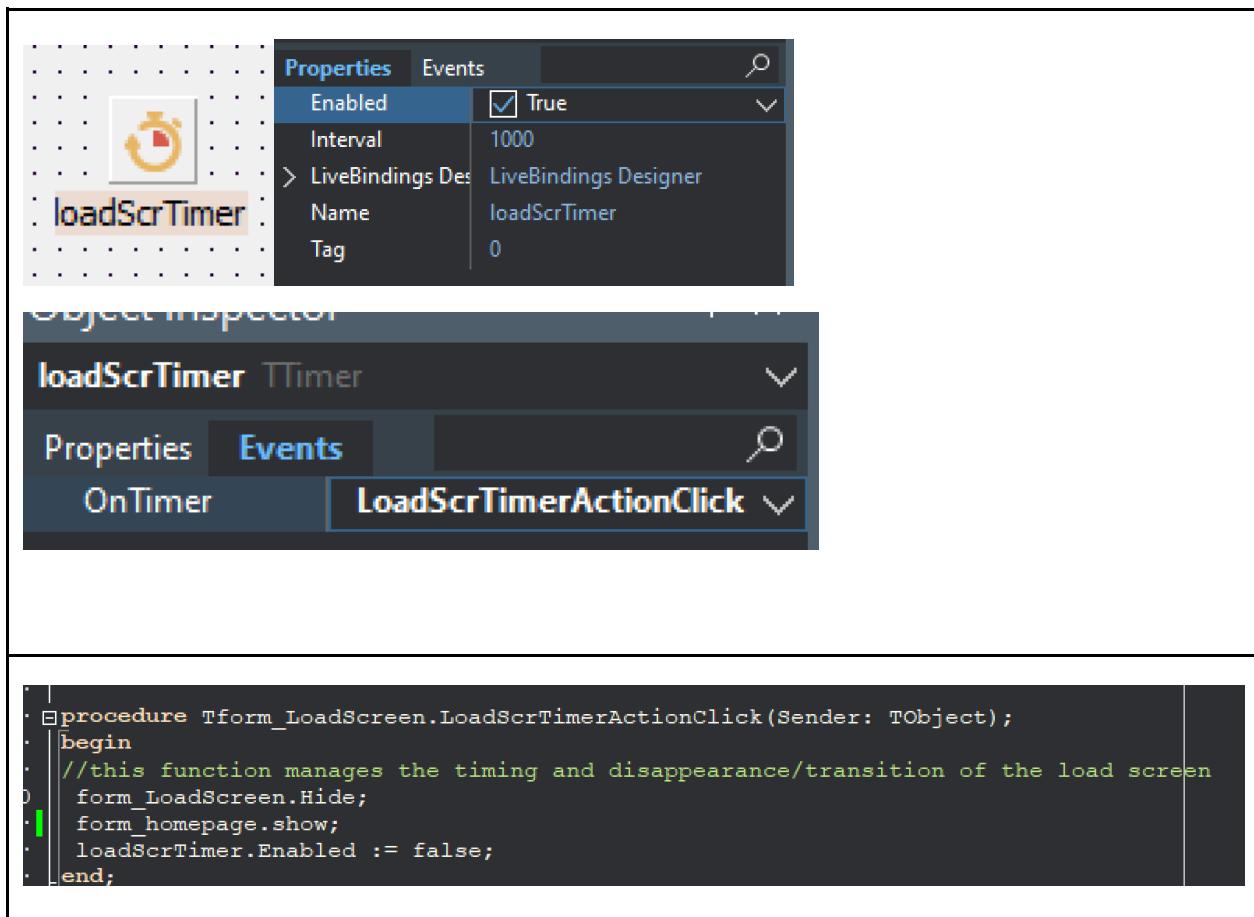


3. As per my design, I needed to change the font colours to the same as that of the border elements to ensure a consistent, professional feel. Hence i used an online colour picked to gain this shade, created a custom colour using this and set the font colour to this (in the same way as done for the background colour).



Adding Interactivity

Timing



The timer is triggered at 1000 seconds. It then calls the procedure “LoadScrTimerActionClick”, which hides form one and shows form two.

Initially there was an error every one second where the homepage kept reloading. Hence I added the line which disables the timer when the action occurs. As the timer is disabled, ensuring it does not repeat these actions again.

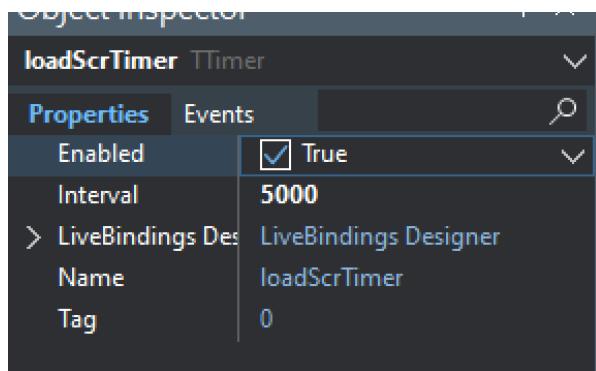
Stage 1 Review

User Feedback and Changes:

When I asked my users for feedback on the loading screen, they passed the following comments:

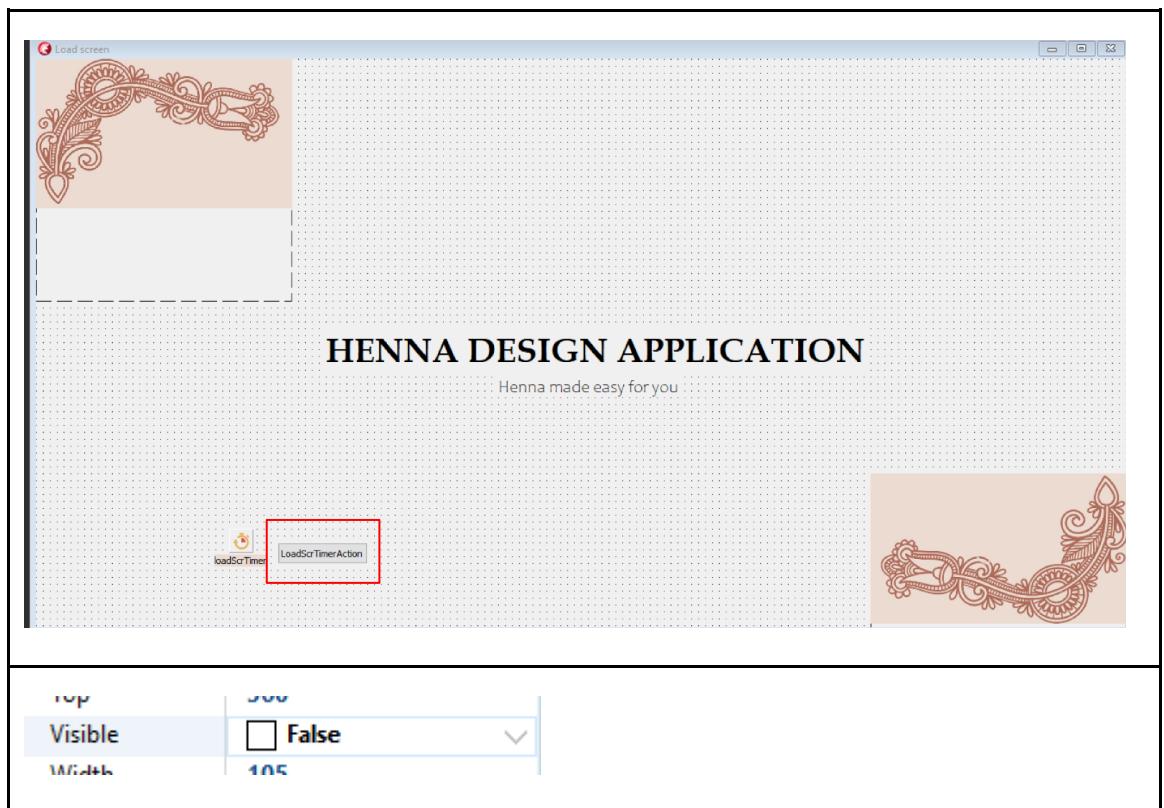
1. “The loading screen is very fast. I barely have time to look at and process the load screen before being taken to the main page.”

To fix this error I changed the interval of the timer to 5000.



I then showed the user the load screen again, and they were much happier with the time it was displayed for.

2. The button ruins the aesthetic of the loading screen - hence initially I changed the property of this element to invisible



However I realised that this button did not need to exist - instead I only required the procedure it executed. Thus I deleted the button and declared and wrote the procedure through code in the program.

```
type
  TForm_LoadScreen = class(TForm)
    titleLbl: TLabel;
    leftBorderImg: TImage;
    rightBorderImg: TImage;
    captionLbl: TLabel;
    loadScrTimer: TTimer;
    procedure FormCreate(Sender: TObject);
    procedure LoadScrTimerActionClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

  end;
```



When I asked the user about the aesthetic of the loading screen now they said “the load screen looks gorgeous now ! It is simplistic and does not strain my eyes but is really pretty”.

What has been done

The loading screen has been completed. When the program starts up the loading screen automatically displays. Then after 5 seconds, it automatically disappears, and the homepage automatically appears.

How has it been tested

Black box testing has occurred.

- The project is compared to what the output should be when the project is loaded
 - As this is the same as what is expected all tests were passed
-

Criteria Being met

- Loading Screen
-

Functionality Checklist

- Does the loading screen automatically appear?
 - Does the loading screen automatically disappear after a set time?
 - Does the homepage automatically appear?
-

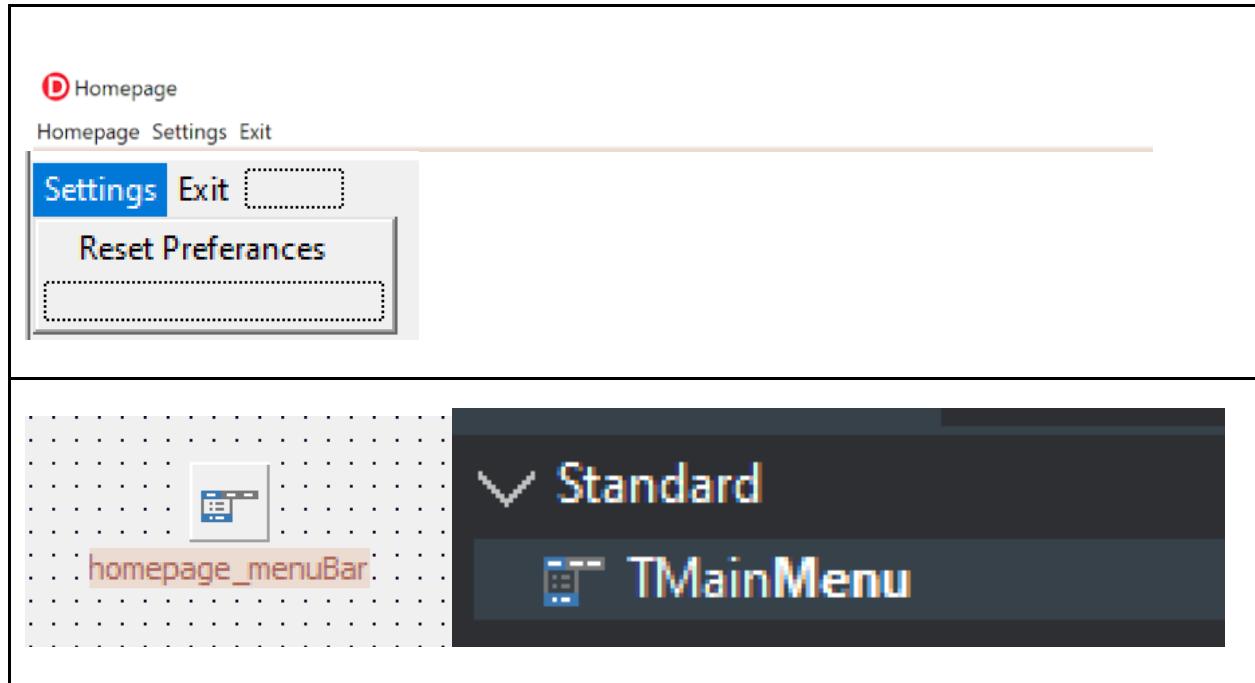
Project Summary

The loading screen is fully functional.

Stage 2 - Homepage Interface

Setting up the components

Menu Bar



To create the menu bar I used an inbuilt Delphi component called `TMainMenu`. I then populated the menu bar with:

- Settings -> Reset Preferences
- Exit

According to my design plan, the menu bar should have been in customs colours which matched the main colour scheme of my application. However menu bars are a standard component in many applications, hence I decided it may be better to leave this as the default. This is intended to make the application easier to navigate, especially for older users who may be used to this layout.

I then consulted my test users in order to ensure this was the right decision to make. Here was their feedback when asked to compare the design prototype with the current version :

- “I prefer the normal menu bar because that is what i have been used to for many, many years”
- “The one you’ve used in the current project stands out, meaning I can see the options more easily. As they are important, I prefer this”
- “It’s only a small part of the page, so I don’t really see a difference in either”

Hence I decided not to add any customisations to the menu bar.

Like and dislike buttons

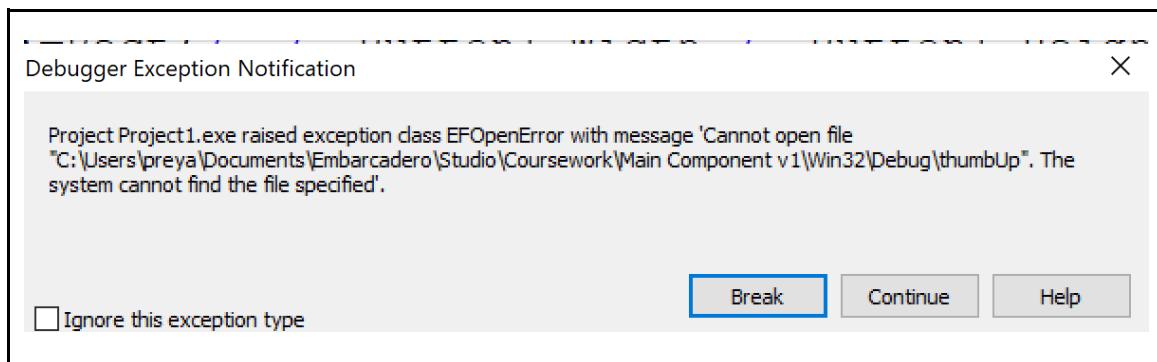
Original approach:

```
procedure Tform_homepage.Button1Click(Sender: TObject);
var
  Bitmap, Bitmap2: TBitmap;
  ButHandle: THandle;
  Rec: TRect;
begin
  Bitmap:=TBitmap.Create;
  Bitmap2:=TBitmap.Create;
  Bitmap2.LoadFromFile('thumbUp');
  Rec:=Rect(2, 2, Button1.Width-2, Button1.Height-2);
  ButHandle:=GetDC(Button1.Handle);
  Bitmap.Canvas.Handle:=ButHandle;
  Bitmap.Canvas.StretchDraw(Rec, Bitmap2);
end.
```

In these code two bitmaps are created. The file is loaded into the second, temporary bitmap. A rectangle is then created, which is slightly less than the button size. The function “GetDc” then gets the button context and stores it in a variable. The Bitmap then uses the context to adjust its properties. The bitmap image is then stretched to fill the rectangle previously created, hence appearing to “fill” the button up.

Errors :

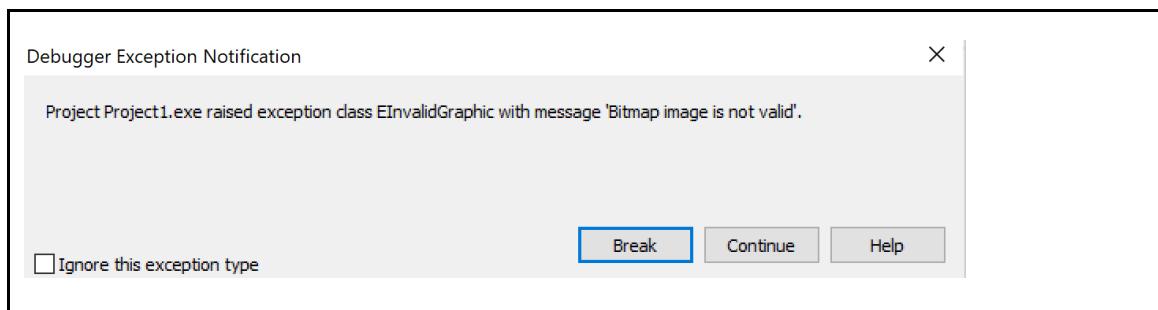
1.



When the code was run, this error appeared. Hence I realise that the file extension needs to be included in the file name, in order to fix this error.

```
Bitmap2.LoadFromFile('thumbUp.bmp');
```

2.



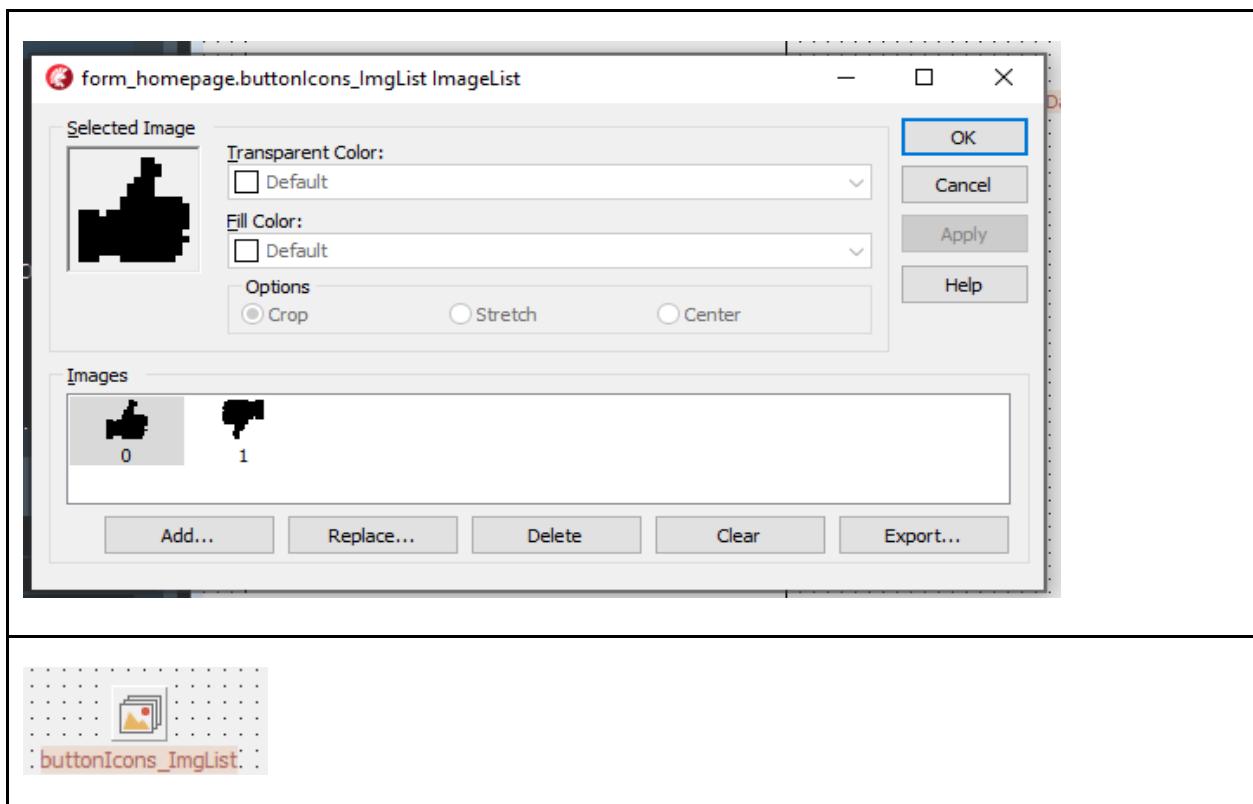
This error meant than my image was in the wrong format.

However at this point I realised that as the image on these buttons never needed to be changed whilst running the program , it made sense to set them through their properties, not by code. This would decrease the amount of code, the likelihood of errors and thus increase its long-term maintainability .

New approach :

This was the new approach I used to create the like and dislike button.

1. Added a button
2. Added an image list
3. Changed button images property to image list



4. Change image index to desired image

| | | | |
|------------|---------------------|------------|---------------------|
| ImageIndex | 0 | ImageIndex | 1 |
| ImageName | | ImageName | |
| > Images | buttonIcons_ImgList | > Images | buttonIcons_ImgList |

5. Changed from font colour to default text colour
6. Change button to bit button as this allows text colour to be changed

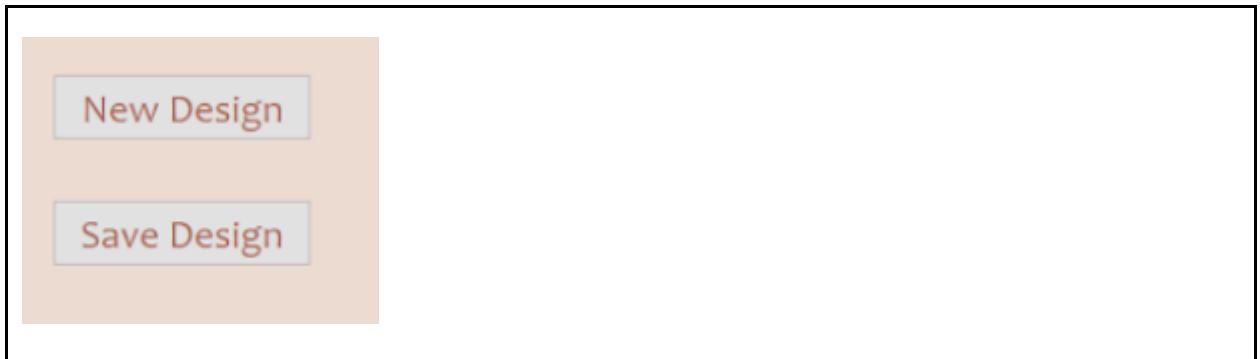


Final Buttons:

 Like Design

 Dislike Design

New Design and Save Buttons



I added two buttons, one for new design and one for save design. I then changed the font colour of these to match my theme and increased the text size in order to make it more easily readable for users.

Henna Canvas

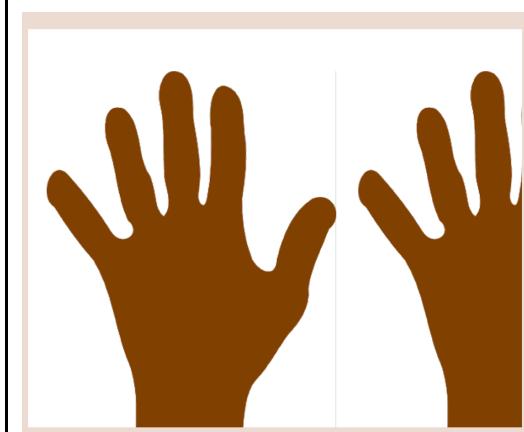
1. I saved the hand template I wanted to the project folder. I then photoshopped this and manipulated the colours to match my design.
2. I added a Delphi TCanvas component

```
procedure TForm_homepage.FormCreate(Sender: TObject);
Var
bitmap : TBitmap;
i,index: Integer;
begin
makeNewLabels();
//Loads hand template into canvas
bitmap := TBitmap.Create;
try
  Bitmap.LoadFromFile('handOutlineV5.bmp');
  Form_homepage.hennaDesign_canvas.Canvas.Brush.Bitmap := Bitmap;
  hennaDesign_canvas.Canvas.FillRect(Rect(0,0,1000,1000));
finally
  Form_LoadScreen.Canvas.Brush.Bitmap := nil;
  Bitmap.Free;
end;
```

3. Using this code I then loaded the hand image into it

Errors:

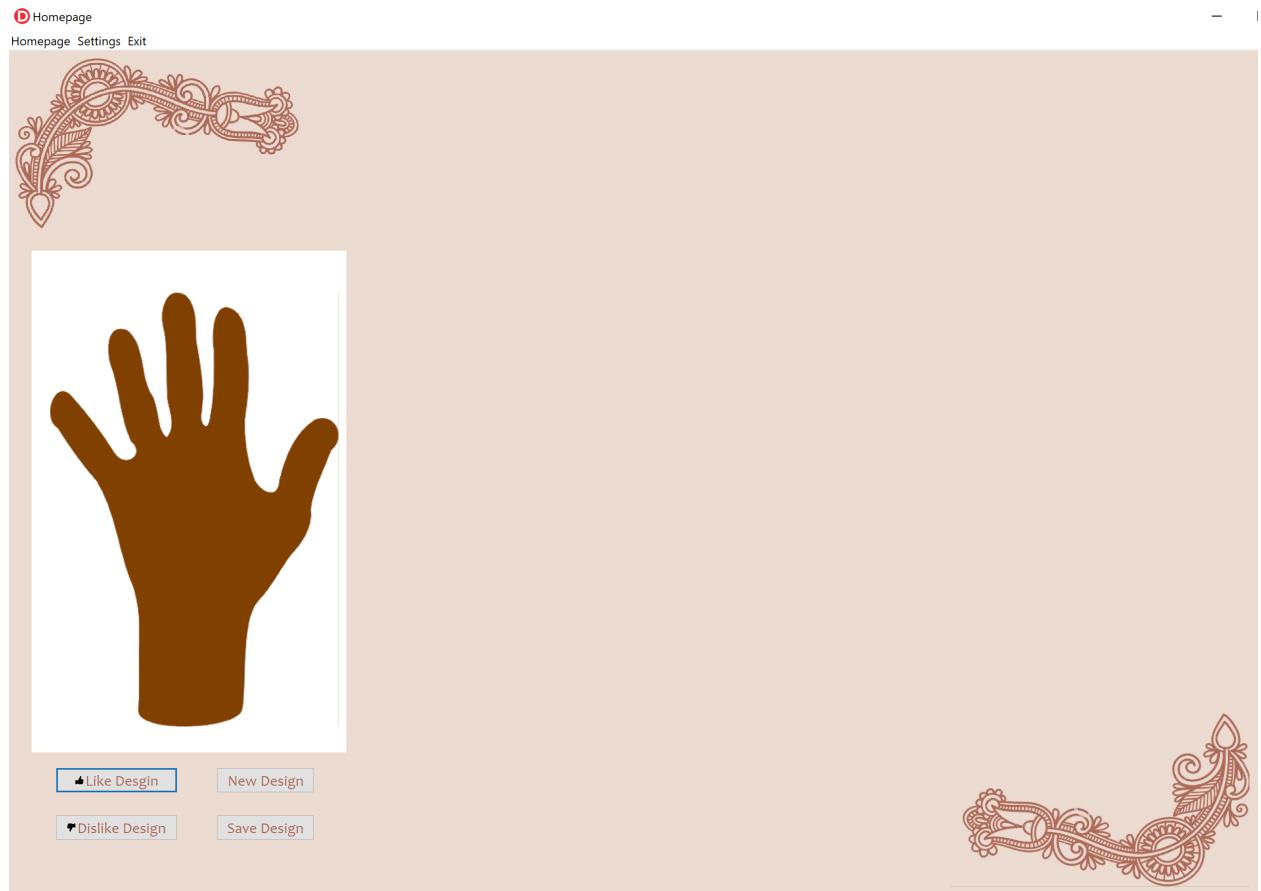
1. This gave me the issue of recurring hands appearing.



As I only wanted a singular hand to appear with no blank canvas or other hands, I resized the canvas component, so it only contained a singular hand.

Stage 2 Review

User Feedback



I showed my stakeholders this design. All three stakeholders seemed pleased with the design.

Changes

- Menu bar is not being customised and left with default properties as it is a standard component in many applications, hence this will make it easier for the users to navigate the application
 - Icons are being added to the buttons via an image list instead of through code
-

What has been done

The homepage interface is complete.

How has it been tested

Black box testing has occurred.

- The project is compared to what the output should be when the project is loaded
 - As this is the same as what is expected all tests were passed
-

Criteria Being met

- My application has a gender-neutral interface

- Lightweight Design
 - Colour pallet chosen in easy on the eye
-

Functionality Checklist

N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

Stage 3 - Elements

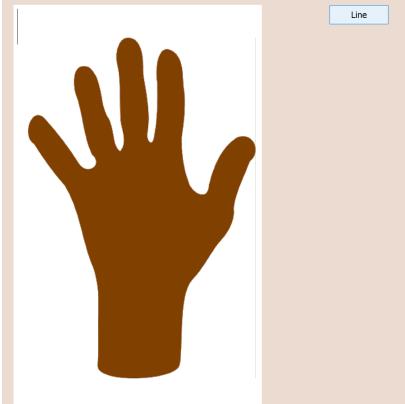
Plan

For this section my plan was as following:

1. Follow the pseudocode to create procedure for individual elements
2. Module test each element by adding a button to do this for hard coded parameters

Elements

Basic Line:

| | |
|--------|---|
| Code | <pre>procedure basicLine(x1,y1,x2,y2:integer); begin form2.hennaDesign_canvas.Canvas.MoveTo(x1,y1); form2.hennaDesign_canvas.canvas.lineto(x2,y2) ; end;</pre> <pre>procedure TForm2.Button1Click(Sender: TObject); begin basicLine(10,10,10,100) ; end;</pre> |
| Canvas |  |

Basic Circle

| | |
|--------|---|
| Code | <pre> procedure basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); begin form2.hennaDesign_canvas.canvas.Ellipse(topLeftX,topLeftY,bottomRightX,bottomRightY); end; procedure TForm2.CircleClick(Sender: TObject); begin basicCircle(10,10,200,200) end; </pre> |
| Canvas |  |

Filled Circle

| | |
|------|---|
| Code | <pre> procedure filledCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); begin with form2.hennaDesign_canvas.Canvas do begin brush.color := clBlack; //changes brush colour to black so whole shape drawn is in this colour basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY); brush.Color := clWhite; //changes brush colour back to deafult end; //form2.hennaDesign_canvas.Canvas.MoveTo(((topLeftX+topLeftY)DIV 2),((topLeftY+bottomRightY)DIV 2)); //form2.hennaDesign_canvas.Canvas.FloodFill(((topLeftX+topLeftY)DIV 2),((topLeftY+bottomRightY)DIV 2),clBlack,fsBorder); end; procedure TForm2.Button2Click(Sender: TObject); begin filledCircle(10,10,200,200); end; </pre> |
|------|---|



Originally, I attempted to flood fill the circle, as per my design (the commented-out code). However I realized that a better solution would be to change the colour of the circle when it is drawn from transparent to black. Therefore now if the circle overlaps with another shape, they will bleed together.

Dashed Circle

The main purpose of this one was to trial calling a nested procedure and changing canvas properties.

| | |
|------|---|
| Code | <pre> procedure dashedCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); begin with form2.hennaDesign_canvas.Canvas do begin pen.style := (psDot); basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY); end; end; { procedure TForm2.Button3Click(Sender: TObject); begin dashedCircle(10,10,200,200); end; } </pre> |
|------|---|



User Testing

- When the dashed circle button was clicked on by any other, it was drawn with a dashed line. This is as the pen style was never changed back to the default of psSolid.

```

procedure dashedCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer);
var
  test : string;
begin
  with form2.hennaDesign_canvas.Canvas do
begin
  pen.style := (psDot); // changes pen style - affects how border is draw
  basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
  test := 'psSolid';
  pen.style := (test); // changed back to deafult
end;
end;
|
```



```

procedure dashedCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer);
var
  test : TPenStyle;
begin
  with form_homepage.hennaDesign_canvas.Canvas do
begin
  pen.style := (psDot); // changes pen style - affects how border is draw
  basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
  test := psSolid;
  pen.style := (test); // changed back to deafult
end;
```

Hence here I used a variable to assign the pen style back to its default. Initially I tried to declare the variable type as string, this is incorrect as it needs to be assigned as a TPenStyle type. This fixed the issue of all elements being drawn in a dashed style.

Random pen style

Initially I attempted to use the random function in Delphi. This function produced a result between zero and one. To ensure I had the write idea I outputted the numbers to a message box.

The screenshot shows a Delphi IDE window. On the left, there is a code editor with the following Delphi code:

```
function choosePenStyle(): TPenStyle;
var
  p : real;
  test : string;
begin
  p := random ;
  test := floattostr(p);
  ShowMessage(test);
end;
```

On the right, there are three vertically stacked message boxes, each titled "Project1". The first message box contains the value "0.331851877970621". The second message box contains the value "0.649067234247923". The third message box contains the value "0.187425252282992". Each message box has an "OK" button at the bottom right.

This produced the expected results. However I quickly realised that the number of pen styles would be an integer, whereas the random number generated would be a floating number. Hence these different variable types would cause issues of conversion. Consequently I decided it would make more sense to use the Random(n) function in Delphi. This function produced a random integer between 0 and n-1 (inclusive).

```

function choosePenStyle(clearAllowed : bool): TPenStyle;
var
  numStyles,p, pInterval : integer;
  test : string;
  selectedStyle : TPenStyle;
begin
  if clearAllowed = true then
    begin
      p := 1 + Random(6);
    end
  else
    begin
      p := 1 + Random(5);
    end;

  numStyles := 6;
  pInterval := p DIV numStyles;
  case p of
    1 : selectedStyle := psSolid;
    2 : selectedStyle := psDash;
    3 : selectedStyle := psDot;
    4 : selectedStyle := psDashDot;
    5 : selectedStyle := psDashDotDot;
    6 : selectedStyle := psClear;
  end ;
  choosePenStyle := selectedStyle;
end;

```

```

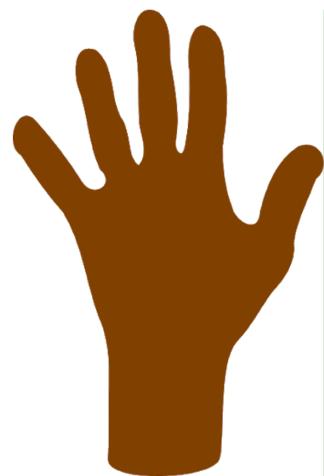
procedure TForm_homepage.Button4Click(Sender: TObject);
begin
  form_homepage.hennaDesign_canvas.canvas.pen.Style := choosePenStyle(true);
  basicLine(10,10,10,1000) ;
  form_homepage.hennaDesign_canvas.canvas.pen.Style := psSolid;
end;

```

I tested the pen styles by clicking the button above, which draws a line in a randomized pen style. I clicked this button until at least one line was drawn in each style.

| | |
|-------|----------|
| Style | Evidence |
|-------|----------|

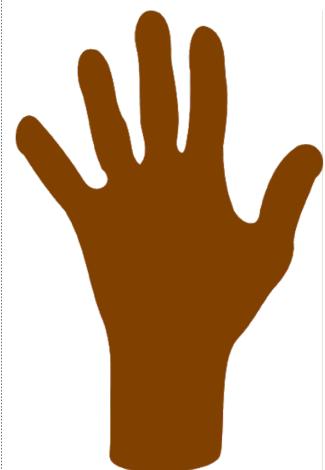
psSolid



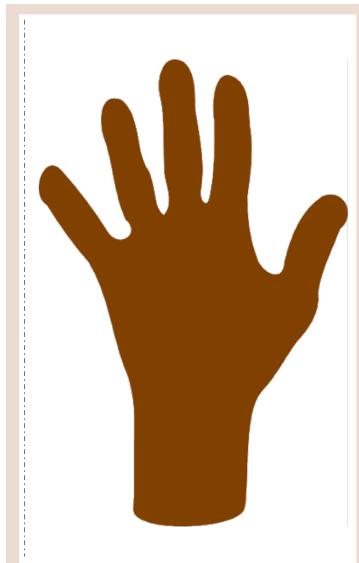
psDash



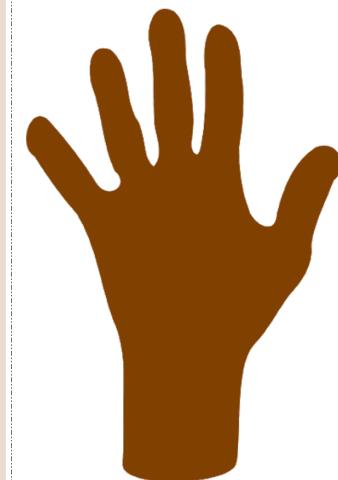
psDot



psDashDot



psDashDotDot



psClear



Inner reflection and mini circle border

Both of these elements share a common property - points on their border which are evenly spaced and opposite to each other need to be calculated.

```

function circleSixPoints(topLeftX,topLeftY,bottomRightX,bottomRightY:integer) : TArray<TCooridantes>;
var
  coordinates : array[0..5] of TCooridantes; //0 starts from the one directly above the centre, then counted clockwise
  centre : TCooridantes;
  radius : integer;
begin
  centre.xCord := (topLeftX + bottomRightX) / 2;
  centre.yCord := (topLeftY + bottomRightY) / 2;
  radius := (topLeftX - centre.xCord) - (topLeftY - centre.yCord)^2);
end;

```

First I calculated the midpoint and radius of the circle, as can be seen in the procedure above.

```

procedure circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); var coordinates :array of TCooridantes )
var //0 starts from the one directly above the centre, then counted clockwise
  centre : TCooridantes;
  radius : integer;
begin
  centre.xCord := (topLeftX + bottomRightX) / 2;
  centre.yCord := (topLeftY + bottomRightY) / 2;
  coordinates [0].xCord := centre.xCord;
  coordinates [0].yCord := topLeftY;
  coordinates [1].xCord := (centre.xCord + bottomRightX)/2;
  coordinates [1].yCord := (topLeftY + centre.yCord)/2;
  coordinates [2].xCord := bottomRightX ;
  coordinates [2].yCord := centre.yCord;
  coordinates [3].xCord := (centre.xCord + bottomRightX)/2;
  coordinates [3].yCord := (bottomRightY + centre.yCord)/2;
  coordinates [4].xCord := centre.xCord ;
  coordinates [4].yCord := bottomRightY;
  coordinates [5].xCord := (centre.xCord + topLeftX)/2 ;
  coordinates [5].yCord := (bottomRightY + centre.yCord)/2;
  coordinates [6].xCord := topLeftX ;
  coordinates [6].yCord := centre.yCord;
  coordinates [7].xCord := (centre.xCord + topLeftX)/2 ;
  coordinates [7].yCord := (topLeftY + centre.yCord)/2;
end;

```

I realised the procedure was incorrectly named, as for it to be aesthetic 8 points on the circle border need to be calculated. In order to make it clear for future reference, I corrected this.

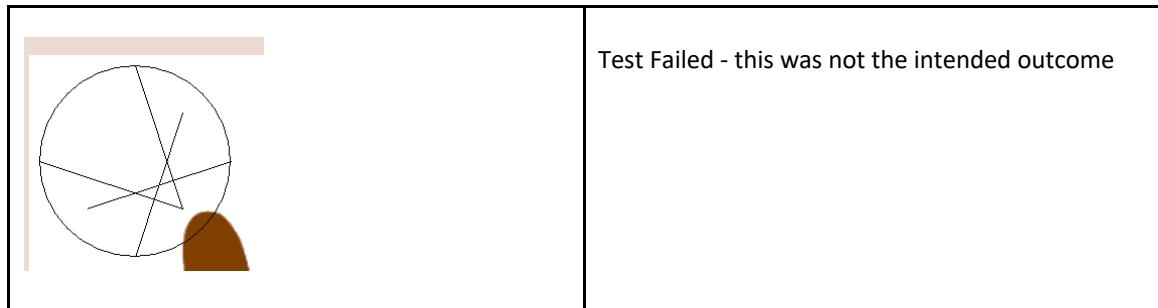
Furthermore initially I had intended the subroutine to be a function which would return an array of the coordinates. However this is tricky in Delphi, it was easier to pass an array by reference and change its values. Thus I changed the subroutine to a procedure.

Inner Reflection Circle

```

procedure innerReflectionCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer) ://passed by reference - so changes values
//0 starts from the one directly above the centre, then counted clockwise
Var
  miniCircleCenters : array[0..7] of TCooridantes;
  i, arrayHalfWay : integer;
begin
  basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
  circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY,miniCircleCenters) ;
  arrayHalfWay := (length(miniCircleCenters))DIV 2)-1 ;
  for i := 0 to arrayHalfWay do
  begin
    basicLine(trunc(miniCircleCenters[i].xCord),trunc(miniCircleCenters[i].yCord),trunc(miniCircleCenters[i+arrayHalfWay].xCord),trunc(miniCircleCenters[i + arrayHalfWay].yCord))
  end;
end;

```



Hence from the image it is apparent the points being returned and consequently joined together are incorrect.

The first thing I did to find the problem was ensure the right formula was being used. As calculating points on a circle's circumference is an old mathematical problem, I conducted a google search to verify the method:

<https://stackoverflow.com/questions/839899/how-do-i-calculate-a-point-on-a-circle-s-circumference>

From this link it became clear that I was using the wrong formula. All points on the circumference must have the same magnitude but be in different directions. Thus trigonometry would be needed to resolve their lengths vertically and horizontally.

| | |
|---|---|
| <pre>coordinates[1].xCord := centre.xCord + radius * cos(PI/4); coordinates[1].yCord := centre.yCord + radius * sin(PI/4); coordinates[2].xCord := bottomRightX -</pre> | <p>Test failed. The diagonal line from top left to bottom right should now be on the circumference. Whilst the direction is right the length of the line is wrong, hence there must be an issue in calculating the radius. When checked the radius was being calculated by the following :</p> <pre>radius := ((topLeftX - centre.xCord) - (topLeftY - centre.yCord)^2))</pre> <p>The radius could actually be calculated by just doing the $\sqrt{(topLeftX - centreX)^2 + (topLeftY - centreY)^2}$. When the code was modified to this, it solved the error.</p> |
| | |

I then applied this to all the points in the array. It is important to note that the angle being measured from most right point on circle as 0 rad. Thus the :

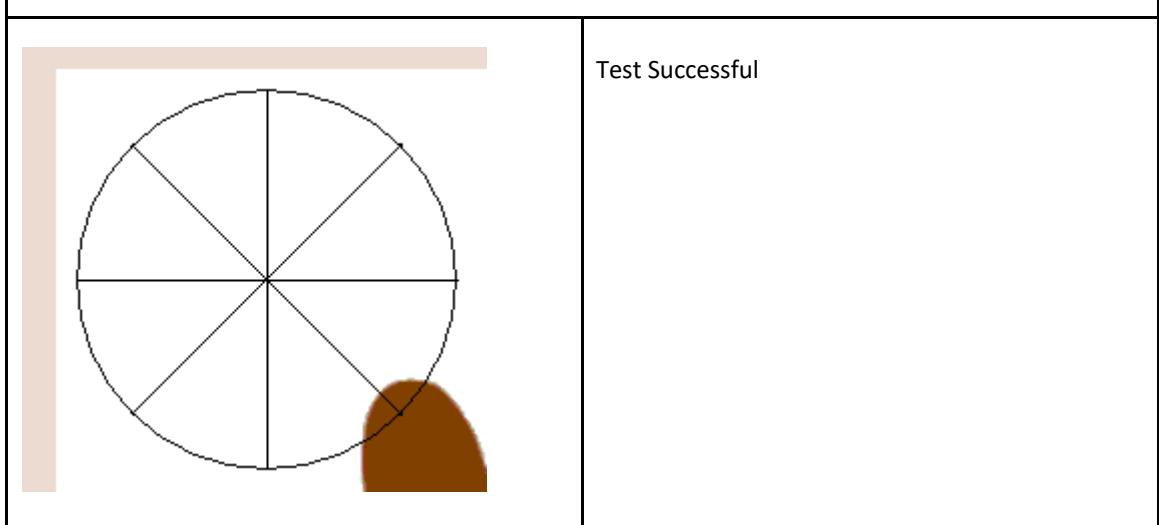
- Top right diagonal is $-1/4\pi$ from full circle (2π), so use the angle $7/8\pi$

- Then increment by $\pi/2$ ($\frac{1}{4}$ circle) and then MOD $\pi/2$ - to give right value in the range of 0 to 2 π

```

procedure circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY:integer);
var //0 starts from the one directly above the centre, then counted clockwise
centre : TCooridantes;
radius : real;
begin
centre.xCord := (topLeftX + bottomRightX) / 2;
centre.yCord := (topLeftY + bottomRightY) / 2;
radius := bottomRightY - centre.yCord ;
coordinates[0].xCord := centre.xCord;
coordinates[0].yCord := topLeftY;
coordinates[1].xCord := centre.xCord + radius * cos((7*PI)/4);
coordinates[1].yCord := centre.yCord + radius * sin((7*PI)/4);
coordinates[2].xCord := bottomRightX;
coordinates[2].yCord := centre.yCord;
coordinates[3].xCord := centre.xCord + radius * cos((1*PI)/4);
coordinates[3].yCord := centre.yCord + radius * sin((1*PI)/4);
coordinates[4].xCord := centre.xCord ;
coordinates[4].yCord := bottomRightY;
coordinates[5].xCord := centre.xCord + radius * cos((3*PI)/4);
coordinates[5].yCord := centre.yCord + radius * sin((3*PI)/4);
coordinates[6].xCord := topLeftX ;
coordinates[6].yCord := centre.yCord;
coordinates[7].xCord := centre.xCord + radius * cos((5*PI)/4);
coordinates[7].yCord := centre.yCord + radius * sin((5*PI)/4);
end;

```

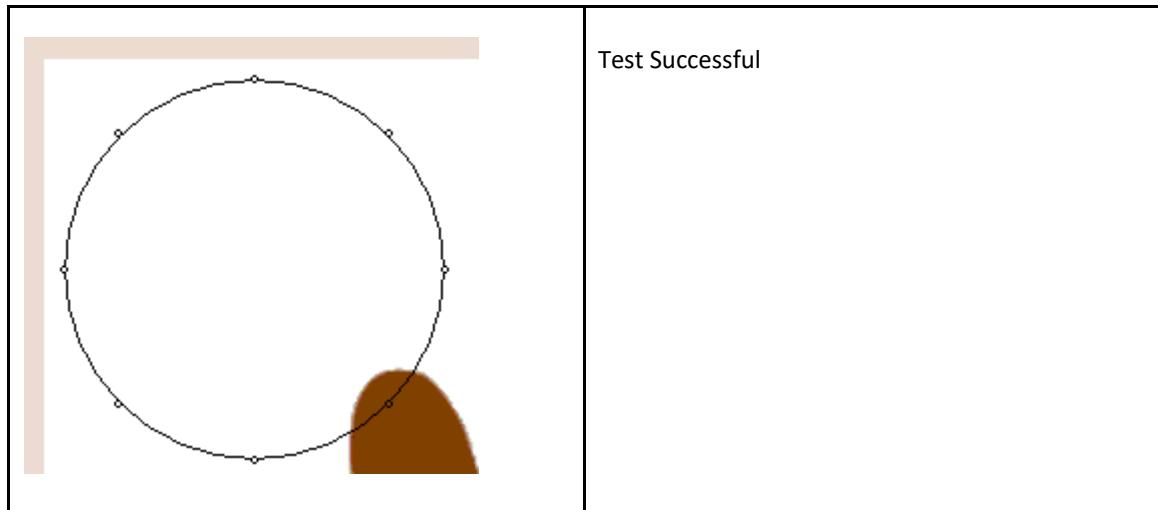


Circle Border

```

procedure miniCircleBorder(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); //passed by reference - so changes values
//0 starts from the one directly above the centre, then counted clockwise
Var
miniCircleCenters : array[0..7] of TCooridantes;
i, arrayHalfWay : integer;
begin
basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY,miniCircleCenters) ;
arrayHalfWay := (length(miniCircleCenters)DIV 2)-1 ;
for i := 0 to 7 do
begin
basicCircle(trunc(miniCircleCenters[i].xCord-2),trunc(miniCircleCenters[i].yCord-2),trunc(miniCircleCenters[i].xCord+2),trunc(miniCircleCenters[i].yCord+2))
end;

```



Using the now correctly calculated coordinates, I iterated through the array of eight points on the circle circumference and drew a circle whose top left was (pointX -2, pointY-2) and bottom right (pointX + 2, pointY + 2).

User feedback:

- No critical comments on the inner reflection circle
- For the mini circle border
 - The circles should be bigger

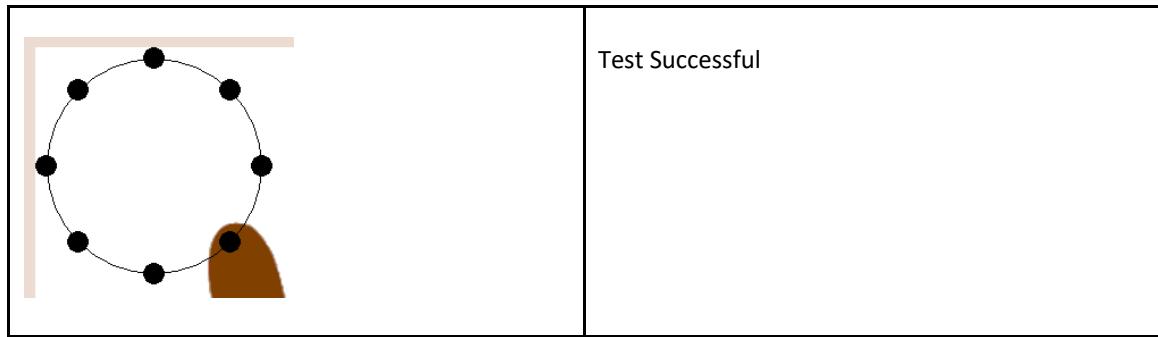
Hence I adapted my code so that the radius of the mini circles is calculated as a proportion of the big circle, and this is then used to determine the top left and bottom right coordinate.

I also used filled circles, as these would be more visible and add nice contrast to the henna design.

```

procedure miniCircleBorder(topLeftX,topLeftY,bottomRightX,bottomRightY:integer); //0 starts from the one directly above the centre, then counted clockwise
Var
  miniCircleCenters : array[0..7] of TCoordantes;
  i, arrayHalfWay : integer;
  centre : TCoordantes;
  radius,miniRadius : real;
begin
  centre.yCord := (topLeftY + bottomRightY) / 2;
  radius := bottomRightY - centre.yCord ;
  miniRadius := radius/10;
  miniCircle((topLeftX,topLeftY,bottomRightX,bottomRightY);
  circleEightPoints((topLeftX,topLeftY,bottomRightX,bottomRightY,miniCircleCenters));
  arrayHalfWay := (length(miniCircleCenters)DIV 2)-1;
  for i := 0 to 7 do
  begin
    filledCircle(trunc(miniCircleCenters[i].xCord-miniRadius),trunc(miniCircleCenters[i].yCord-miniRadius),trunc(miniCircleCenters[i].xCord+miniRadius),trunc(miniCircleCenters[i].yCord+miniRadius));
  end;
end;

```



Triangle

Code:

```
Procedure basicTriangle(x1,y1,x2,y2,x3,y3:integer);
begin
  basicLine(x1,y1,x2,y2);
  basicLine(x1,y1,x3,y3);
  basicLine(x2,y2,x3,y3);
  addToLinkedList('basicTriangle',elementsLl,elementsTop,elementsLlSp);
end;
```

Canvas:



Stage 3 Review

User Feedback SUMMARY

- For the mini circle border the circles should be bigger
-

Changes

- Random numbers are generated as a
 - Formula for points of circumference of circle changed
 - The size of the mini circles is calculated as a proportion of the big circle
-

What has been done

Procedures to draw all elements have been created.

How has it been tested

Black box testing has occurred.

- The project is compared to what the output should be when the project is loaded
- As this is the same as what is expected all tests were passed

User testing has occurred.

- The users were allowed to click buttons as they desired – this discovered the issue of changing the pen stroke
-

Criteria Being met

- Partially : The program generates henna designs
-

Functionality Checklist

N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Stage 4 – Basic Templates

Plan

1. Combine the elements to for templates
2. Test the templates

Template 1

1. Figuring out the centre
 - a. As this template has the primary focus of helping users practise symmetrical patterns, it is essential to work out the hand centre.
 - b. As it is likely that the hand centre would need to be accessed in other templates, I declared it as a global variable.

```
handCentre : TCooridantes;
```

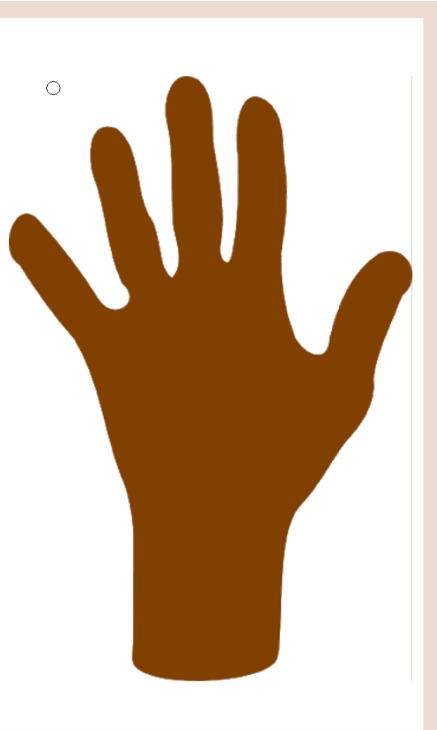
- c. I set up template 1 to draw basic circle with a radius of 10 and middle point where the hand centre was declared

```
procedure template1(middleCord : TCooridantes); // so actual data never changed - safety measure
begin
  basicCircle(trunc(middleCord.xCord-10), trunc(middleCord.yCord-10), trunc(middleCord.xCord+10), trunc(middleCord.yCord+10));
```

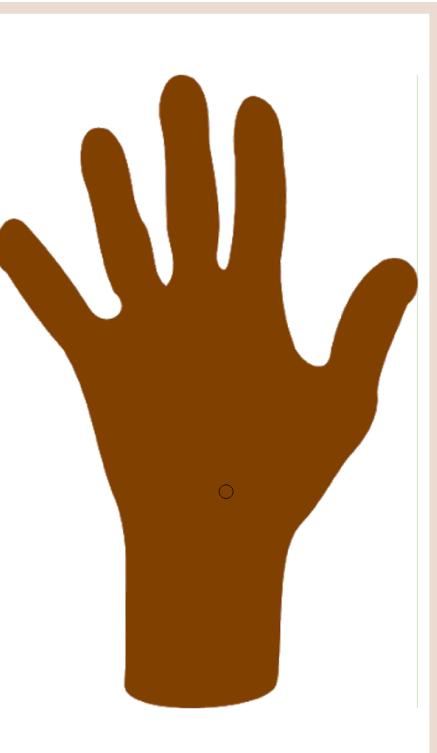
- d. Using trial and error, paired with visual adjustment and users' opinion I altered the values of the handCentre.xCord and handCentre.yCord. Doing it heuristically like this would be easier over precise calculations, as the hand shape is not uniform and requires human judgment, as is the case with many art forms. Each iteration allowed me to get closer to the centre of the hand.

| | |
|--------|--------|
| Values | Canvas |
|--------|--------|

```
handCentre.xCord := 100;  
handCentre.yCord := 100;
```



```
handCentre.xCord := 350;  
handCentre.yCord := 650;
```





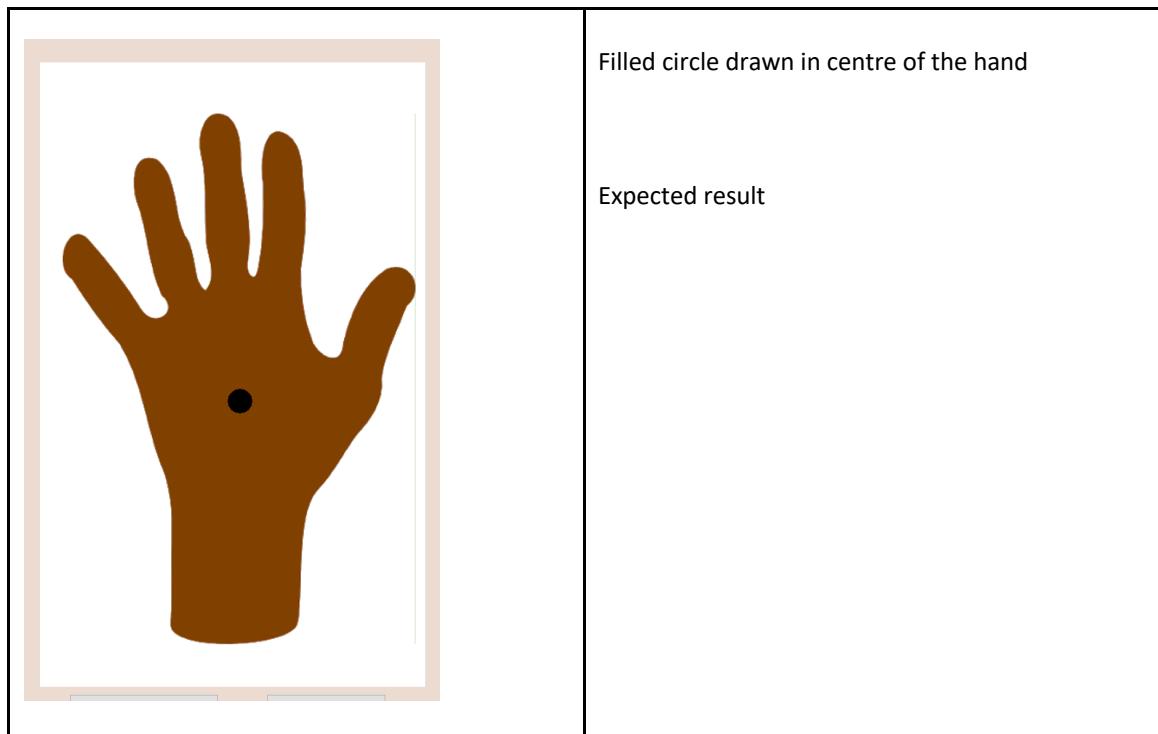
All three of my test users said the last one looked in the centre of the hand, therefore I believe these coordinates are valid to use.

2. Calling the element procedures

```

procedure templatel(middleCord : TCooridantes); // so acutal data never changed - saftey measure
begin
  filledCircle(trunc(middleCord.xCord-20),trunc(middleCord.yCord-20),trunc(middleCord.xCord+20),trunc(middleCord.yCord+20));
end;

```



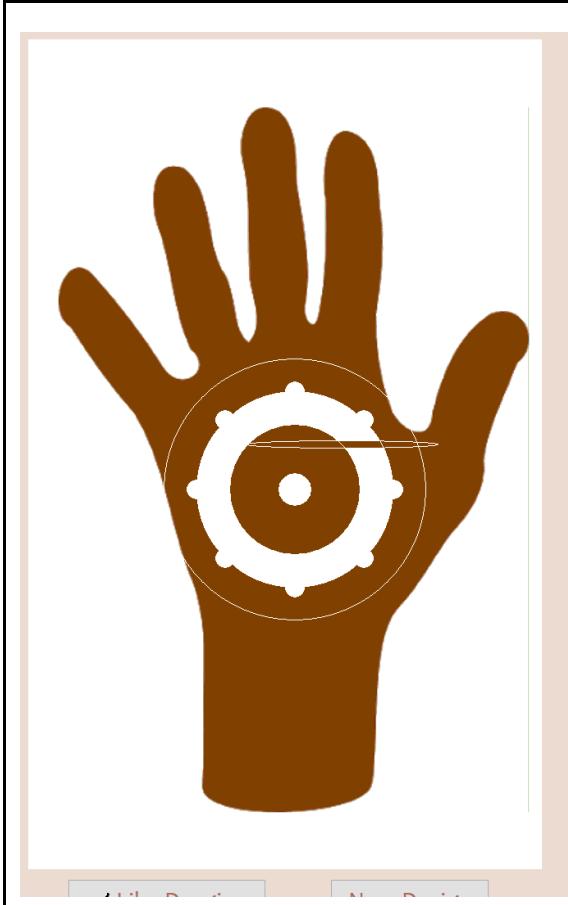
Filled circle drawn in centre of the hand

Expected result

| | | |
|--|------|-------|
| On | form | load: |
| <pre>defaultBrushStyle := bsClear; defaultColour := clWhite; form2.hennaDesign_canvas.Canvas.Brush.Color := defaultColour; form2.hennaDesign_canvas.Canvas.pen.Color := defaultColour;</pre> | | |

This changed the brush and pen colour to white when the form loads, allowing all templates to be drawn in white. Hence if the shapes bleed out of the hand outline, they will match with the background, allowing a clean design to be drawn.

| | |
|--|--|
| <pre>procedure template1(middleCord : TCoordinantes); // so actual data never changed - safety measure var initialRadius,currentRadius : integer; begin initialRadius := 20; // can randomise after filledCircle(trunc(middleCord.xCord-initialRadius),trunc(middleCord.yCord-initialRadius),trunc(middleCord.xCord+initialRadius),trunc(middleCord.yCord+initialRadius)); currentRadius := initialRadius*4; basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius)); currentRadius := initialRadius*8; miniCircleBorder(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius)); currentRadius := initialRadius*16; basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius)); with form2.hennaDesign_canvas.Canvas do begin basicCircle(trunc(middleCord.xCord-currentRadius)+100,trunc(middleCord.yCord-currentRadius)+100, 500,500); Brush.Style := bsSolid; Brush.Color := clWhite; FillDraw(trunc(middleCord.xCord-currentRadius)+100,trunc(middleCord.yCord-currentRadius)+100, clWhite, fsBorder); end;</pre> | |
|--|--|

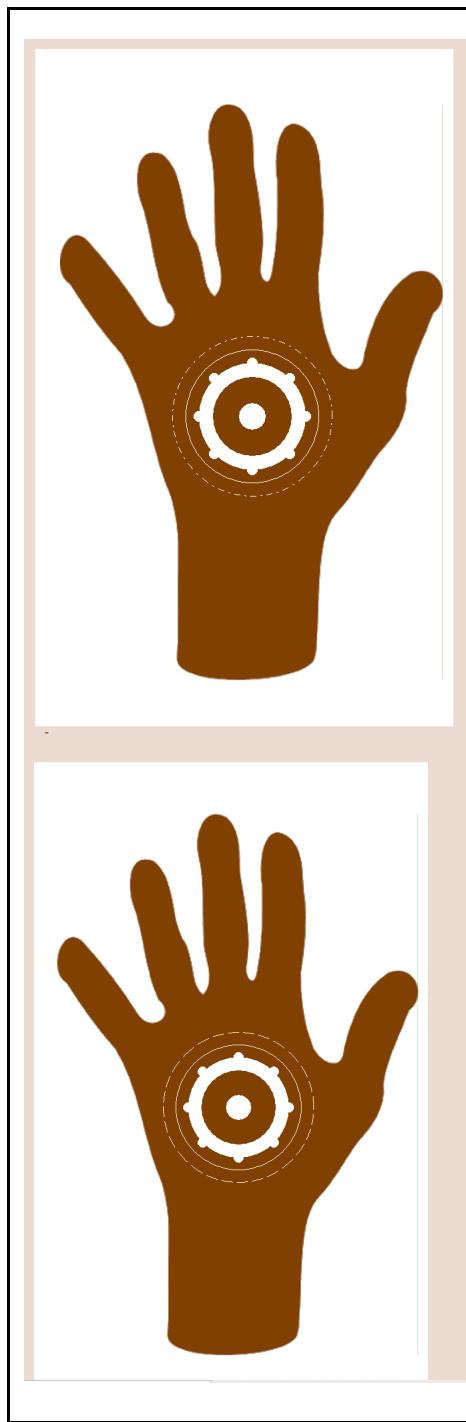


Initially a large proportion of the hand was turning white when the template was run. I determined this must be due to the floodfill. Hence I drew a circle to determine where the floodfill was being started from. Thus I visually moved the ellipse till it was in the correct location and the right area was being flood filled, as can be seen in the image on the left.

```

begin
with form2.hennaDesign_canvas.Canvas do
begin
initialRadius := 20; //can randomise after
filledCircle(trunc(middleCord.xCord-initialRadius),trunc(middleCord.yCord-initialRadius),trunc(middleCord.xCord+initialRadius),trunc(middleCord.yCord+initialRadius));
currentRadius := initialRadius*3;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
currentRadius := initialRadius*4;
miniCircleBorder(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
currentRadius := initialRadius*5;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
Brush.Style := bsSolid;
Brush.Color := clWhite;
BrushStyle := defaultBrushStyle;
pen.Style := choosePenStyle(false);
currentRadius := initialRadius*6;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
pen.Style := defaultPenStyle;
end;
end;

```



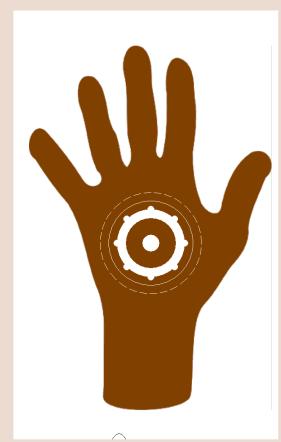
To improve the aesthetics of the design, I made the circles smaller and added a randomly chosen decorative border around the edge.

I also changed the code, so the dimensions of the circle are determined from an initial set of values. This makes my code more resistant to any changes in the application, e.g. a change in the hand size.

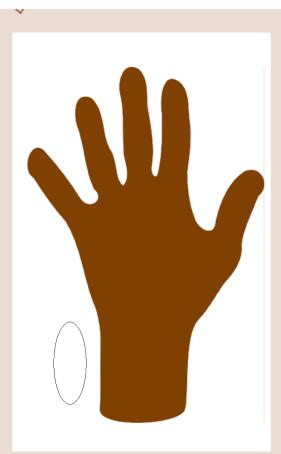
3. Determining the wrist coordinates

To get a vague idea of the corner points of the section of the wrist on which henna is usually drawn, I drew an ellipse. Applying the same method as the hand centre coordinates, I modified the coordinates of the ellipse.

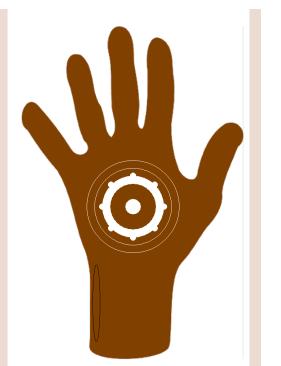
```
||basicCircle(200,1000,300,1500);
```



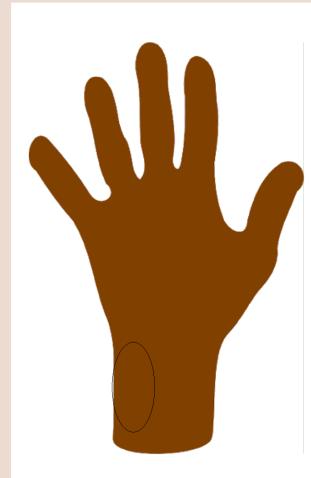
```
||basicCircle(180,700,100,900);
```



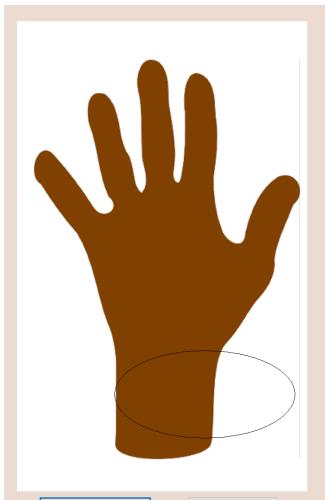
```
||basicCircle(220,700,240,900);
```



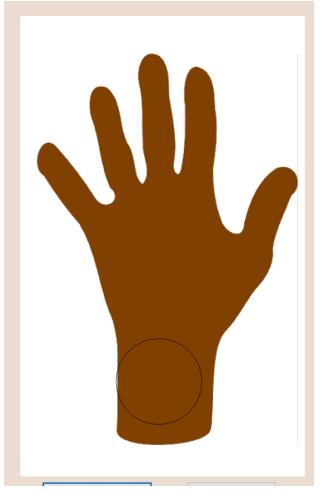
```
|| basicCircle(220,700,240,900);
```

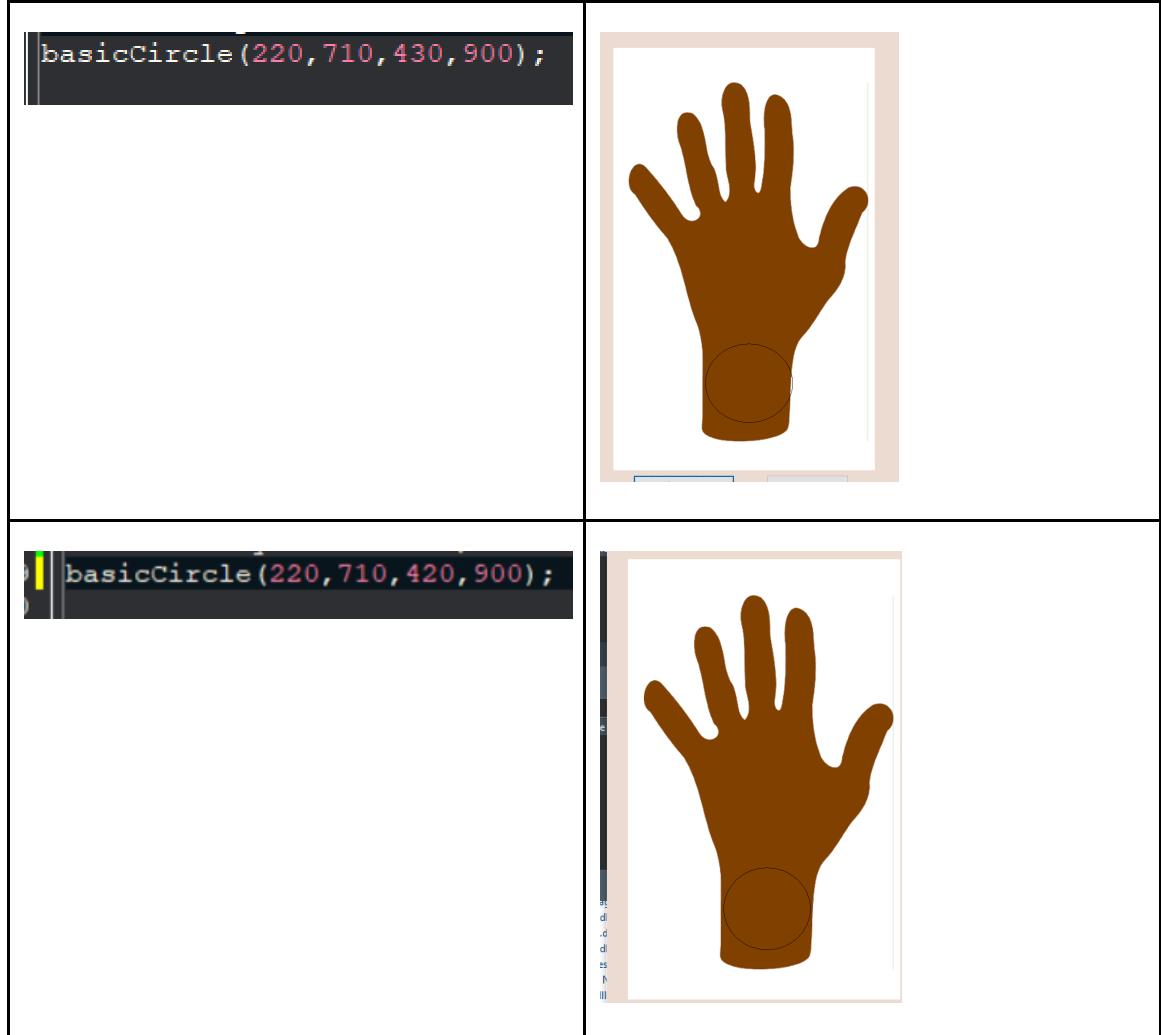


```
basicCircle(210,710,600,900);
```



```
|| basicCircle(210,710,400,900);
```

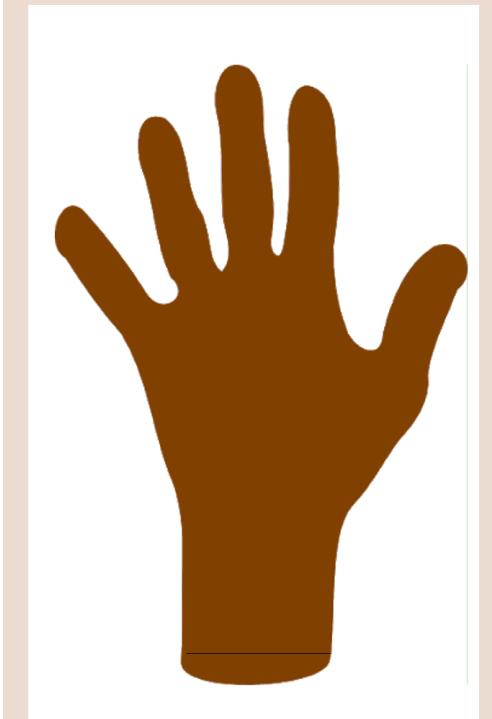




The coordinate produced by the last ellipse seemed to cover the wrist area sufficiently, hence I stored the coordinates in a variable of type coordinate block when the form was loaded.

| |
|--|
| <pre>wristMaxPoints.topLeft.xCord :=220; wristMaxPoints.topLeft.yCord :=900; wristMaxPoints.bottomLeft.xCord :=220; wristMaxPoints.bottomLeft.yCord :=710; wristMaxPoints.topRight.xCord :=420; wristMaxPoints.topRight.yCord :=900; wristMaxPoints.bottomRight.xCord :=420; wristMaxPoints.bottomRight.yCord :=710;</pre> |
|--|

To test these were added properly , I attempted to draw lines between the coordinates.

| | |
|---|---|
| <pre>baseLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord), trunc(wristMaxPoints.topRight.xCord), trunc(wristMaxPoints.topRight.yCord))</pre> | |
|  | <p>Failed</p> <p>This line was supposed to be drawn as a line on the top part of the wrist section.</p> <p>When I looked at the stored coordinates again it became apparent that top and bottom coordinates had been swapped.</p> <p>Thus I realised the variables.</p> <pre>wristMaxPoints.topLeft.xCord :=-210; wristMaxPoints.topLeft.yCord :=-750; wristMaxPoints.bottomLeft.xCord :=-210; wristMaxPoints.bottomLeft.yCord :=-900; wristMaxPoints.topRight.xCord :=420; wristMaxPoints.topRight.yCord :=-750; wristMaxPoints.bottomRight.xCord :=420; wristMaxPoints.bottomRight.yCord :=-900;</pre> <p>This fixed the issue.</p> |

Using stakeholder feedback I refined the initial values to the following:

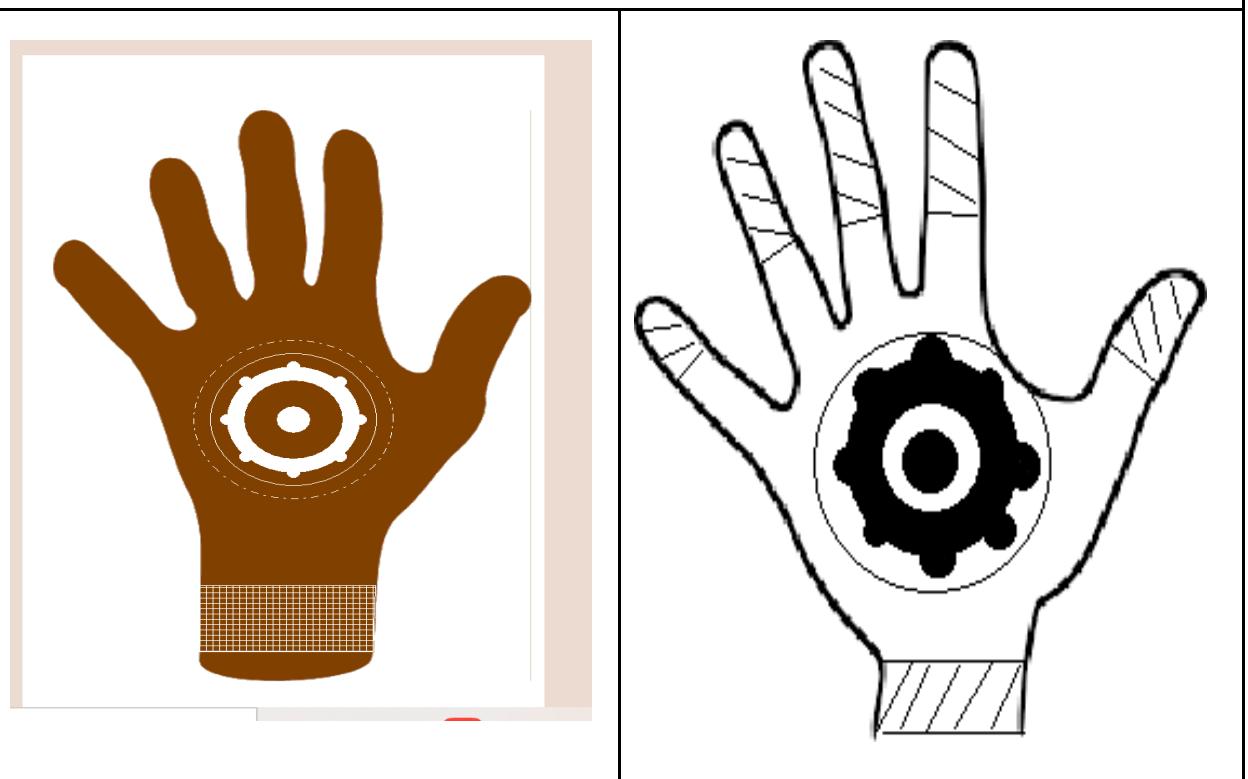
```
handCentre.xCord := 325;
handCentre.yCord := 550;
wristMaxPoints.topLeft.xCord :=212;
wristMaxPoints.topLeft.yCord :=800;
wristMaxPoints.bottomLeft.xCord :=212;
wristMaxPoints.bottomLeft.yCord :=900;
wristMaxPoints.topRight.xCord :=423;
wristMaxPoints.topRight.yCord :=800;
wristMaxPoints.bottomRight.xCord :=420;
wristMaxPoints.bottomRight.yCord :=900;
```

4. I then incorporated the wrist section in my template.

```

procedure template1(middleCord : TCoordantes); // so actual data never changes
var
initialRadius,currentRadius : integer;
begin
with form2.hennaDesign_canvas.Canvas do
begin
initialRadius := 20; //can randomise after
currentRadius := initialRadius*2;
filledCircle(trunc(middleCord.xCord-initialRadius),trunc(middleCord.yCord-initialRadius),trunc(middleCord.xCord+initialRadius),trunc(middleCord.yCord+initialRadius)); //saftey measure
currentRadius := initialRadius*3;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
currentRadius := initialRadius*4;
miniCircleBorder(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
currentRadius := initialRadius*5;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
Brush.Style := bsSolid;
Brush.Color := clWhite;
FloodFill(trunc((middleCord.xCord-currentRadius)+(initialRadius*2.5)),trunc((middleCord.yCord-currentRadius)+((initialRadius)*2.5)), clWhite, fsBorder);
Brush.Style := defaultBrushStyle;
pen.Style := choosePenStyle(false);
currentRadius := initialRadius*6;
basicCircle(trunc(middleCord.xCord-currentRadius),trunc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+currentRadius),trunc(middleCord.yCord+currentRadius));
pen.Style := defaultPenStyle;
basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord),trunc(wristMaxPoints.topRight.xCord),trunc(wristMaxPoints.topRight.yCord));
basicLine(trunc(wristMaxPoints.bottomLeft.xCord), trunc(wristMaxPoints.bottomLeft.yCord),trunc(wristMaxPoints.bottomRight.xCord),trunc(wristMaxPoints.bottomRight.yCord));
basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord),trunc(wristMaxPoints.bottomLeft.xCord),trunc(wristMaxPoints.bottomLeft.yCord));
basicLine(trunc(wristMaxPoints.topRight.xCord), trunc(wristMaxPoints.topRight.yCord),trunc(wristMaxPoints.bottomRight.xCord),trunc(wristMaxPoints.bottomRight.yCord));
Brush.Style := bsCross; //can randomise after
Brush.Color := clWhite;
FloodFill(trunc((wristMaxPoints.topLeft.xCord)+10),trunc((wristMaxPoints.topLeft.yCord)+10), clWhite, fsBorder);
Brush.Style := defaultBrushStyle;
end;
end;

```



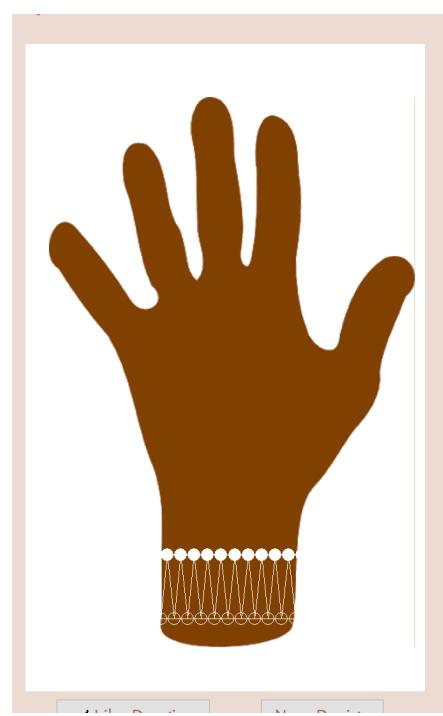
Success

The code above draws a box around the wrist area. It then changed the brush style to cross. A point inside the wrist box is then chosen and floodfilled.

Template 2

I decided to keep template two relatively simple for the time being. This would allow me to code the rest of the program and test it with a relatively small amount of data. Hence this would ensure the functionality of my code and ensure it is written in a scalable way.

```
procedure template2(middleCord : TChordantes);
var
  heightDiff, widthDiff, widthInterval : real;
  i : integer;
begin
  with form2.hennaDesign.canvas.Canvas do
  begin
    basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord), trunc(wristMaxPoints.topRight.xCord), trunc(wristMaxPoints.topRight.yCord));
    basicLine(trunc(wristMaxPoints.bottomLeft.xCord), trunc(wristMaxPoints.bottomLeft.yCord), trunc(wristMaxPoints.bottomRight.xCord), trunc(wristMaxPoints.bottomRight.yCord));
    widthDiff := wristMaxPoints.topRight.xCord - wristMaxPoints.topLeft.xCord;
    widthInterval := widthDiff / 10;
    for i := 0 to 9 do
    begin
      basicTriangle((trunc(wristMaxPoints.topLeft.xCord + (widthInterval * i))), trunc(wristMaxPoints.bottomLeft.yCord), trunc((widthInterval/2)+(wristMaxPoints.topLeft.xCord + (widthInterval * i))), trunc(wristMaxPoints.topLeft.yCord), trunc(wristMaxPoints.topLeft.yCord));
      for i := 0 to 10 do
      begin
        basicCircle((trunc( (wristMaxPoints.topLeft.xCord + (widthInterval * i)-10) ), trunc(wristMaxPoints.bottomLeft.yCord-10), (trunc( (wristMaxPoints.topLeft.xCord + (widthInterval * i)+10)), trunc(wristMaxPoints.bottomLeft.yCord+10), (widthInterval * i))), trunc(wristMaxPoints.topLeft.yCord - 10), trunc((widthInterval/2)+(wristMaxPoints.topLeft.xCord + 10)+(widthInterval * i)), trunc(wristMaxPoints.topLeft.yCord));
      end;
    end;
  end;
end;
```



Success

Stage 4 Review

User Feedback

- All users liked both designs
 - Neha said the second one was slightly simplistic; however this is intentional as it is to help people learn the geometrical principles in an easy way
-

Changes

N/A

What has been done

Two templates have been added.

How has it been tested

Black box testing has occurred.

- The project is compared to what the output should be when the project is loaded
 - As this is the same as what is expected all tests were passed
-

Criteria Being met

- Partially : The program generates henna designs
-

Functionality Checklist

- Partially : Does the new design button produce a new design?
-

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

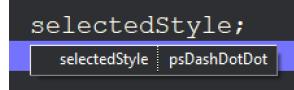
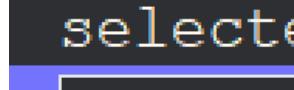
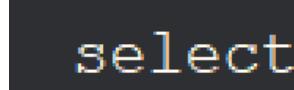
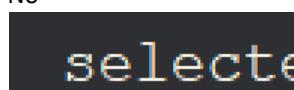
Two templates can now be drawn.

Stage 5 – Linked List

Creating and Testing the Linked List

1. Next I started coding my linked list. This was essential before coding anything else as it would allow a record of the elements used in the template to be kept.
2. Tested adding each element to the linked list

| | |
|---------------|---|
| Line | .basicLine |
| Circle | <p>dashedStroke</p> <p>Wrong - check code</p> <pre>procedure basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY); begin //Uses inbuilt ellipse function with equal x and y distance to make a circle form2.hennaDesign_canvas.canvas.Ellipse(topLeftX,topLeftY,bottomRightX,bottomRightY); addToBinaryTree('dashedStroke',elementsLl,elementsTop,elementsLlSp); end;</pre> |
| | <p>Should add basicCircle to LL instead</p> <pre>procedure basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY); begin //Uses inbuilt ellipse function with equal x and y distance to make a circle form2.hennaDesign_canvas.canvas.Ellipse(topLeftX,topLeftY,bottomRightX,bottomRightY); addToBinaryTree('basicCircle',elementsLl,elementsTop,elementsLlSp); end;</pre> |
| Filled Circle | .basicCircle |
| Dotted circle | basicCircle filledCircle |
| Random stroke | <p>No output - causes program to freeze</p> <p>Commented out linked list code to see if it works</p> <p>Does not freeze always - only error with some types</p> |

| | |
|--|--|
| |  <p>No error</p> |
| |  <p>No error</p> |
| |  <p>No error</p> |
| |  <p>No error</p> |
| | <p>Check output of array after each button click ;</p> <p>Get solid line, dotted line, dashed + dotted added</p> <p>Got rid of unnecessary code - fixed error</p> <p>However when running multiple times didn't work - also notices template one didn't work - however both worked when LL code removed - checked through it</p> |

```

else if linkedList[SP].item > item then
begin
  lastIndexUsed := lastIndexUsed +1;
  linkedList[lastIndexUsed].item := item;
  tempIndex := SP;
  SP := lastIndexUsed;
  linkedList[lastIndexUsed].pointer := SP;
end

```

Item to be added at beginning of list - instead the items are being adding pointer to SP instead of temp which has original SP value

```

else if linkedList[SP].item > item then
begin
  lastIndexUsed := lastIndexUsed +1;
  linkedList[lastIndexUsed].item := item;
  tempIndex := SP;
  SP := lastIndexUsed;
  linkedList[lastIndexUsed].pointer := tempIndex;
end
else
begin

```

Missed out else statement

```

while (placeFound = false) AND (linkedList[p].pointer <> -1) AND (addItem = true) do
begin
  if item = linkedList[linkedList[p].pointer].item then
    begin
      addItem := false;
    end
  else if item > linkedList[linkedList[p].pointer].item then
    begin
      p := linkedList[p].pointer;
    end
  else
    begin
      placeFound := true;
    end;
end;

```

Never exited while loop until it got to end

Works after

circle dots

basicCircle basicLine filledCircle

inner reflection circle

basicCircle basicLine

Circle dots/lines

basicCircle basicLine

- In order to make testing easier, I added a clear array button. This would clear the linked list implementation of the array.

```
procedure TForm2.Button13Click(Sender: TObject);
var index : integer;
begin
for index := 0 to 30 do
begin
  elementsLl[index].pointer := -1;
  elementsLl[index].item := '';
end;
end;
```

I soon realised that this procedure would be needed elsewhere in my code. Hence I made it into an independent procedure.

```
procedure TForm2.Button13Click(Sender: TObject);
begin
  clearLinkedList(30,elementsLl) ;
end;
```

```
procedure clearLinkedList(length : integer; var LL : array of TLinkedListItem);
var index : integer;
begin
for index := 0 to length do
begin
  elementsLl[index].pointer := -1;
  elementsLl[index].item := '';
end;
end;
```

After clicked lots of different elements, I tried the clearLinkedList button . As I had added a debugger point at the end of the procedure I used the data tip to view the values of the linked list. As they were all " and -1 as expected, this button has been tested and worked.

- Next I tested the linked list function for the templates

| Template number | Output | Expected values | Success? |
|-----------------|--|--|--|
| 1 | basicCircle filledCircle basicLine dashedStroke dottedStroke | basicCircle basicLine dashedStroke dottedStroke filledCircle | Whilst for both template the values depicted were correct, they were not in the correct order. |
| 2 | basicLine basicTriangle basicCircle filledCircle | basicCircle basicLine basicTriangle filledCircle | |

a. Check syntax being used

```
procedure TForm2.Button14Click(Sender: TObject);
begin
  if 'filledCircle' > 'basicLine' then
  begin
    labell.Caption := 'works';
  end
  else
  begin
    labell.Caption := 'nope';
  end;
end;
```

As this produced the result 'works' in the caption, the error was not to do with the syntax used.

b. Logic of the code

When I checked the code for the display button it was the following

```
procedure TForm_homepage.Button12Click(Sender: TObject);
var
  i : integer;
begin
  labell.Caption := '';
  for i := 0 to 30 do
  begin
    labell.Caption := labell.Caption + ' ' + elementsLl[i].item;
  end;
end;
```

This meant the code was outputting the linked list in index order - thus this is the order which they are added to the linked list. Initially this would have been fine as the intention of the button was to check the items being added to the linked list. So I decided to write a separate procedure for traversing the linked list.

5. Traversal code

```

procedure Tform_homepage.Button17Click(Sender: TObject);
var
  index,i,p, splitIndex: Integer;
  tempString1, tempString2 : string ;
begin
  i := 0;
  p := elementsLlSp;
  while p <> -1 do
    begin
      for index := 1 to (elementsLl[p].item.Length) do
        begin
          if elementsLl[p].item[index] = upperCase(elementsLl[p].item[index]) then
            begin
              splitIndex := index;
            end;
          end;
          tempString1 := elementsLl[p].item.Substring(1,splitIndex-2) ;
          tempString2 := elementsLl[p].item.Substring(splitIndex-1,elementsLl[p].item.Length-1) ;
          dispalyArray[i].labels.Caption := upperCase( elementsLl[p].item[1] ) + tempString1 + ' ' + tempString2 ;
          p := elementsLl[p].pointer ;
          i := i + 1;
        end;
      while i<=30 do
        begin
          dispalyArray[i].labels.Caption := '';
          i := i +1;
        end;
    end;
end;

```

6. Testing the traversal

| Template number | Output | Expected values | Success? |
|-----------------|--|--|----------|
| 1 | Ked List traversal basicCircle basicLine dashedStroke dottedStroke filledCircle | basicCircle basicLine dashedStroke dottedStroke filledCircle | Yes |
| 2 | basicCircle basicLine basicTriangle filledCircle | basicCircle basicLine basicTriangle filledCircle | Yes |

Stage 5 Review

User Feedback

N/A – this is not an end result but a backend process which needs to take place

Changes

N/A

What has been done

The procedures related to the linked list have been created.

How has it been tested

The contents of the linked list array have been outputted in a caption.

The contents of the linked list have been traversed and outputted in a caption.

Criteria Being met

N/A

Functionality Checklist

N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

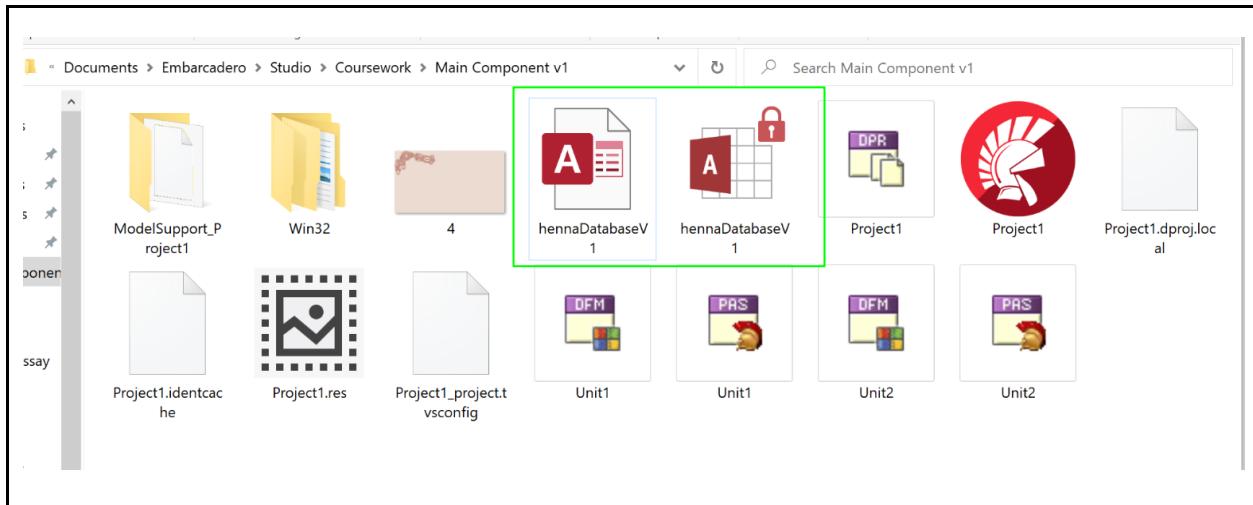
All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

Stage 6 - Database Setup

Creating the file



I created an access file, which I saved in the project folder in order to keep my project file together. This would make it easily accessible in the future.

Populating the Tables

| Element Name | API query | Type | Procedure to | User Rating | Click to Add |
|---------------|------------------|--------|-------------------|-------------|--------------|
| basicCircle | Simple, Circle | circle | basicCircle(topL | 10 | |
| basicLine | Simple, Line | line | basicLine(x1,y1, | 10 | |
| basicTriangle | Simple, Triangle | shapes | basicTriangle(x1 | 10 | |
| dashedStroke | Dashed, Line | stroke | | 10 | |
| dottedStroke | Dash, Line | stroke | | 10 | |
| filledCircle | Filled, Circle | circle | filledCircle(topL | 10 | |
| skeletonLeaf | Veins, Leaf | leaf | skeletonLeaf(to | 10 | |
| * | | | | 0 | |

The screenshot shows a software interface with two tables:

Style Preference Table:

| Category | User Weight | Click to Add |
|----------------|-------------|--------------|
| South Asian | 100 | |
| Middle Eastern | 100 | |
| North African | 100 | |
| * | 0 | |

Templates Table:

| TemplatesID | Template Name | South Asian | North African | Middle Eastern | Click to Add |
|-------------|-------------------|-------------|---------------|----------------|--------------|
| 1 | concentricCircles | 100 | 0 | 50 | |
| 2 | diagonalTriangles | 0 | 100 | 0 | |
| * | (New) | 0 | 0 | 0 | |

(Note - the data in the API query counts as one item as it is all needed to find the YouTube tutorial - hence it doesn't need to be split up to be normalised)

I created the tables and then added the appropriate values, as per my design.

Stage 6 Review

User Feedback

N/A – this is not an end result but a backend process which needs to take place

Changes

N/A

What has been done

The database has been created.

The database has been populated.

How has it been tested

Compared to the design and matches.

Criteria Being met

N/A

Functionality Checklist

N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

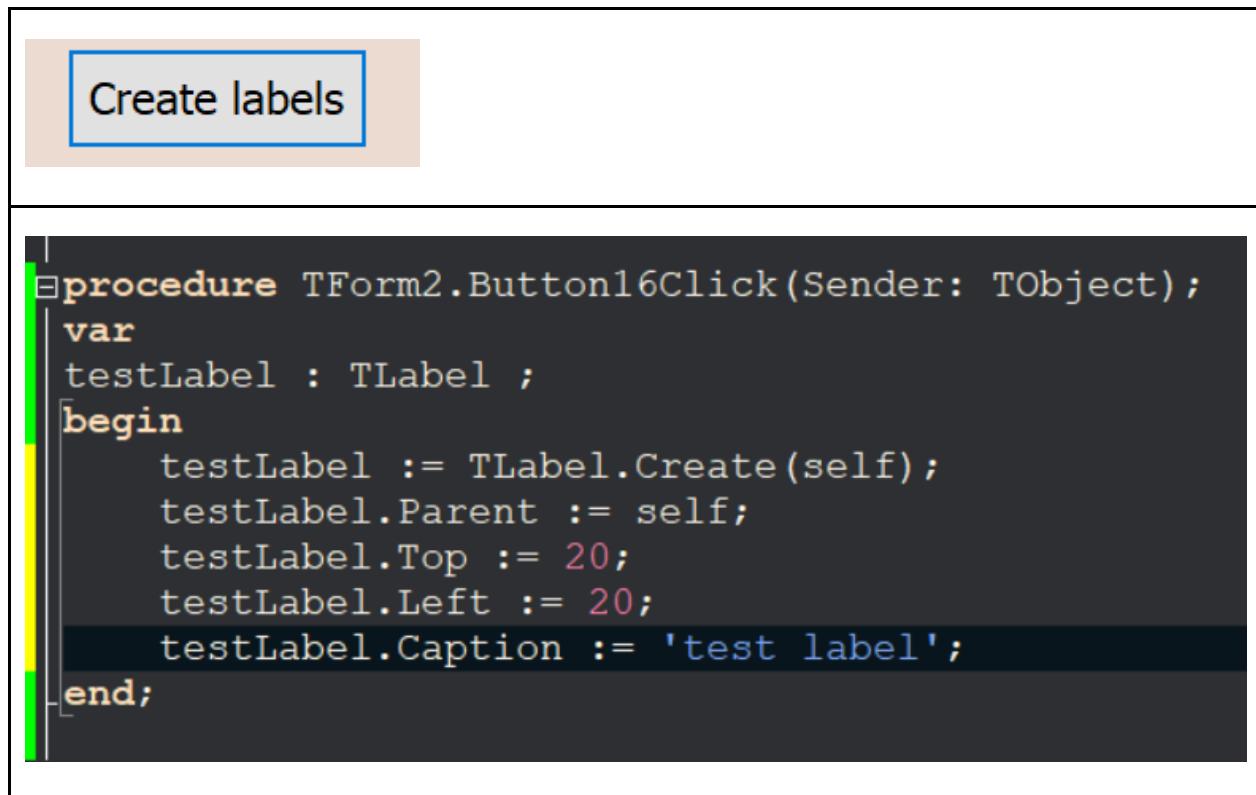
The linked list is functional.

The henna database now exists.

Stage 7 – Displaying the Elements

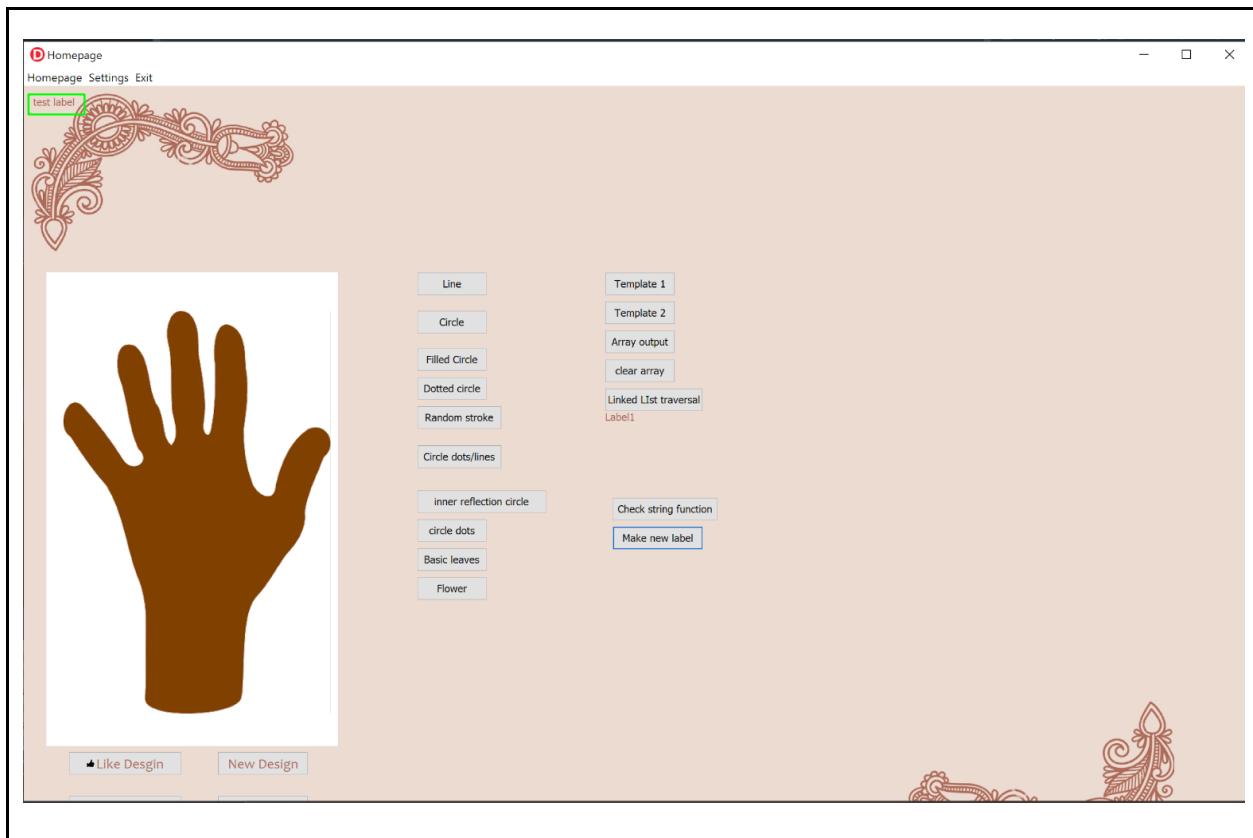
Labels

Initially I wrote the code to dynamically create a label through code. This code creates a single label at the coordinates (20,20) with the text “test label”.



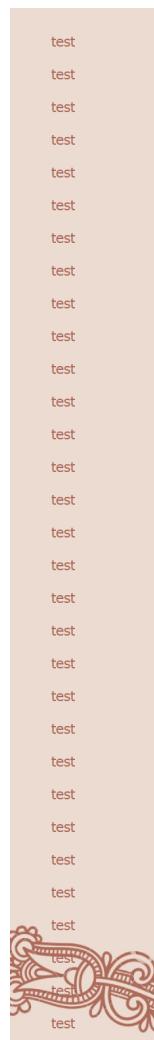
The screenshot shows a Delphi IDE interface. At the top, there is a button labeled "Create labels". Below it, the code editor displays the following Pascal code:

```
procedure TForm2.Button16Click(Sender: TObject);
var
  testLabel : TLabel ;
begin
  testLabel := TLabel.Create(self);
  testLabel.Parent := self;
  testLabel.Top := 20;
  testLabel.Left := 20;
  testLabel.Caption := 'test label';
end;
```



As this was successful, I decided to repeat the process to populate all the labels of my display labels. Hence I looped through all of my label array and created a label in each index.

```
var
i,j : integer;
begin
for i := 0 to 30 do
begin
  labels[i] := TLabel.Create(self);
  labels[i].Parent := self;
  labels[i].Top := 100 + j;
  labels[i].Left := 2000;
  j := 50 + j;
  labels[i].Caption := 'test' ;
end;
end;
```



As this was successful, I proceed to add the code for populating the labels and display the elements. First this meant needing to traverse the linked list. Hence in the code below it looks at the first item in the linked list. It then gets the data item and splits it into the two words which the element name consists of and gets it in an English format. The it then assigns the first label in the array's caption to the element item. The next index is changed to the pointer of the linked list item. Hence this process is repeated for the next index, and then for all items until the next index is "-1" informing us we have reached the end of the linked list.

```

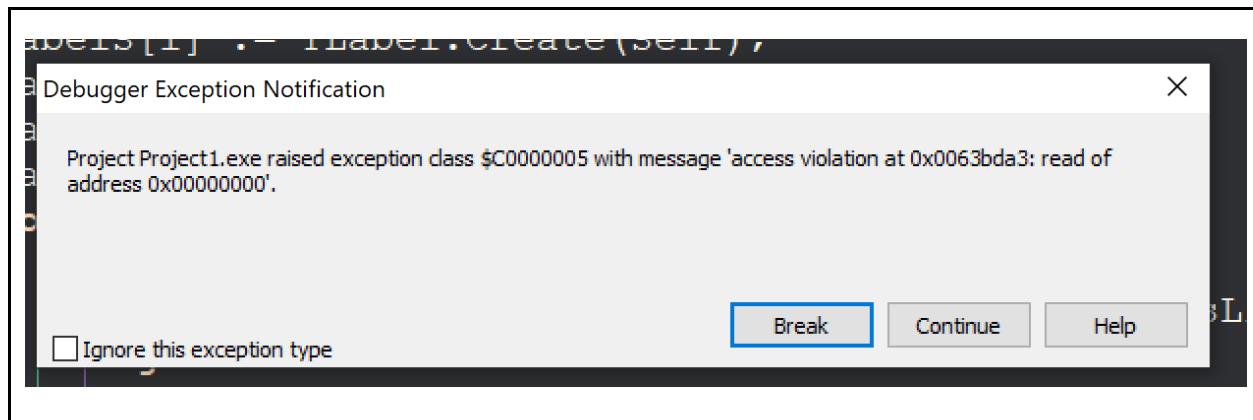
procedure TForm2.Button17Click(Sender: TObject);
var
  index,i,p, splitIndex: Integer;
  tempString1, tempString2 : string ;
begin
  i := 0;
  p := elementsLlSp;
  while p <> -1 do
    begin
      for index := 1 to (elementsLl[p].item.Length) do
        begin
          if elementsLl[p].item[index] = upperCase(elementsLl[p].item[index]) then
            begin
              splitIndex := index;
            end ;
          end;
        tempString1 := elementsLl[p].item.Substring(1,splitIndex-2) ;
        tempString2 := elementsLl[p].item.Substring(splitIndex-1,elementsLl[p].item.Length-1) ;
        labels[i].Caption := upperCase( elementsLl[p].item[1]) + tempString1 + ' ' + tempString2 ;
        p := elementsLl[p].pointer ;
        i := i + 1;
      end;
    end;

```

Testing Label Population :

| Template number | Linked list transversal | Labels | Successful ? |
|-----------------|--|---|--------------|
| 1 | basicCircle basicLine dashedStroke dottedStroke filledCircle | Basic Circle Basic Line Dashed Stroke Dotted Stroke Filled Circle | Yes |
| 2 | basicCircle basicLine basicTriangle filledCircle | N/A | No |

When the labels were clicked for the second template, this exception was raised:



Just to ensure that the issue was not with the contents of the array, I attempted to clear the array. However this made no difference.

Thus the next thing I checked was the clear linked list function and checked the code for this. Here I realised that the start pointer for the linked list was never reset to -1, thus this interfered with adding a second set of elements to the linked list.

```
procedure clearLinkedList(length : integer; var LL : array of TLinkedListItem);
var index : integer;
begin
elementsLlSP := -1;
for index := 0 to length do
begin
  elementsLl[index].pointer := -1;
  elementsLl[index].item := '';
end;
```

Testing Label Population Fix :

| Template number | Linked list transversal | Labels | Successful ? |
|-----------------|---|---|--------------|
| 1 | <code>basicCircle basicLine dashedStroke dottedStroke filledCircle</code> | Basic Circle Basic Line Dashed Stroke Dotted Stroke Filled Circle | Yes |
| 2 | <code>basicCircle basicLine basicTriangle filledCircle</code> | Basic Circle Basic Line Basic Triangle Filled Circle | Yes |

Stage 7 Review

User Feedback

- The font colour is really easy to read as it contrasts well

Changes

- Start pointer value reset
- Element name split up to be displayed

What has been done

Labels have been created.

The elements used in a template have been displayed by populating the labels.

How has it been tested

- Comparing the expected values with the actual values

Criteria Being met

N/A

Functionality Checklist

- Are the elements used displayed?

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

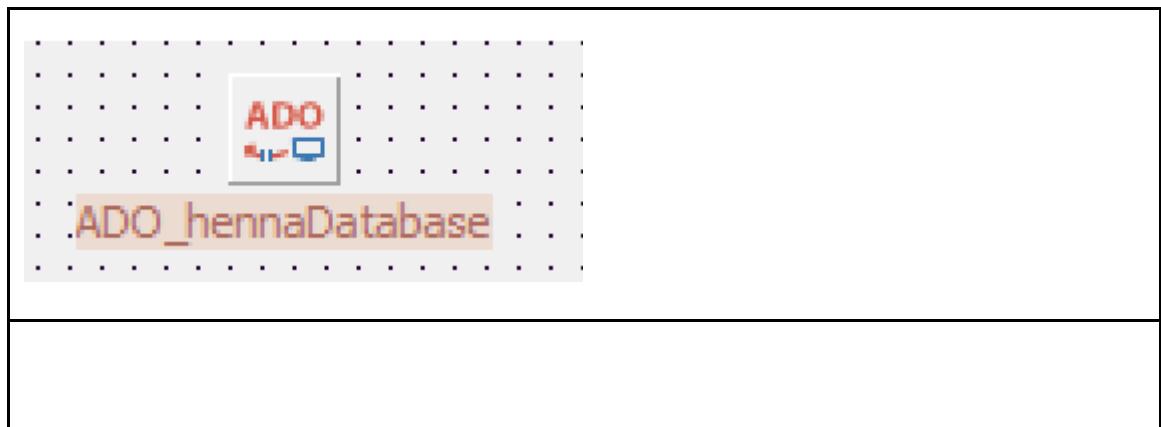
The henna database now exists.

The elements used in the templates are now displayed.

Stage 8 - Linking database with code

To link the database into my Delphi program I did the following

1. I created an ADO connection element

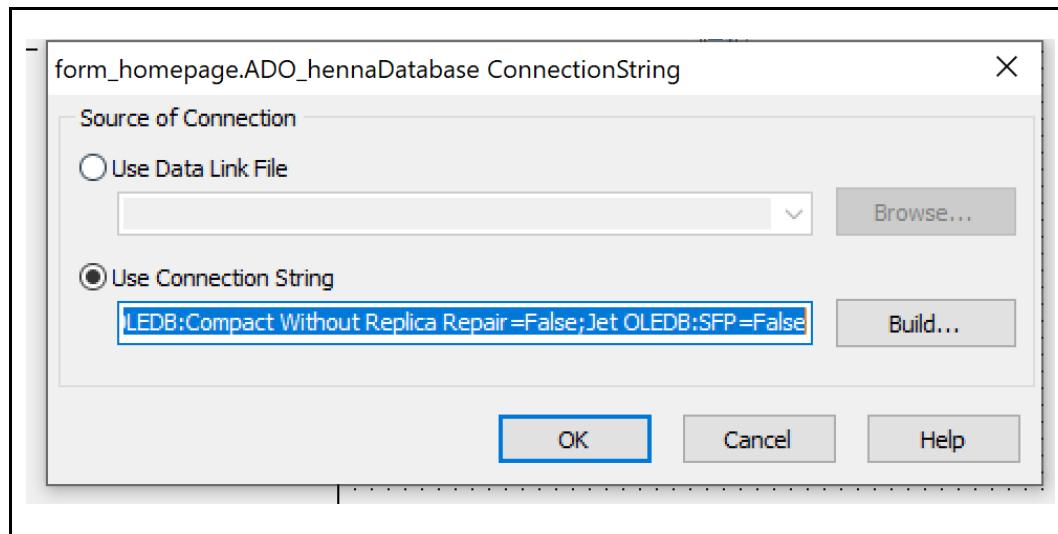


- a. I tried to connect to the database via a test connections string. However the database did not appear in the file directory which was opened by the windows popup.
- b. Hence to try and debug this error I duplicated the file and renamed it. However the file still did not appear in the directory.
- c. Thus I googled this issue and found the following link :

<https://www.addintools.com/documents/access/where-is-mdb-save-as.html#:~:text=In%20the%20middle%20area%2C%20click,your%20database%20as%20MDB%20files.>

Hence when I read the webpage I realised that the database had to be in a MBD format in order to be compatible with Delphi

- d. So I saved the file in the MBD format. It now appeared in the directory and I created a Jet connection string with it



- e. I set a blank password, hence making the database freely accessible without the user having to provide a password

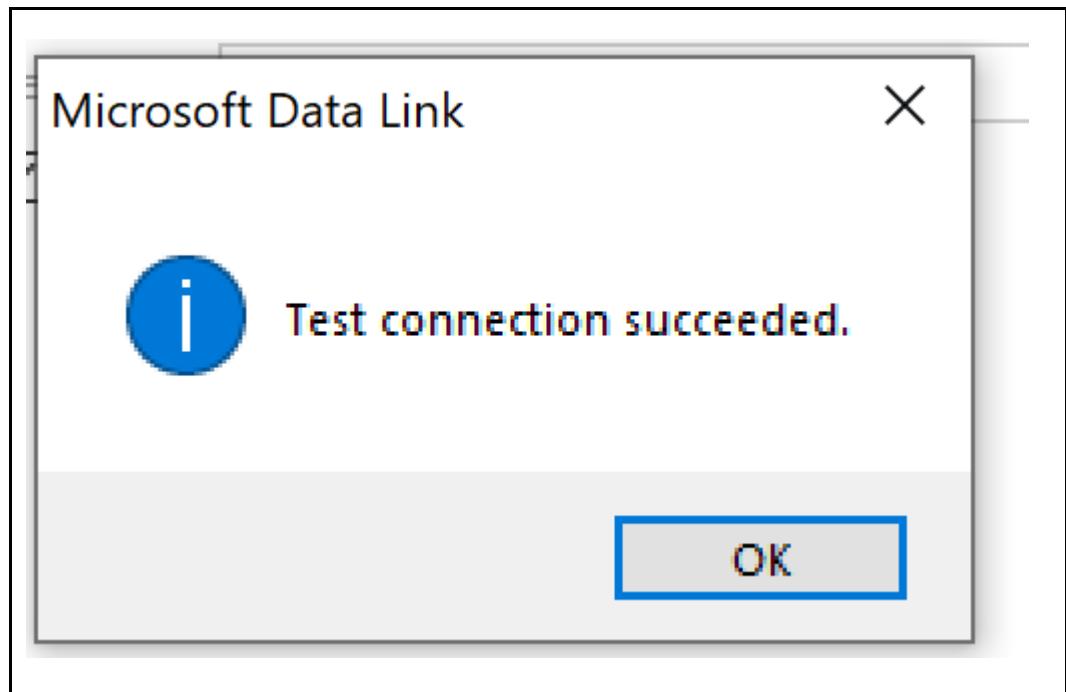
2. Enter information to log on to the database:

User name:

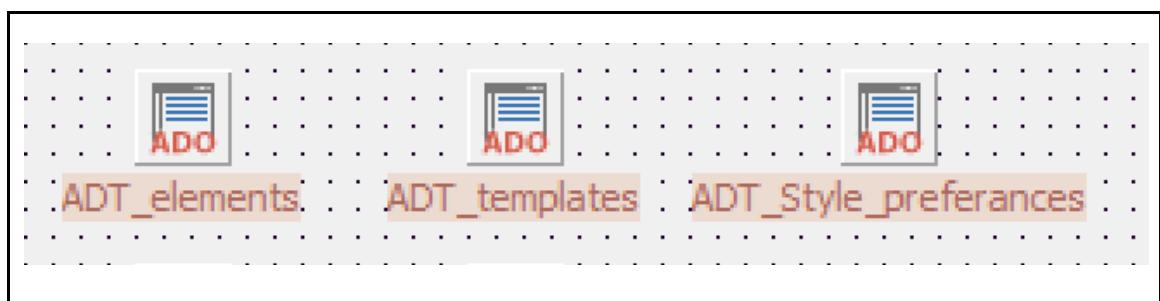
Password:

Blank password Allow saving password

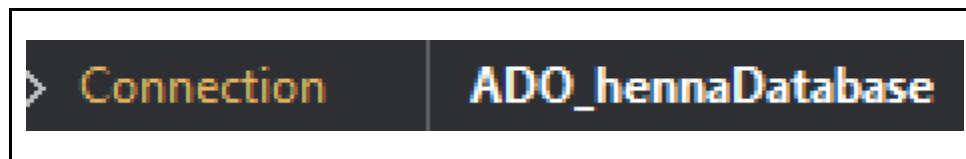
- f. I tested the connection to ensure the link was working. This was successful



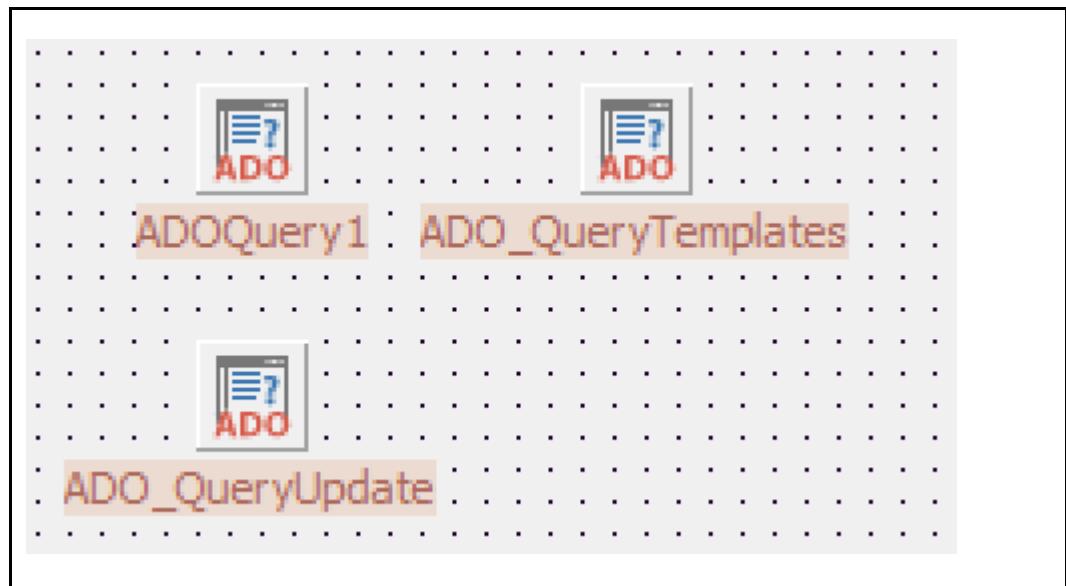
2. I then set up ADO table elements for all the tables in my database via the following steps



- a. I changed their connection to ADO_hennaDatabase



- b. I then set their table to the corresponding one from the database
3. I then set up ADO_Qualifiers for retrieving data from the database, which I will use further on in my code
- a.



I then connected all of them to ADO_hennaDatabase.

Stage 8 Review

User Feedback

- N/A – backend process so no testable results

Changes

- Format of access database changed

What has been done

The access database has been linked to the Delphi code.

How has it been tested

- Connection is successful

Criteria Being met

N/A

Functionality Checklist

N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

Stage 9 - Youtube API

Initially I attempted to select all record with the corresponding element name, and get the APIQuery from it

```
procedure TForm2.Button19Click(Sender: TObject);
begin
  with ADOQuery1 do
  begin
    Active := false;
    SQL.Clear;
    SQL.Add('SELECT APIquery FROM Elements WHERE ElementName = "basicCircle" ;');
    Active := true;
  end;
end;
```

However as we know there is only one record of interest, it made more sense to get the data from this particular record instead.

```
procedure TForm2.Button20Click(Sender: TObject);
begin
  with adt_elements do
  begin
    open ;
    if fieldvalues['ElementName'] = 'basicCircle' then
    begin
      label2.Caption:= fieldvalues['APIquery'];
    end;
  end;
end;
```

Query v2

Simple, Circle

When testing this it gave us the expected value from the database and displayed it in the label.

I then applied this to all of the elements used in a template.

```
procedure queryLLItems();
var
  i,p : integer;
begin
  with form2.adt_elements do
    begin
      open ;
      p := elementsLlSp;
      while p <> -1 do
        begin
          if fieldvalues['ElementName'] = elementsLl[p].item then
            begin
              form2.label2.Caption:= form2.label2.Caption + fieldvalues['APIquery'];
            end;
          p := elementsLl[p].pointer;
        end;
    end;
end;
```

However when testing this, it did not go as expected. For both template 1 and 2 , only the following values were returned and added to the label.

Label2Simple, Circle

This was the result produced. Hence it was apparent that only the first item was being retrieved from the database.

Hence I used a variable watch and a breakpoint to keep track of the variables, however they all gave the expected results.

```

  i,p : integer;
begin
  with form2.adt_elements do
    begin
      open ;
      p := elementsLlSp;
      while p <> -1 do
        begin
          if fieldvalues['ElementName'] = elementsLl[p].item then
            begin
              form2.label3.caption := form2.label3.caption + 'Called';
              form2.label2.Caption:= form2.label2.Caption + fieldvalues['APIquery'];
            end;
          p := elementsLl[p].pointer;
        end;
    end;
end;

```

Thus I suspected it was a logic error, and added the word called to a label every time the if statement was entered. As the word called was only entered once it became apparent that there was an issue with the if statement.

```

procedure queryLLItems();
var
  i,p : integer;
begin
  with form2.adt_elements do
    begin
      open ;
      p := elementsLlSp;
      while p <> -1 do
        begin
          if fieldvalues['ElementName'] = 'basicLine' then
            begin
              form2.label2.Caption:= form2.label2.Caption + fieldvalues['APIquery'];
            end;
          p := elementsLl[p].pointer;
        end;
    end;
end;

```

To figure out if the issue was with accessing the element from the linked list or retrieving the data from the database I hard coded in “basic line” to see if this produced a different result. However it made no difference, thus the issue must be with retrieving the data from the database.

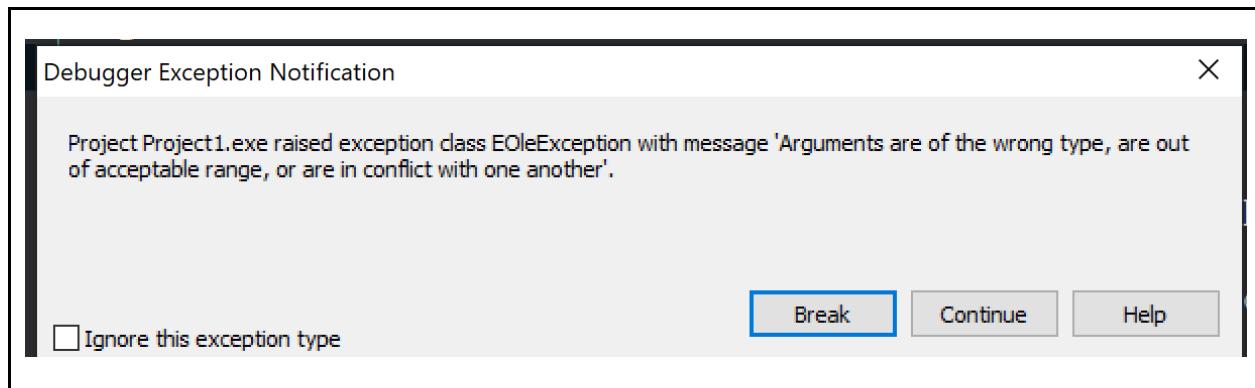
I then realised that the record number for the record number being accessed in the database needs to be specified. Thus I modified the code to include a for loop which iterate through all the records for each linked list element accessed.

```

procedure queryLLItems();
var
i,p : integer;
begin
with form2.adt_elements do
begin
open ;
p := elementsLlSp;
while p <> -1 do
begin
for i := 1 to recordcount do
begin
recno := i;
if fieldvalues['ElementName'] = elementsLl[p].item then
begin
form2.label2.Caption:= form2.label2.Caption + fieldvalues['APIquery'];
end;
end;
p := elementsLl[p].pointer;
end;
end;
end;

```

Initially the for loop was set from $i = 0$ to $recordcount-1$, as i assumed it had a zero indexing. However this raised the following error:



Hence when I changed it to the code above it got rid of the error.

Testing :

| Template number | Database API revival | Elements | Database API query data | Successful ? |
|-----------------|--|---|--|--------------|
| 1 | Simple, CircleSimple, LineDashed, LineDotted, LineFilled | Basic Circle Basic Line Dashed Stroke Dotted Stroke Filled Circle | Simple, Circle Simple, Line Dashed, Line Dotted, Line Filled, Circle | Yes |
| 2 | Simple, CircleSimple, LineSimple, TriangleFilled | Basic Circle Basic Line Basic Triangle Filled Circle | Simple, Circle Simple, Line Simple, Triangle Filled, Circle | Yes |

Whilst the formatting of the all the API query results from the database may appear strange to the human eye, this doesn't matter in the program as it is going to directly be used in the YouTube API. Thus it doesn't need to be in a human presentable format.

Initially when calling the YouTube API I tried the following code:

```

var
tempString,test : string;
tempPWide : PWideChar;
begin
begin
tempPWide := 'https://www.youtube.com/results?search_query=henna' + '+henna+tutorial';
shellexecute(Application.Handle,'open',tempPWide,nil,nil,0);
end;
end;
```

This was successful and opened a YouTube tutorial with the query “henna +henna+tutorial”

However when this code was changed to the URL above + the caption of the first display label it gave an error about the formation of the URL. Hence I tried converting it to type PChar as required.

However the same error persisted. Thus I watched the crash dump values of the variables when the error was raised. At this point I noticed the index being accessed in the array was a massive one, and thus should be invalid. Hence I decided to fix this first, then come back to my other error.

```
procedure TForm2.youtubeAPI(Sender : TObject);
begin
  with (Sender as TButton) do
    shellexecute(Application.Handle,'open',PChar('https://www.youtube.com/results?search_query=henna+henna+Tutorial' + caption ),nil,nil,0);
end;
```

```
procedure TForm2.Button22Click(Sender: TObject);
begin
var
  i,j: integer;
begin
  j := 0;
  for i := 0 to 30 do
  begin
    dispalyArray[i].button := TButton.Create(Self);
    with dispalyArray[i].button DO
    begin
      Visible := true;
      Caption := dispalyArray[i].labels.Caption;
      Parent := Self;
      Height := 23;
      Width := 100;
      Left := 2200;
      Top := 100 + j;
      if dispalyArray[i].labels.Caption <> '' then
      begin
        tempIndex := i;
        OnClick := youtubeAPI;
      end;
    end;
    //  dispalyArray[i].button.OnClick := shellexecute(handle,'open','http://');
    //try assinging it to an invisible buttons code
    //  dispalyArray[i].button.OnClick := (Handle, 'open', PChar('http://'));
    //  dispalyArray[i].button.Click := shellexecute(handle,'open','http://');

    j := 50 + j;
  end;
```

```

procedure TForm1.Button1Click(Sender: TObject);
private
  { Private declarations }
  procedure youtubeAPI(Sender : TObject);
public

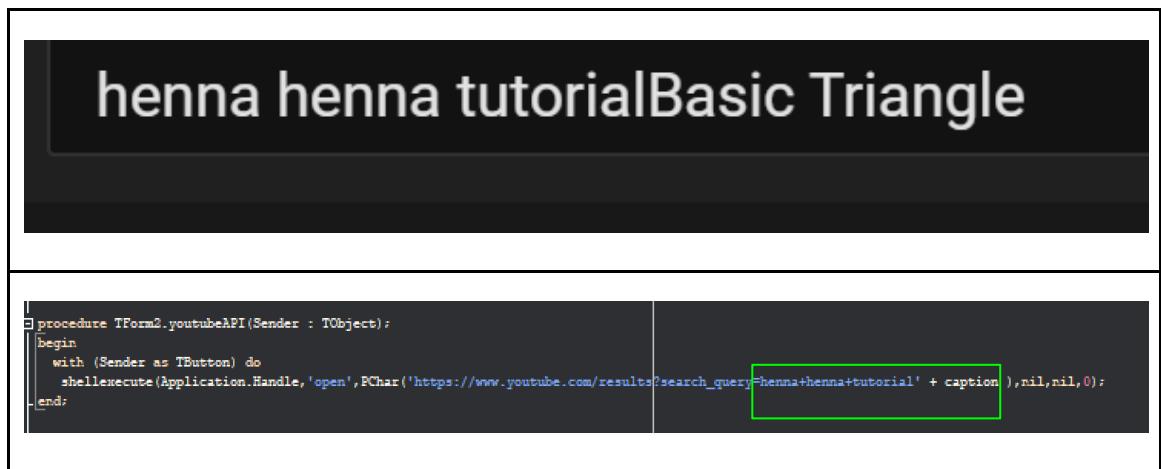
```

I decided it made more sense to also change the button caption to the API query when the button was created. Hence now when the button is clicked the line above is executed. This was a more efficient version as it did not require accessing any other objects, hence also reduced the chance of errors. This worked without any issues and opened the corresponding YouTube page. Hence when the button is created, its onClick event is sent to youtubeAPI.

Delphi flagged up an error with this assignment. Thus I added a private deceleration for youtubeAPI procedure at the top of the code. Hence this was now a valid assignment for the button action, thus the error was solved.

User testing:

- The word henna is repeated in the query and there is no gap between tutorial and the element name



Hence I checked the URL which was queries and found it was due to the words and formatting of the query. Thus I modified this and it fixed the error.

Stage 9 Review

User Feedback

- The word henna is repeated in the query and there is no gap between tutorial and the element name
 - This is so cool !
 - The tutorials are really helpful and make learning henna more manageable
-

Changes

- Query URL modified
 - Button caption is changed to API query
 - SQL statements broken up
-

What has been done

Buttons have been created which take the user to tutorials for the desired element.

How has it been tested

- YouTube tutorials opened for all elements displayed successfully
 - Buttons displaying expected captions
-

Criteria Being met

- Elements being linked to YouTube tutorials
-

Functionality Checklist

- Can you go to the relevant YouTube pages for the elements?
-

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

Stage 10 - saving canvas

```
procedure TForm1.Homepage.Bitbutton1Click(Sender: TObject);
var
  Bitmap: TBitmap;
  Source: TRect;
  Dest: TRect;
begin
  Bitmap := TBitmap.Create;
  try
    with Bitmap do
      begin
        Width := hennaDesign_canvas.Width;
        Height := hennaDesign_canvas.Height;
        Dest := Rect(0, 0, Width, Height);
      end;
    with hennaDesign_canvas do
      Source := Rect(0, 0, Width, Height);
    Bitmap.Canvas.CopyRect(Dest, hennaDesign_canvas.Canvas, Source);
    Bitmap.SaveToFile('MYFILE.BMP');
  finally
    Bitmap.Free;
  end;
end;
```

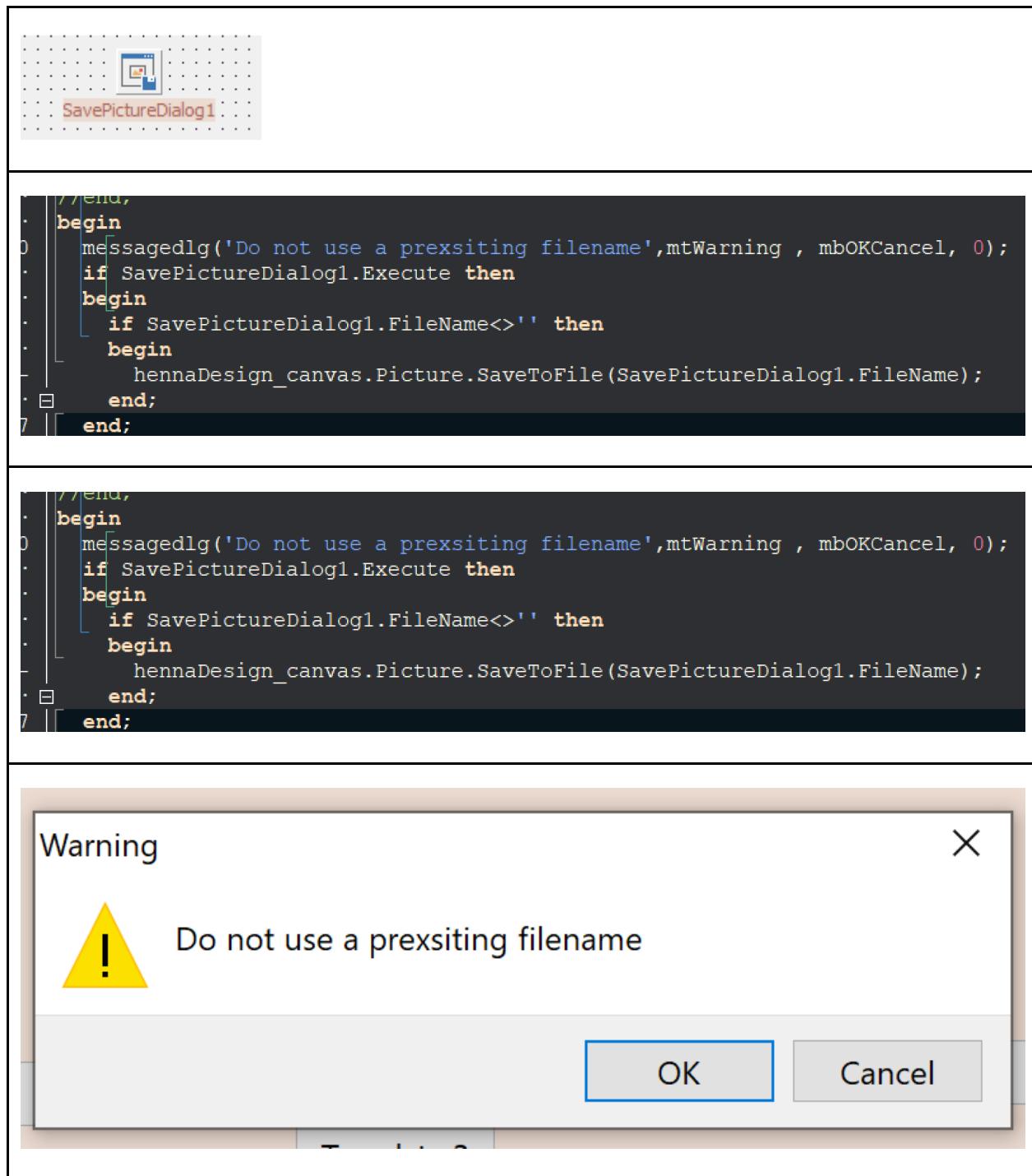
This code worked successfully and saved the file in the computer directory as “MYFILE.BMP”.

User Testing:

Whilst this worked fine technically, when given to users to test they raised several points.

- The save option only saved one canvas, as it was overwritten the next time the button was clicked.
- The user was not given an option of where to save the file.
- The file was saved in the win folder , hence hard to find especially for non-technical people.

Modified Code



The screenshot shows a Delphi IDE interface with three stacked sections. The top section displays a component palette icon for 'SavePictureDialog1'. The middle section shows a portion of the Delphi code:

```
// end;
begin
  messagedlg('Do not use a presxiting filename',mtWarning , mbOKCancel, 0);
  if SavePictureDialog1.Execute then
  begin
    if SavePictureDialog1.FileName<>'' then
    begin
      hennaDesign_canvas.Picture.SaveToFile(SavePictureDialog1.FileName);
    end;
  end;
```

The bottom section shows the same code with a small change in the first line:

```
// end;
begin
  messagedlg('Do not use a presxiting filename',mtWarning , mbOKCancel, 0);
  if SavePictureDialog1.Execute then
  begin
    if SavePictureDialog1.FileName<>'' then
    begin
      hennaDesign_canvas.Picture.SaveToFile(SavePictureDialog1.FileName);
    end;
  end;
```

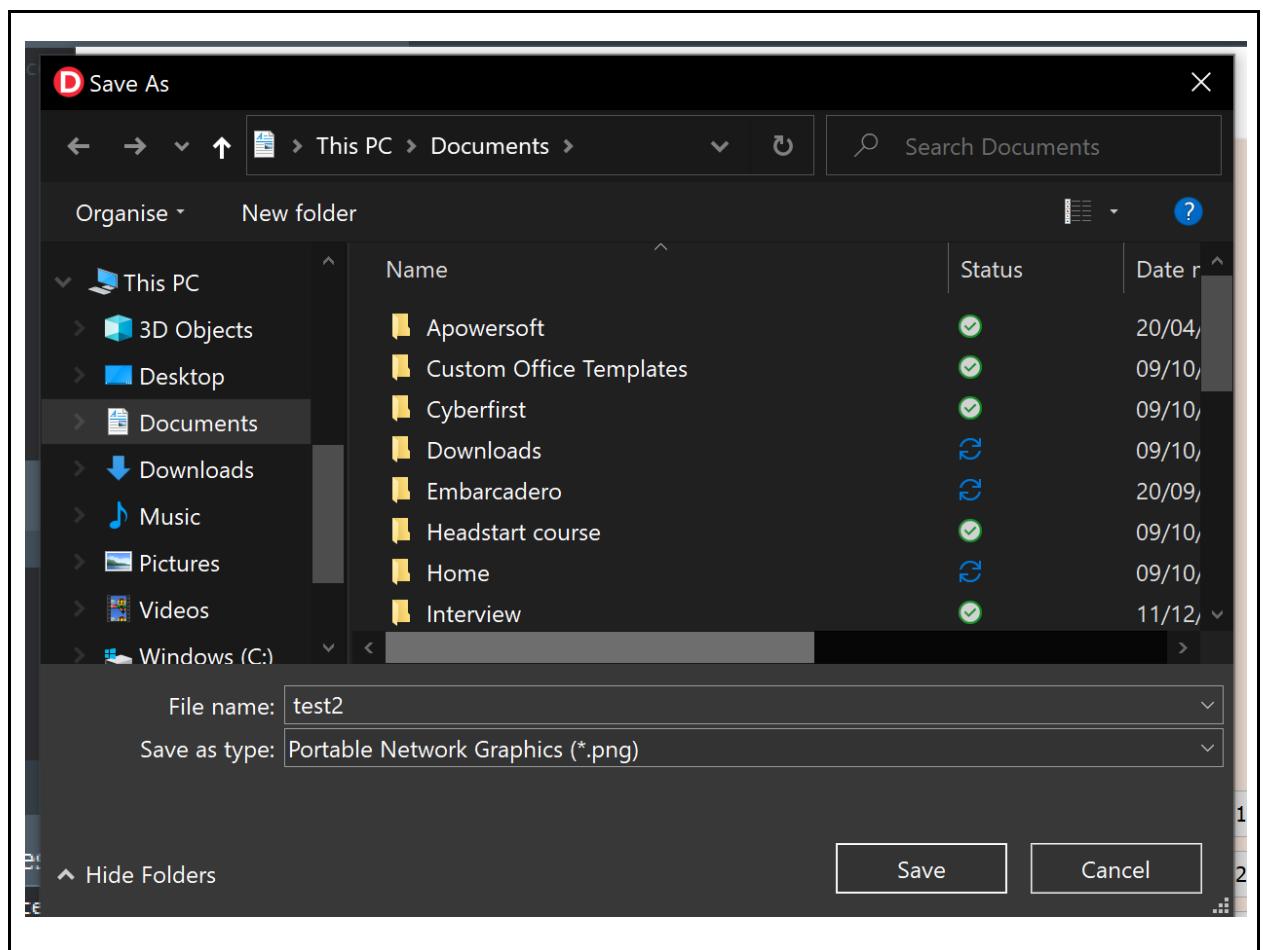
Below the code is a screenshot of a Windows 'Warning' dialog box. The title bar says 'Warning'. The message area contains a yellow exclamation mark icon and the text 'Do not use a presxiting filename'. At the bottom are 'OK' and 'Cancel' buttons.

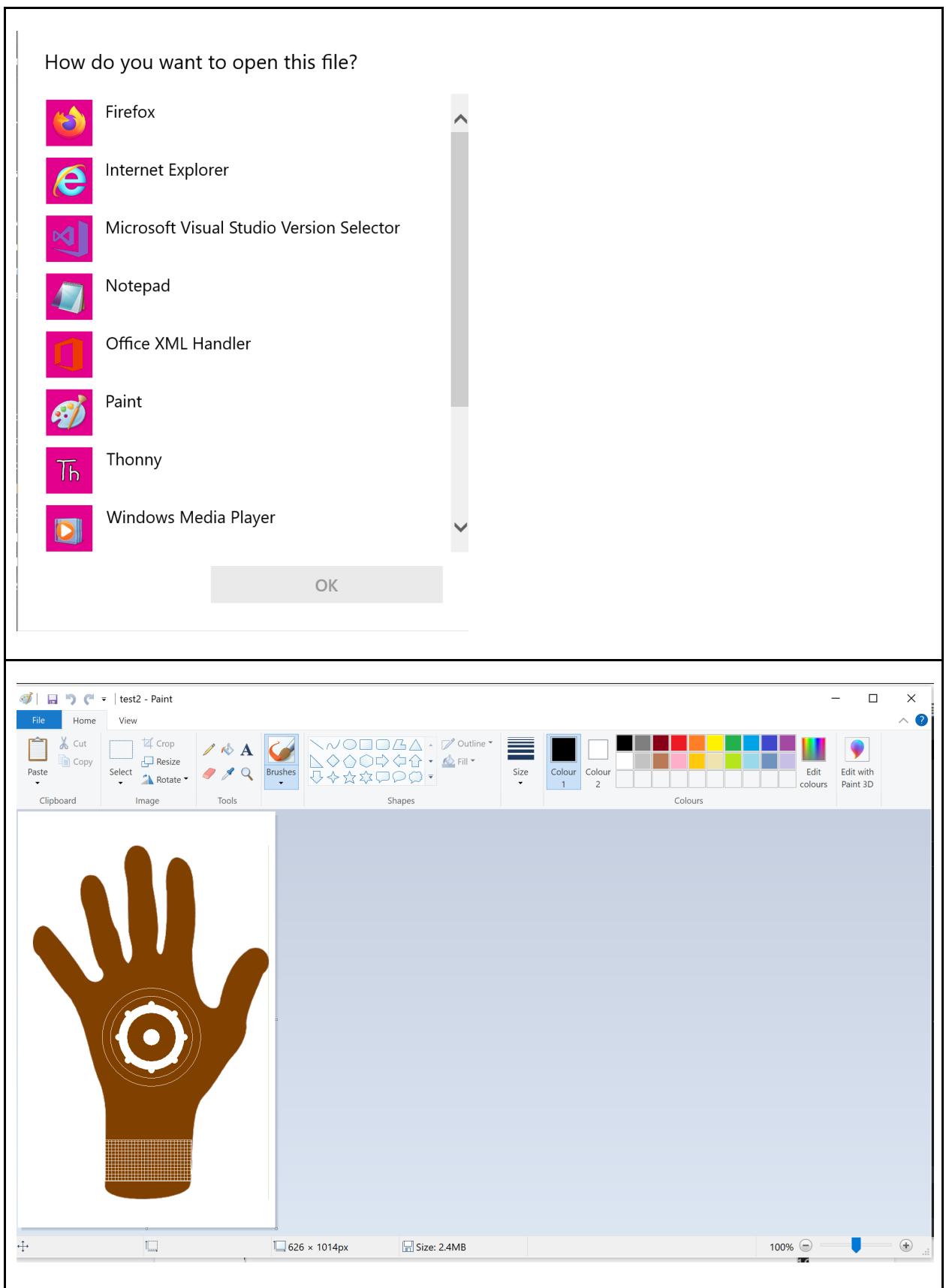
The code now displayed a warning, telling the user to use a pre editing file name. This precondition means that a file is not overwritten. Initially this was in the wrong order, the user first picked the file save location and name

and was then displayed with the warning. However once the warning line position was moved outside the if statement the logical error was resolved

A dialogue box appears which allows the user to graphically pick a location to save the file in. This is preferable for most users, as very few would know how to do this through the traditional method of a file path.

I added a save picture component to enable the canvas to be saved to the user's computer.





When the file was clicked on the user was given the following options to open the file in. It then allows the user to open the file in paint.

User Testing 2:

All the users were fine with this saving method and reacted as expected, so no further changes were needed.

Stage 10 Review

User Feedback

- The original method for saving is complicated for a user
- File names are overwritten
- Saves in odd location

Changes

- Warning added to prevent overwriting a pre-existing file
- Location to save obtained through standard windows dialogue box
- User friendly saving method used

What has been done

The save button has been given functionality.

How has it been tested

- User testing
- Module testing

Criteria Being met

- Ability to save the designs in an image format

Functionality Checklist

- Can you save designs?

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

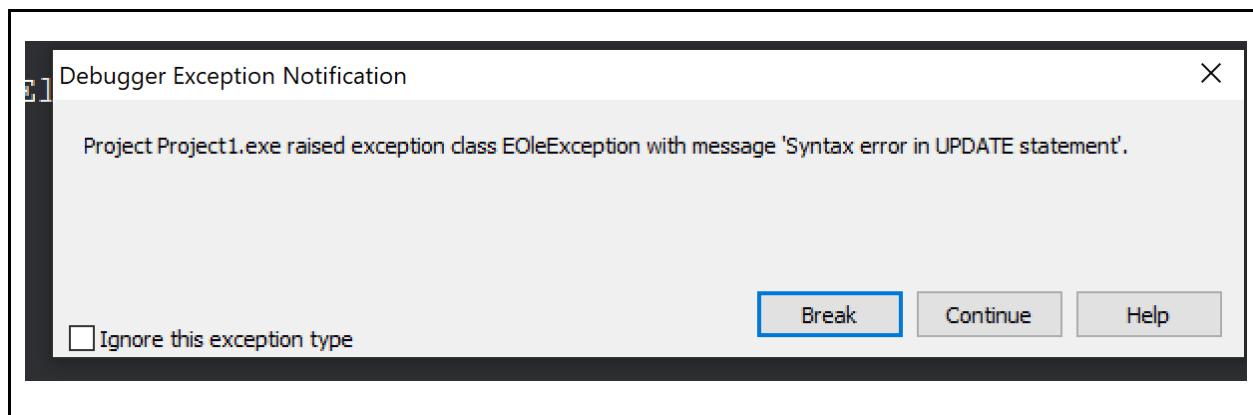
Templates can be saved to your computer.

Stage 11 - Liking and disliking designs

```
procedure TForm2.likeDeign_btnClick(Sender: TObject);
begin
begin
  with ADOQuery1 do
  begin
    Active := false;
    SQL.Clear;
    SQL.Add('UPDATE Elements');
    SQL.Add('SET User Rating=(UserRating*1.1) WHERE ElementName=basicLine');
    ExecSQL
  end;
  ADT_Elements.Refresh;
end;
end;
```

```
procedure TForm2.likeDeign_btnClick(Sender: TObject);
begin
begin
  with ADO_QueryUpdate do
  begin
    Active:=False;
    SQL.Clear;
    SQL.Add('UPDATE Elements');
    SQL.Add('SET UserRating = 11 WHERE ElementName = "basicLine"');
    // SQL.Add('SET User Rating=11 WHERE ElementName=basicLine');
    ExecSQL ;
  end;
  ADT_Elements.open;
  ADT_Elements.Refresh;
end;
```

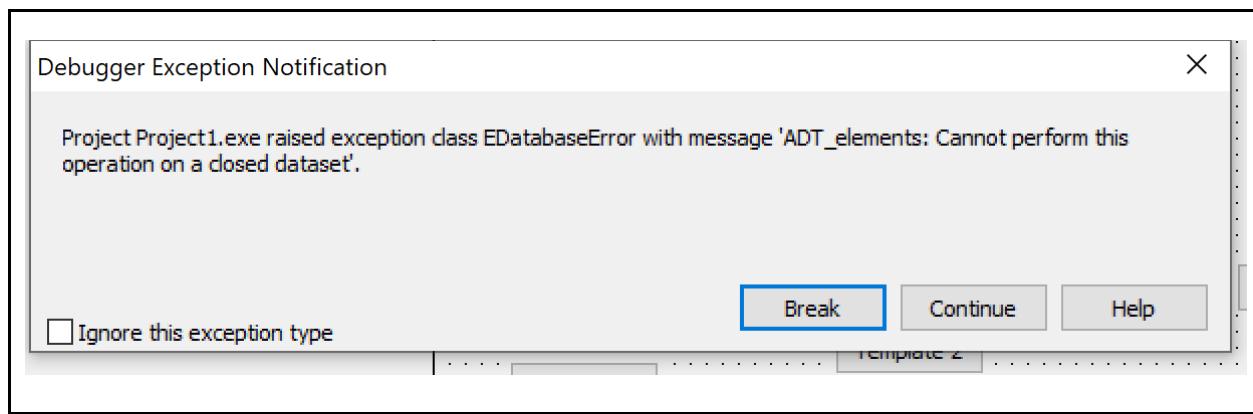
I wrote the following code in order to update the elements rating when the like button was clicked. It had the intention of updating the elements table and changing the UserRating by a multiplier of 1.1 where the ElementName is basicLine. However the first issue encountered was a syntax one:



I realised there was a gap in the User Rating, which was causing this. Hence when the space was removed from "User Rating" this error was mitigated.

The next error received was ElementValue has no default value. I realised this was as ADO_hennaDatabase was connected to an old version of the database. Thus when I re did the connection string process this error was fixed.

The next error was :



Hence I added the line ADT_Elements.open to allow changes to the dataset.

Testing :

When the button was clicked the default value of 10 is expected to be changed to $10 \times 1.1 = 11$

| ElementName | APIquery | Type | UserRating | Click to Add |
|---------------|------------------|--------|------------|--------------|
| basicCircle | Simple, Circle | circle | 10 | |
| basicLine | Simple, Line | line | 10 | |
| basicTriangle | Simple, Triangle | shapes | 10 | |
| dashedStroke | Dashed, Line | stroke | 10 | |
| dottedStroke | Dotted, Line | stroke | 10 | |
| filledCircle | Filled, Circle | circle | 10 | |
| skeletonLeaf | Veins, Leaf | leaf | 10 | |
| solidStroke | Solid, Line | stroke | 10 | |
| * | | | 0 | |

| ElementName | APIquery | Type | UserRating | Click to Add |
|---------------|------------------|--------|------------|--------------|
| basicCircle | Simple, Circle | circle | 10 | |
| basicLine | Simple, Line | line | 11 | |
| basicTriangle | Simple, Triangle | shapes | 10 | |
| dashedStroke | Dashed, Line | stroke | 10 | |
| dottedStroke | Dotted, Line | stroke | 10 | |
| filledCircle | Filled, Circle | circle | 10 | |
| skeletonLeaf | Veins, Leaf | leaf | 10 | |
| solidStroke | Solid, Line | stroke | 10 | |
| * | | | 0 | |

Hence this test was successful.

Then my plan was to apply this to all the element used in the linked list. As the order does not matter the linked list does not need to be traverse, instead you can just loop through all the indexes of the linked list and change their rating.

```

procedure TForm2.likeDeign_btnClick(Sender: TObject);
var
currentElement : string ;
i : integer ;
begin
begin
for i := 0 to 30 do
begin
currentElement := elementsLl[i].item;
with ADO_QueryUpdate do
begin
Active:=False;
SQL.Clear;
SQL.Add('UPDATE Elements');
SQL.Add('SET UserRating = (UserRating*1.1) WHERE ElementName = "' + currentElement + '"');
// SQL.Add('SET User Rating=11 WHERE ElementName=basicLine');
ExecSQL ;
end;
ADT_Elements.open;
ADT_Elements.Refresh;
end;
end;

```

```
SQL.Add('SET UserRating = (UserRating*1.1) WHERE ElementName = currentElement');
```

The line selected above was causing an error. The whole of the SQL statement needs to be in single quote marks, and as long as it was not in double quotes it would not be interpreted as a string, thus the modification made worked.

Whilst trying to debug the error I also attempted the following code :

```

with ADT_Elements do
begin
if ADT_Elements.Locate('ElementName', 'basicLine', []) = true then
begin
ADT_Elements.Edit ;
ADT_Elements['UserRating'] := '11';
ADT_Elements.Post;
end;
end;

```

However this seemed unnecessarily complicated, and I soon ditched the idea.

I tried this with multiple clicks when template 2 was displayed.

| Template elements | Database after multiple dislikes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|-------------|------------|--------------|------------|--------------|-------------|----------------|--------|---|--|-----------|--------------|------|---|--|---------------|------------------|--------|---|--|--------------|--------------|--------|----|--|--------------|--------------|--------|----|--|--------------|----------------|--------|---|--|--------------|-------------|------|----|--|-------------|-------------|--------|----|--|---|--|--|---|--|
| <p>Basic Circle</p> <p>Basic Line</p> <p>Basic Triangle</p> <p>Filled Circle</p> | <table border="1"> <thead> <tr> <th>ElementName</th><th>APIquery</th><th>Type</th><th>UserRating</th><th>Click to Add</th></tr> </thead> <tbody> <tr> <td>basicCircle</td><td>Simple, Circle</td><td>circle</td><td>4</td><td></td></tr> <tr> <td>basicLine</td><td>Simple, Line</td><td>line</td><td>4</td><td></td></tr> <tr> <td>basicTriangle</td><td>Simple, Traingle</td><td>shapes</td><td>4</td><td></td></tr> <tr> <td>dashedStroke</td><td>Dashed, Line</td><td>stroke</td><td>10</td><td></td></tr> <tr> <td>dottedStroke</td><td>Dotted, Line</td><td>stroke</td><td>10</td><td></td></tr> <tr> <td>filledCircle</td><td>Filled, Circle</td><td>circle</td><td>4</td><td></td></tr> <tr> <td>skeletonLeaf</td><td>Veins, Leaf</td><td>leaf</td><td>10</td><td></td></tr> <tr> <td>solidStroke</td><td>Solid, Line</td><td>stroke</td><td>10</td><td></td></tr> <tr> <td>*</td><td></td><td></td><td>0</td><td></td></tr> </tbody> </table> | ElementName | APIquery | Type | UserRating | Click to Add | basicCircle | Simple, Circle | circle | 4 | | basicLine | Simple, Line | line | 4 | | basicTriangle | Simple, Traingle | shapes | 4 | | dashedStroke | Dashed, Line | stroke | 10 | | dottedStroke | Dotted, Line | stroke | 10 | | filledCircle | Filled, Circle | circle | 4 | | skeletonLeaf | Veins, Leaf | leaf | 10 | | solidStroke | Solid, Line | stroke | 10 | | * | | | 0 | |
| ElementName | APIquery | Type | UserRating | Click to Add | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicCircle | Simple, Circle | circle | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicLine | Simple, Line | line | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicTriangle | Simple, Traingle | shapes | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dashedStroke | Dashed, Line | stroke | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dottedStroke | Dotted, Line | stroke | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| filledCircle | Filled, Circle | circle | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| skeletonLeaf | Veins, Leaf | leaf | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| solidStroke | Solid, Line | stroke | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| * | | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Hence this was a success.

When reviewing my database design I decided to change the TemplatesTable weighting options to Yes/No fields (hence containing a Boolean data type). This change was due to the fact that templates are either inspired by one culture or not, they do not have a continuous value.

Hence the database table for templates now looked like this:

| TemplatesID | Template Name | South Asian | North Africar | Middle Easte | Click to Add |
|-------------|-------------------|-------------------------------------|-------------------------------------|--------------|-------------------------------------|
| 1 | concentricCircles | <input checked="" type="checkbox"/> | | | <input checked="" type="checkbox"/> |
| 2 | diagonalTriangles | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| * | (New) | | | | |


```

// End
var
  i : integer;
begin
  with form2.ATD_Templates do
  begin
    open;
    for i:=1 to recordcount do
      begin
        recno:=i;
        if fieldvalues['TemplatesID'] = currentTemplate then
          begin
            form2.label4.caption := FieldValues['South Asian Weight'] + FieldValues['North African Wei
  end;
end;

```

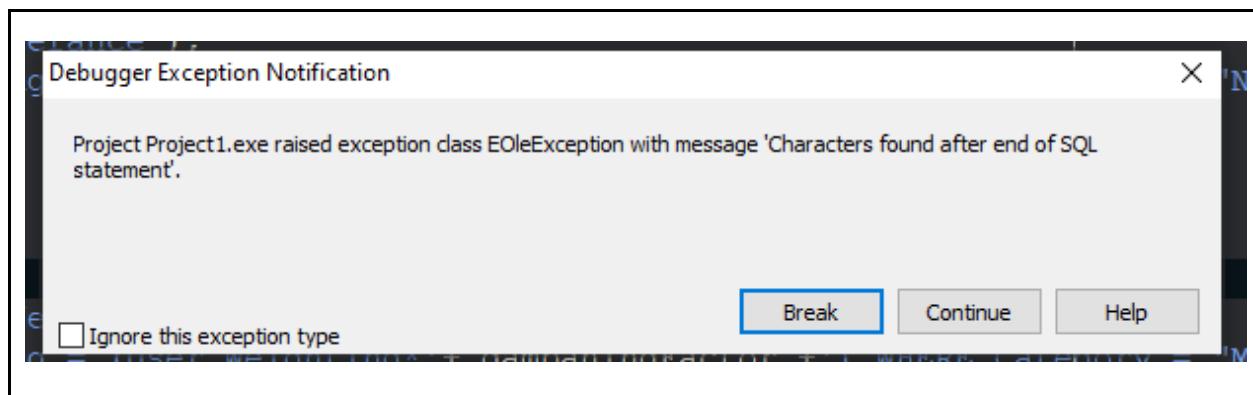
I wrote the above procedure to display the style values of the selected template.

I produced the following test table from the results :

| Template | SA | NA | ME |
|----------|----|----|----|
| 1 | T | F | T |
| 2 | T | T | F |

Hence if we compare it to the database, we can see this is what we expected.

```
uses
  Vax;
  I : integer;
  SA,NA,ME : bool;
begin
  with Form2.ATD_Templates do
    begin
      Open;
      for I:=1 to recordcount do
        begin
          Techno:=I;
          if fieldvalues['TemplateID'] = currentTemplate then
            begin
              //form2.Label14.Caption := FieldValues['South Asian Weight'] + FieldValues['North African Weight'] + FieldValues['Middle Eastern Weight'];
              SA := FieldValues['South Asian Weight'];
              NA := FieldValues['North African Weight'];
              ME := FieldValues['Middle Eastern Weight'];
            end;
        end;
      begin
        with Form2.ADO_QueryTemplates do
          begin
            Active:=False;
            if SA = true then
              begin
                SQL.Clear;
                SQL.Add('UPDATE Style_Preference');
                SQL.Add('SET User_Weighting = (User_Weighting'+ dampaningFactor +'') WHERE Category = "South Asian";');
                ExecSQL ;
              end;
            if NA = true then
              begin
                SQL.Clear;
                SQL.Add('UPDATE Style_Preference');
                SQL.Add('SET User_Weighting = (User_Weighting'+ dampaningFactor +'') WHERE Category = "North African";');
                ExecSQL ;
              end;
            if ME = true then
              begin
                SQL.Clear;
                SQL.Add('UPDATE Style_Preference');
                SQL.Add('SET User_Weighting = (User_Weighting'+ dampaningFactor +'') WHERE Category = "Middle Eastern";');
                ExecSQL ;
              end;
            form2.ATD_Style_preferences.open;
            Form2.ATD_Style_preferences.Refresh;
          end;
      end;
    end;
end;
```



Thus I adapted the code to get the style values for the template and store these in the variables. Then for all the styles which were in the template the dampeningFactor was applied, changing its value. However initially I combined all the SET and WHERE statements together, under a single UPDATE and execute statement. This produced the error documented above. Hence I split the SQL statements up individually, clearing the SQL before adding each one to the query, and then executing each one individually, which worked.

Stage 11 Review

User Feedback

N/A – back-end process

Changes

- Style Weighting changed to Boolean data type in database
- SQL statements split up
- Table opened
- Decided that the element weighting would have no effect on randomisation – however still left in as user can open database and see their element trends

What has been done

The like and dislike buttons have been given functionality.

How has it been tested

- Seeing if database values are as expected

Criteria Being met

- Designs being specialised to a user
- The ability to like or dislike each design

Functionality Checklist

- Does the like button increase weightings of styles?
- Does the like button increase the weighting of style?
- Does the dislike button decrease weightings of styles?
- Does the dislike button decrease the weighting of style?

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

Templates can be saved to your computer.

The like and dislike button change the weighting of styles and elements

Stage 12 - New Design

First I individually wrote the procedures which were still needed to generate a new design :

Resetting the canvas

```
procedure newDesign();  
  Var  
    bitmap : TBitmap;  
    i, index: Integer;  
  begin  
    //Loads hand template into canvas  
    bitmap := TBitmap.Create;  
    try  
      Bitmap.LoadFromFile('handOutlineV5.bmp');  
      form2.hennaDesign_canvas.canvas.Brush.Bitmap := Bitmap;  
      form2.hennaDesign_canvas.canvas.FillRect(Rect(0,0,1000,1000));  
    finally  
      Form1.Canvas.Brush.Bitmap := nil;  
      Bitmap.Free;  
    end;
```

Resting linked list variables

```
elementsLlSp:= -1;  
elementsTop := -1;  
for index := 0 to 30 do  
begin  
  elementsLl[index].pointer:=-1;  
  elementsLl[index].item := '';  
end;  
end;
```

Pick Template

```

procedure pickTemplate();
var
  i, j, cumulativeWeighting, randomNumb, upperLimit, style : integer ;
  styleInfo : array[1..3,0..2] of integer ;
  numbFound, templateFound : bool;
  styleName : string;
begin
  cumulativeWeighting := 0;
  with form_homepage.ATD_Style_preferences do
    begin
      open;
      for i:=1 to 3 do
        begin
          recno:=i;
          styleInfo[i,0] := i;
          styleInfo[i,1] := cumulativeWeighting + 1;
          cumulativeWeighting := cumulativeWeighting + fieldvalues['User_Weighting'];
          styleInfo[i,2] := cumulativeWeighting;
        end;
    end;
  randomNumb := RandomRange(1,cumulativeWeighting) ;
  numbFound := false;
  j := 0;
  while numbFound = false do
    begin
      j := j + 1;
      upperLimit := styleInfo[j,2];
      if (upperLimit >= randomNumb) then
        begin
          style := j;
          numbFound := true;
        end;
    end;
  with form_homepage.ATD_Style_preferences do
    begin
      open;
      recno:=style;
      styleName := fieldvalues['Category'] + ' Weight';
    end;
  templateFound := false;
  with form_homepage.ATD_templates do
    begin
      open;
      while templateFound = false do
        begin
          randomNumb := RandomRange(1, recordcount) ;
          recno:=randomNumb;
          if fieldvalues[styleName] = true then
            begin
              templateFound := true;
            end;
        end;
    end;
  currentTemplate := randomNumb;
  //form_homepage.labell.caption := intToStr(currentTemplate);
end;

```

Drawing template

```

procedure runTemplate();
begin
  case currentTemplate of
    1 : template1(handCentre) ;
    2 : template2(handCentre) ;
  end;
end;

```

When creating the newDesign function, I decided to modify my original design and break this process down into procedures which I had already created in implementation, rather than having to merge them together. These procedures are also smaller and deal with even more specific tasks, improving its long-term maintainability.

I decided to break my code down into the following procedures :

- clearCanvas
- resetVariables
- clearLinkedList
- pickTemplate
- runTemplate
- populateElementLabels
- createButtons

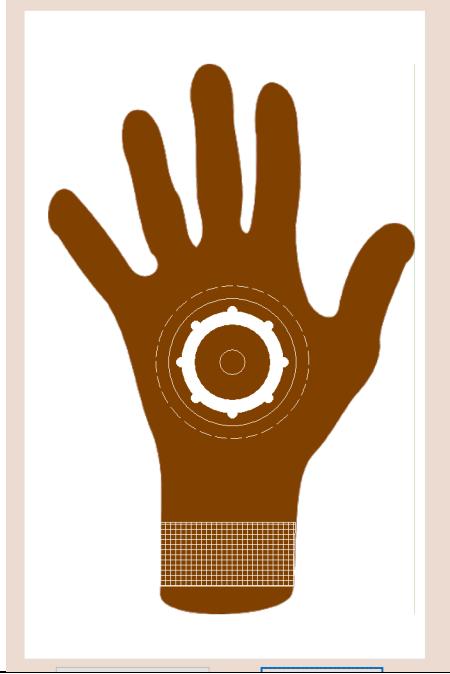
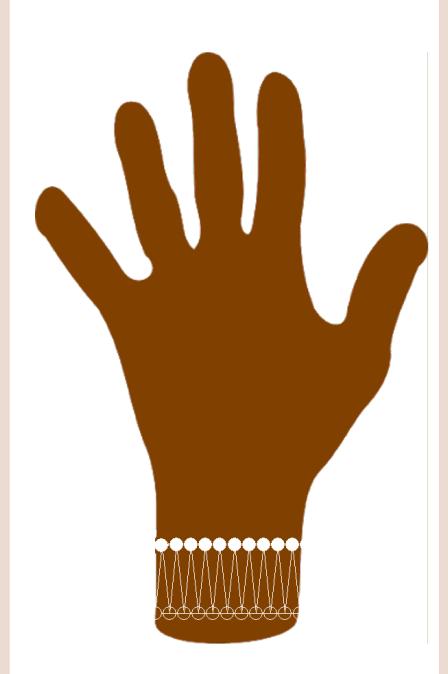
These all existed as isolated procedures in the program. To create a new design, I called them all in the order given above.

```

procedure Tform_homepage.newDesign_btnClick(Sender: TObject);
begin
  clearCanvas();
  resetVariables();
  clearLinkedList(30,elementsLl) ;
  pickTemplate();
  runTemplate();
  populateElementLabels();
  createButtons();
end;

```

Hence to test and ensure this button worked I attempted to generate both templates by clicked the new design button:

| Template | Evidence |
|----------|--|
| 1 |  |
| 2 |  |

As both templates were generated, the new design button passed testing and can be declared as functional.

Stage 12 Review

User Feedback

- N/A – nothing new to show users

Changes

- Decomposed subroutines into even smaller modules, as this fit better with the way my code had been created during implementation

What has been done

The new design button has been given functionality.

How has it been tested

Black box testing has occurred.

- The project is compared to what the output should be when the project is loaded
- As this is the same as what is expected all tests were passed

Criteria Being met

- The program generates henna designs

Functionality Checklist

- Does the new design button produce a new design?

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

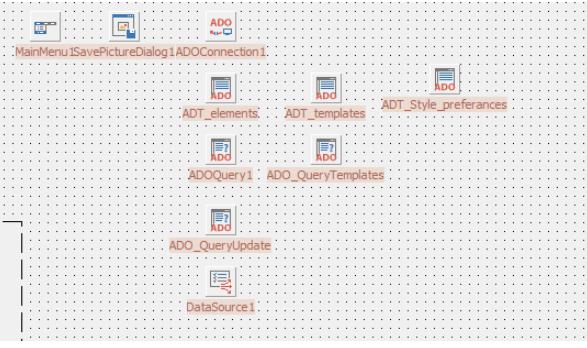
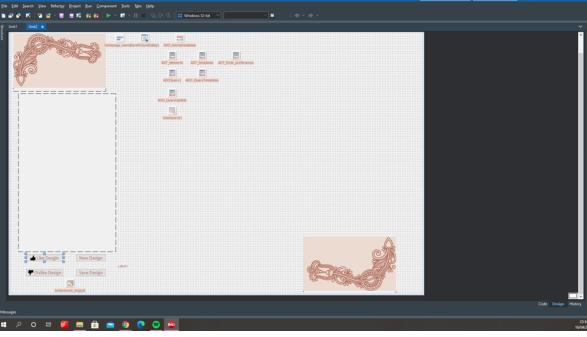
YouTube tutorials can be opened for elements used in the templates.

Templates can be saved to your computer.

The new design button randomly picks and draws a template.

Stage 13 - Tidying up the code

This stage is primarily to improve the maintainability of my code. It is also due to user feedback of the homepage looking messy.

| Action | Evidence |
|---|---|
|  A screenshot of the Delphi IDE's Object Inspector. It shows a tree view of components under 'MainMenuBar1' and 'SavePictureDialog1'. Components include ADOConnection1, ADOTemplates1, ADT_style_preferences, ADOQuery1, ADO_QueryTemplates, ADO_QueryUpdate, and DataSource1. Most components have an 'Add' button next to them. | I grouped all the non-visible components of my application into more logical organisations. Now the main elements are in the top row, and their related successors in layer below them, making the relations between them self-explanatory. |
|  A screenshot of the Delphi IDE showing a simplified component hierarchy. The main window contains a single 'DataSource1' component, which is highlighted with a dashed border. The component palette on the left shows various components like 'ADOConnection1', 'ADOTemplates1', 'ADT_style_preferences', 'ADOQuery1', 'ADO_QueryTemplates', and 'ADO_QueryUpdate', but they are not currently assigned to the form. | Deleting all testing buttons |

Stage 13 Review

User Feedback

- The interface looks a lot tidier and neater

Changes

- Grouped components together

What has been done

The code homepage from has been tidied up.

How has it been tested

- N/A

Criteria Being met

- N/A

Functionality Checklist

- N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

Templates can be saved to your computer.

The new design button randomly picks and draws a template.

Stage 14 – Randomisation and Complex Templates

Finger block

To find an approximation of the coordinates that needed to be filled in to create a pattern on the fingers I upload the henna hand template on the following website:

https://www.mobilefish.com/services/record_mouse_coordinates/record_mouse_coordinates.php

This allowed me to place dots and gain approximations of these coordinates.

The screenshot shows a web-based application for recording mouse coordinates. At the top left is a brown silhouette of a hand with red dots placed on its fingers and thumb. To the right of the hand is a large empty white area. Below the hand is a green sidebar containing the following information:

- Image width = 610 px
- Image height = 997 px
- Number of coordinates = 6

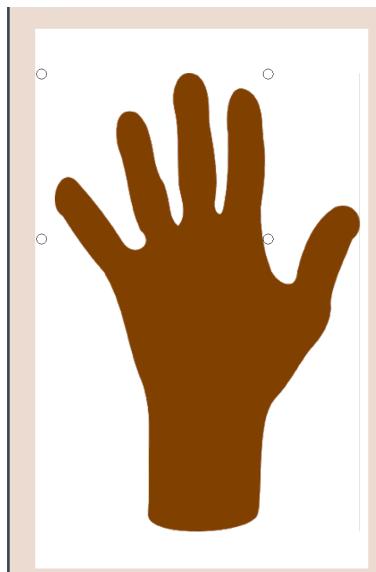
Below this, a scrollable list displays the recorded coordinates:

- 77,385
- 414,352
- 420,165
- 428,67
- 2,75
- 5,379

At the bottom of the sidebar are several buttons:

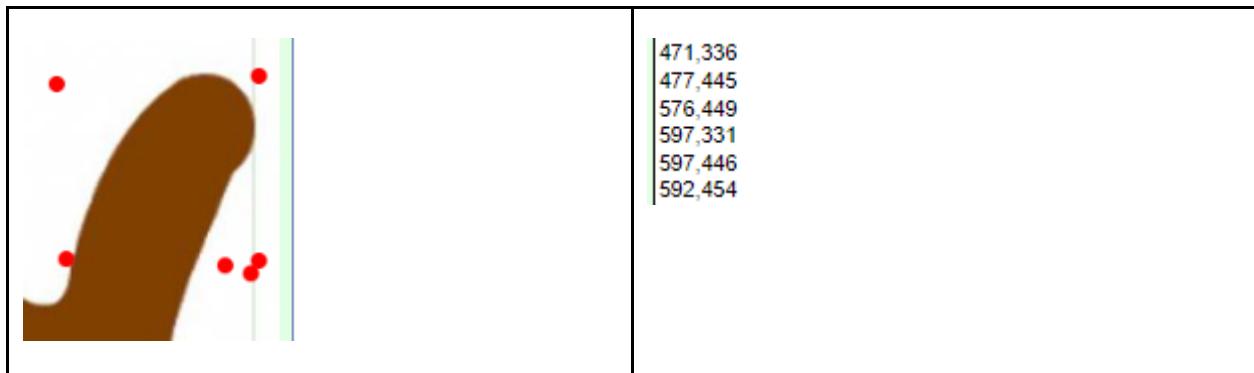
- X-coord: px Y-coord: px
- Add
- Change
- Delete
- Delete all coordinates

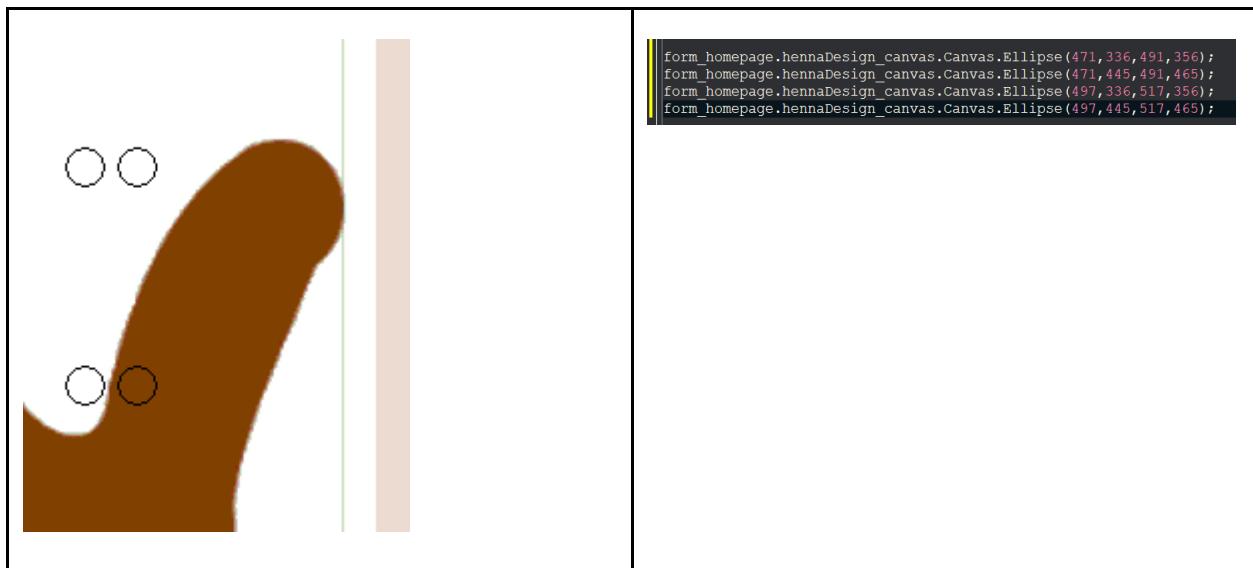
I then repeated the steps for the calculation of the wrist block through Delphi.



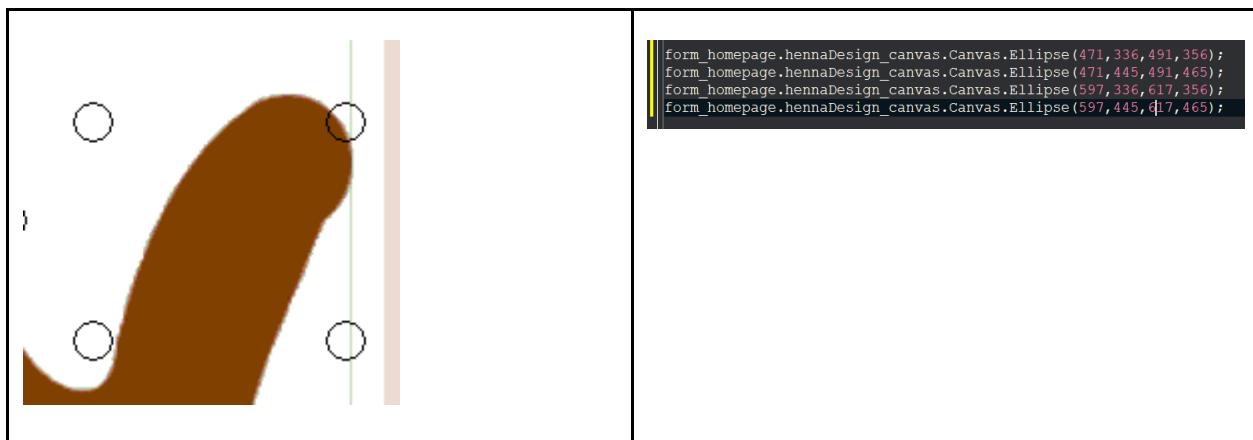
```
form_homepage.hennaDesign_canvas.Canvas.Ellipse(2,75,22,95);  
form_homepage.hennaDesign_canvas.Canvas.Ellipse(2,385,22,405);  
form_homepage.hennaDesign_canvas.Canvas.Ellipse(428,75,448,95);  
form_homepage.hennaDesign_canvas.Canvas.Ellipse(428,385,448,405);
```

I asked my test users if visually these coordinates would encompass the whole of the fingers. After all of them agreed, I proceed to find the thumb coordinates.





The code produced a result which did not even sit on the thumb. Hence when looking at the code I quickly realized the mistake I had made was the x coordinates of the right hand were 100 less than they should be.



Hence with the above error fixed, I set about storing these coordinates in variables, like the wrist block.

```

fingerMaxPoints.topLeft.xCord :=2;
fingerMaxPoints.topLeft.yCord :=75;
fingerMaxPoints.bottomLeft.xCord :=2;
fingerMaxPoints.bottomLeft.yCord :=385;
fingerMaxPoints.topRight.xCord :=428;
fingerMaxPoints.topRight.yCord :=75;
fingerMaxPoints.bottomRight.xCord :=428;
fingerMaxPoints.bottomRight.yCord :=385;

thumbMaxPoints.topLeft.xCord :=471;
thumbMaxPoints.topLeft.yCord :=336;
thumbMaxPoints.bottomLeft.xCord :=471;
thumbMaxPoints.bottomLeft.yCord :=445;
thumbMaxPoints.topRight.xCord :=617;
thumbMaxPoints.topRight.yCord :=356;
thumbMaxPoints.bottomRight.xCord :=617;
thumbMaxPoints.bottomRight.yCord :=465;

```

Stroke Randomisation

```

function chooseStrokeStyle: TBrushStyle;
var
numStyles,p, pInterval : integer;
test : string;
selectedStyle : TBrushStyle;
begin
    p := 1 + Random(8);

    case p of
        1 : selectedStyle := bsSolid;
        2 : selectedStyle := bsHorizontal;
        3 : selectedStyle := bsVertical;
        4 : selectedStyle := bsFDiagonal;
        5 : selectedStyle := bsBDiagonal;
        6 : selectedStyle := bsCross;
        7 : selectedStyle := bsDiagCross;
        8 : selectedStyle := bsClear;
    end ;

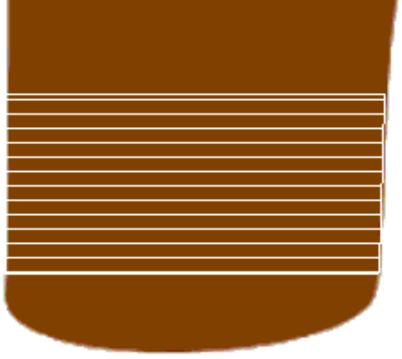
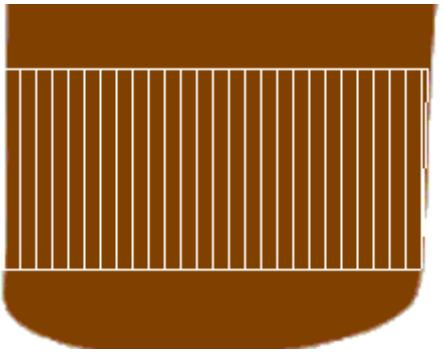
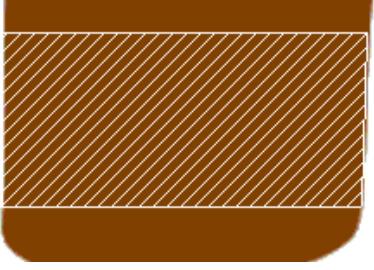
    if selectedStyle = bsSolid then
    begin
        addToLinkedList('solidFill',elementsLl,elementsTop,elementsLlSp);
    end
    else if (selectedStyle = bsCross) or (selectedStyle = bsDiagCross) then
    begin
        addToLinkedList('crisscrossFill',elementsLl,elementsTop,elementsLlSp);
    end
    else if selectedStyle <> bsClear then
    begin
        addToLinkedList('lineFill',elementsLl,elementsTop,elementsLlSp);
    end;
end;
chooseStrokeStyle := selectedStyle;
end;

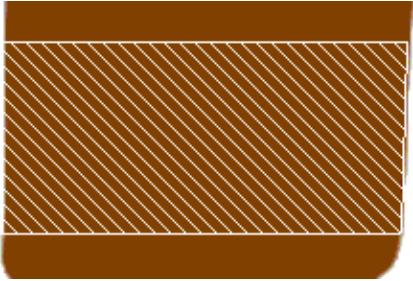
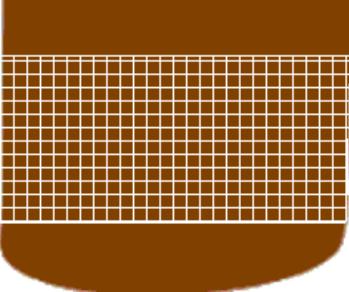
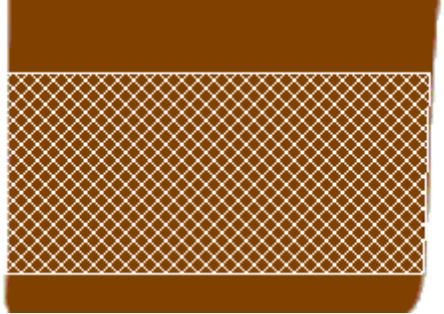
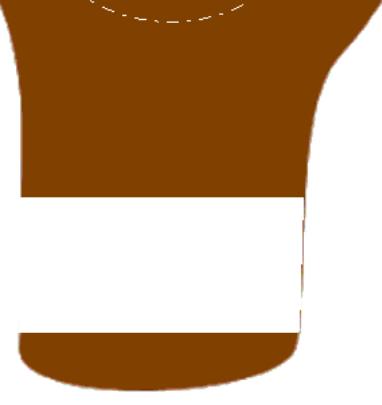
```

I modified the code for randomising a pen stroke so that the brush fill is now randomised.

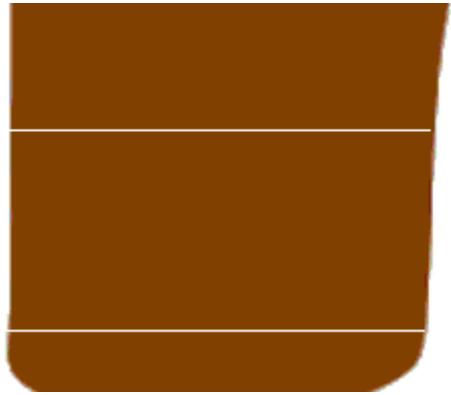
Testing:

I generate new designs till one of each fill was visible.

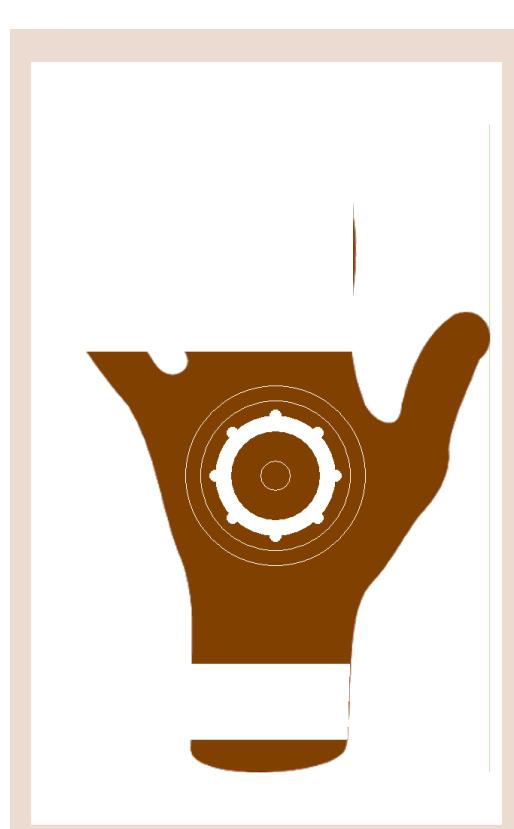
| Style | Evidence |
|--------------|--|
| bsHorizontal |  |
| bsVertical |  |
| bsFDiagonal |  |

| | |
|-------------|--|
| bsBDiagonal |  |
| bsCross |  |
| bsDiagCross |  |
| bsSolid |  |

bsClear



Adding finger to template 1

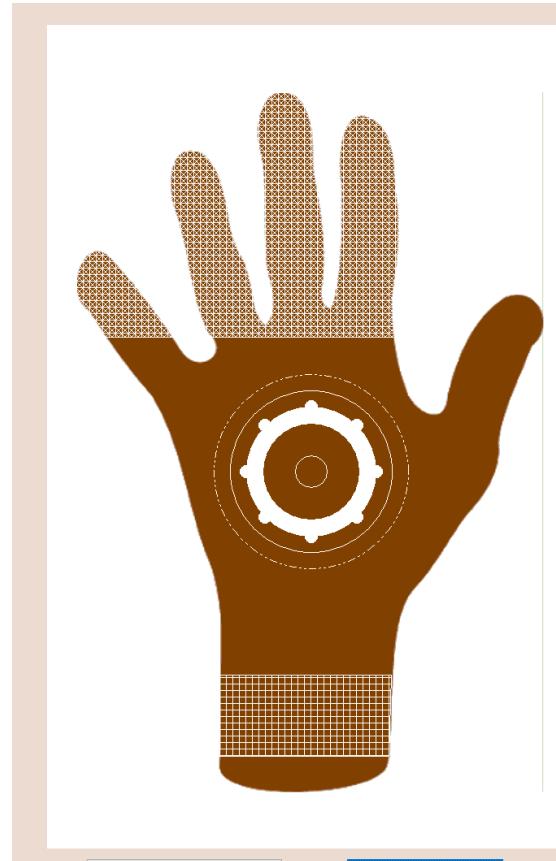


```
pen.Color := clBlack;
rectangle(trunc(fingerMaxPoints.topLeft.xCord),trunc(fingerMaxPoints.topLeft.yCord) ,trunc(fingerMaxPoints.bottomRight.xCord),trunc(fingerMaxPoints.bottomRight.yCord ));
Brush.Style := chooseStrokeStyle;
FloodFill(trunc((fingerMaxPoints.topLeft.xCord)+10),trunc((fingerMaxPoints.topLeft.yCord+10)), clBlack, fsBorder);
pen.Color := defaultColour;
rectangle(trunc(fingerMaxPoints.topLeft.xCord),trunc(fingerMaxPoints.topLeft.yCord) ,trunc(fingerMaxPoints.bottomRight.xCord),trunc(fingerMaxPoints.bottomRight.yCord ));
Brush.Style := defaultBrushStyle;
```

I modified the wrist code to work for the fingers section. However, as the fingers box outline did not necessarily lie on the fingers and create a white border, I first created a black rectangle in the area I wanted to fill. I then flood filled this. I then drew over the black rectangle in a white one, hence “erasing it” from the template. Whilst this worked for the majority of the fills, it gave an odd appearance for the solid fill, as it caused the fingers to disappear.

```
rectangle(trunc(fingerMaxPoints.topLeft.x);
repeat
    Brush.Style := chooseStrokeStyle;
until (Brush.Style <> bsSolid) ;
```

Hence I adapted my code, so it picked a new fill until the fill was one that was not solid. I tested this by observing ten template ones in a row, as none of the fingers has a solid fill I assumed this worked. In user testing the users also pointed out that the fill did not quite cover the last finger on the right, hence I needed to extend the rightmost coordinates of the finger block and missed the wrist slightly, which I incremented by 2px.



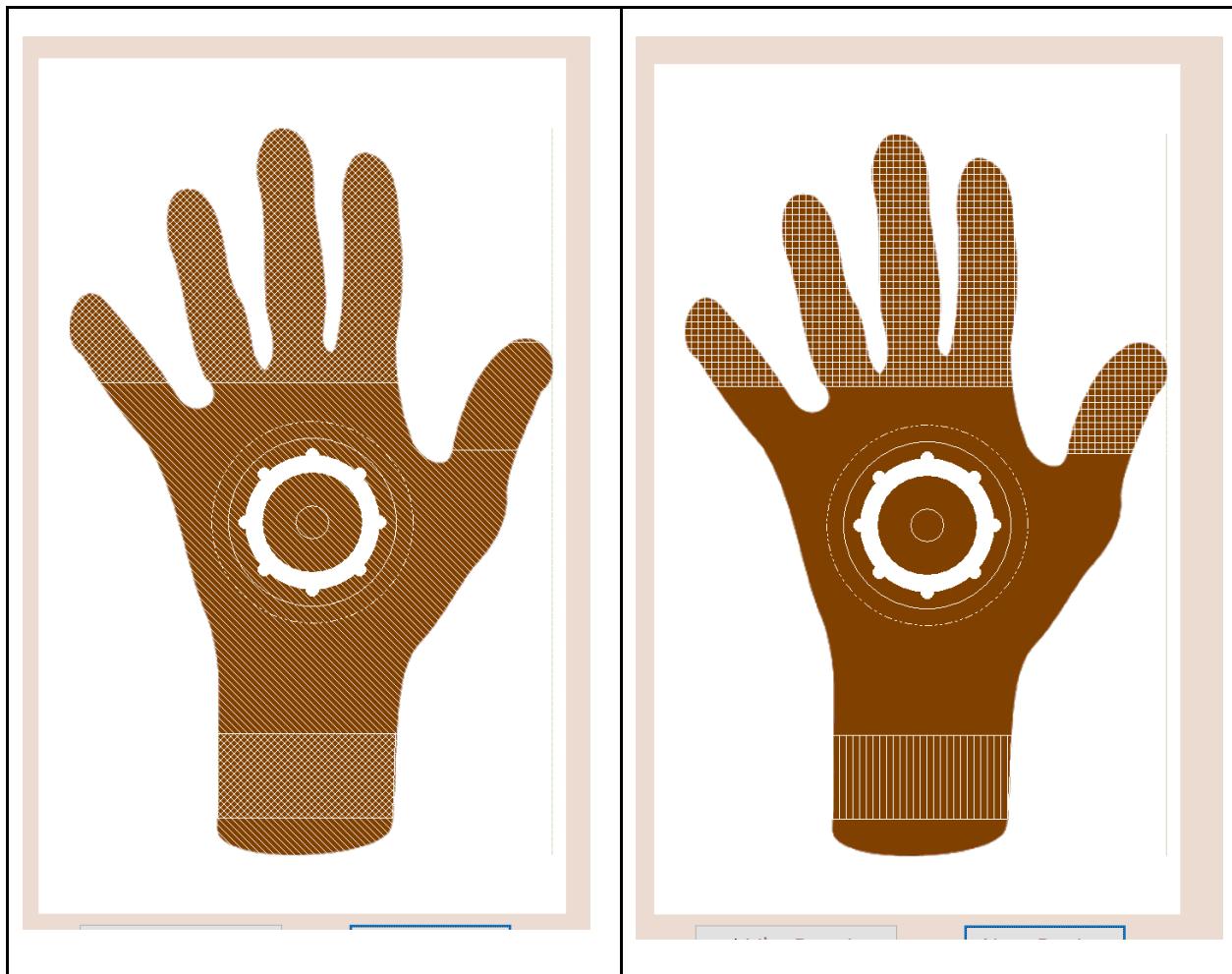
```
fingerMaxPoints.topLeft.xCord :=2;
fingerMaxPoints.topLeft.yCord :=75;
fingerMaxPoints.bottomLeft.xCord :=2;
fingerMaxPoints.bottomLeft.yCord :=385;
fingerMaxPoints.topRight.xCord :=438;
fingerMaxPoints.topRight.yCord :=75;
fingerMaxPoints.bottomRight.xCord :=438;
fingerMaxPoints.bottomRight.yCord :=385;
```

The user was now satisfied with this.

I then repeated the process but for the thumb coordinates. However as the brush style needed to match the fingers, I did not randomize this.

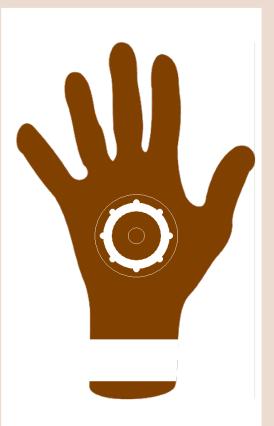
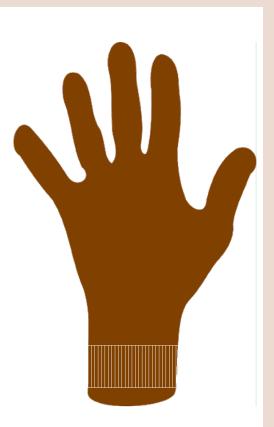
This gave the following result, hence I had to check the logic of my code.

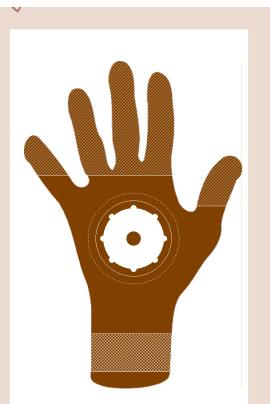
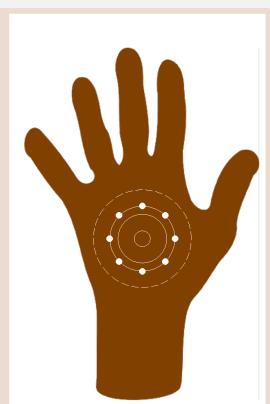
I forgot to add the line which changed the pen colour to `clBlack` before drawing the rectangle. When this was added, the design was drawn successfully.



I then increase the templates randomisation by adding if draw statements around blocks of elements. This massively increases the number of possible designs.

To check there were no major errors with the design, I randomly drew 5 template 1 designs and asked my users to evaluate if they were sensible designs.

| Template | Tanvi | Neha | Sarah |
|---|-------|--|-------|
|  | Yes | Yes | Yes |
|  | Yes | Yes | Yes |
|  | Yes | No <p>As an experienced henna artist, this design seems incredibly simplistic.</p> <p>However as my application is designed for beginners, this sort of design is a good starting point to learn the techniques, thus this is not an issue.</p> | Yes |

| Template | Tanvi | Neha | Sarah |
|--|-------|------|-------|
|  | Yes | Yes | Yes |
|  | Yes | Yes | Yes |

Hence I concluded technically and aesthetically template one was a success.

Changing filled circle

```

procedure filledCircle(topLeftX,topLeftY,bottomRightX,bottomRightY:integer);
begin
with form_homepage.hennaDesign_canvas.Canvas do
begin
form_homepage.hennaDesign_canvas.Canvas.Brush.Style := chooseStrokeStyle;
brush.color := defaultColour; //changes brush colour to black so whole shape drawn is in this colour
basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
brush.Color := clWhite; //changes brush colour back to deafult
form_homepage.hennaDesign_canvas.Canvas.Brush.Style := defaultBrushStyle;
addToLinkedList('filledCircle',elementsLl,elementsTop,elementsLlSp);
end;
end;

```

I changed the fill circle so that the stroke was randomised, this further increases the combinations and is especially useful for those who are learning and practicing new techniques.

Adding Template 3

To add my abstract design template to my program I broke it down into the following sections :

- Code template
- Make changes to program
- Make changes to the database
- Test

Coding:

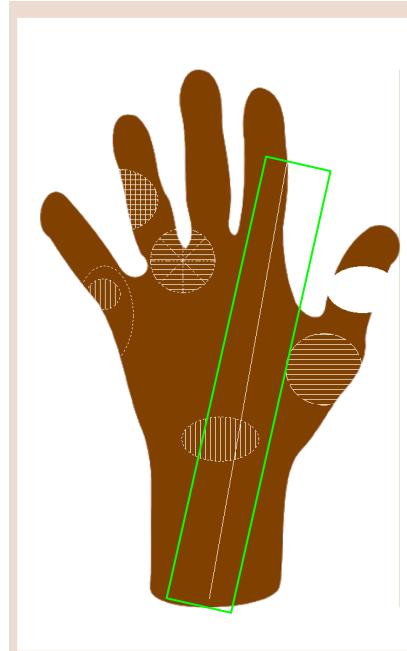
```
procedure template3();
var
  i: integer; tempNum : integer;
  middleCord : TCoordinates;
begin
  for i := 0 to 20 do
  begin
    if toDraw() = true then
    begin
      form_homepage.hennaDesign_canvas.canvas.pen.style := choosePenStyle(i);
      form_homepage.hennaDesign_canvas.canvas.brush.Style := chooseBrushStyle(i);
      middleCord.xCord := Random(form_homepage.hennaDesign_canvas.Width-200)+100;
      middleCord.yCord := Random(form_homepage.hennaDesign_canvas.height-200)+100;
      element := random();
      tempNum := random(100);
      case element of
        1 : basicCircle(Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height));
        2 : basicCircle(trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
        3 : filledCircle(trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
        4 : dashedCircle (trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
        5 : basicTriangle(Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width));
        6 : innerReflectionCircle (trunc(middleCord.xCord-tempNum),trunc(middleCord.yCord-tempNum),trunc(middleCord.xCord+tempNum),trunc(middleCord.yCord+tempNum));
        7 : miniCircleBorder (trunc(middleCord.xCord-tempNum),trunc(middleCord.yCord-tempNum),trunc(middleCord.xCord+tempNum),trunc(middleCord.yCord+tempNum));
      end;
    end;
  end;
end;
```

As individual module tests have been done on the individual elements used to compose this, very little testing was needed at this stage. I added a button whose only function was to call this procedure .

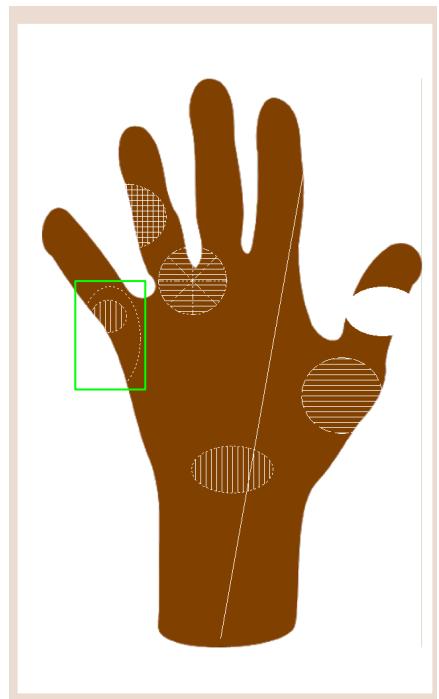
To ensure it works as planned, I called it until every element was drawn at least once.

| Element | Evidence |
|---------|----------|
| | |

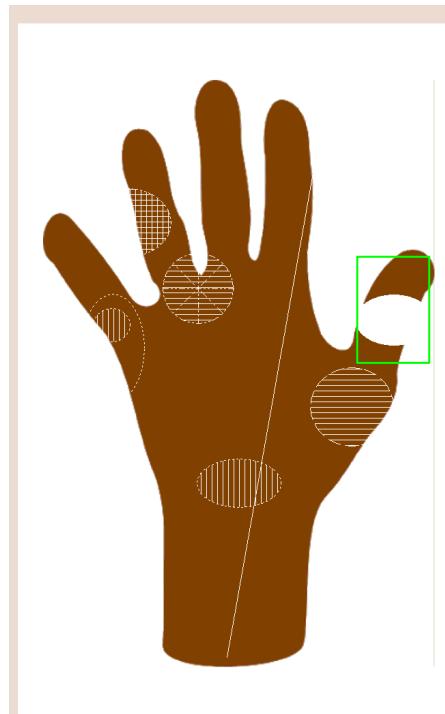
Basic Line



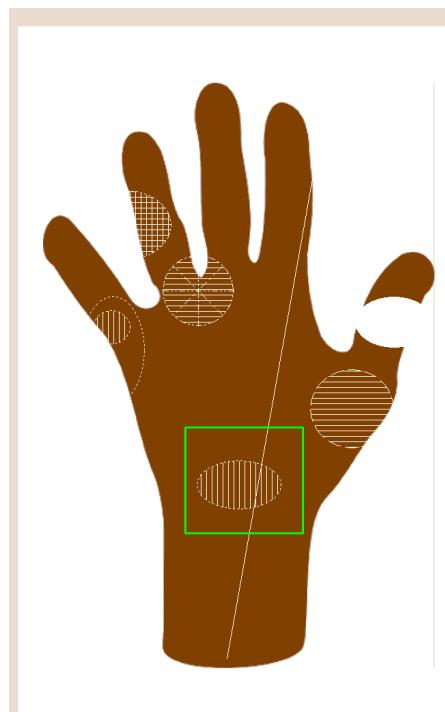
Basic Circle



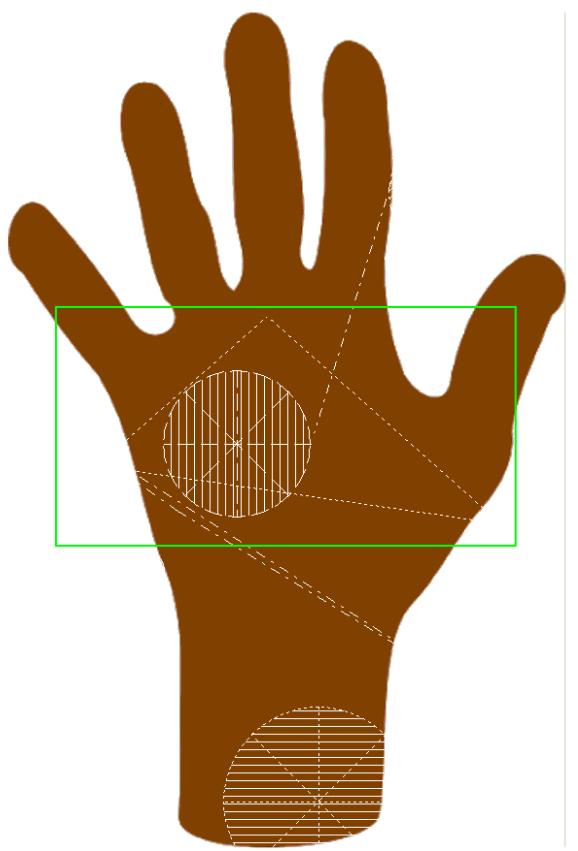
Filled Circle

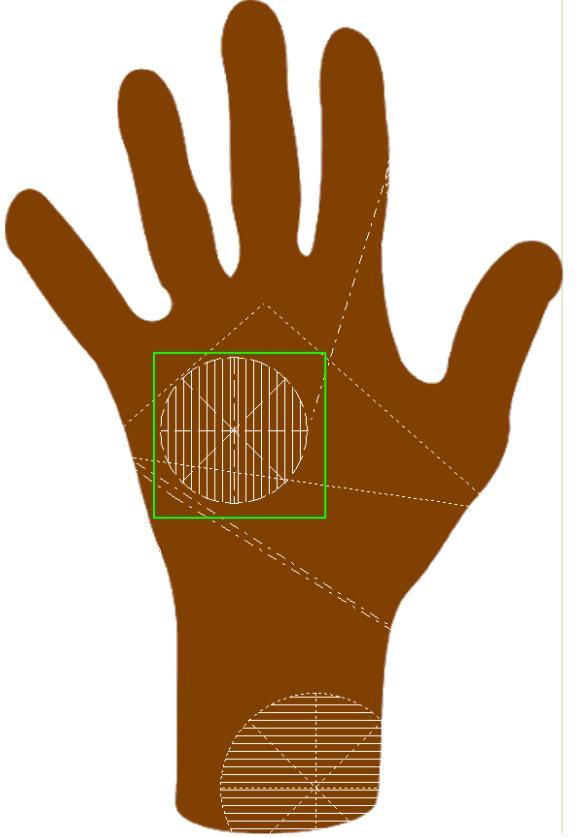


Dashed Circle



Basic Triangle

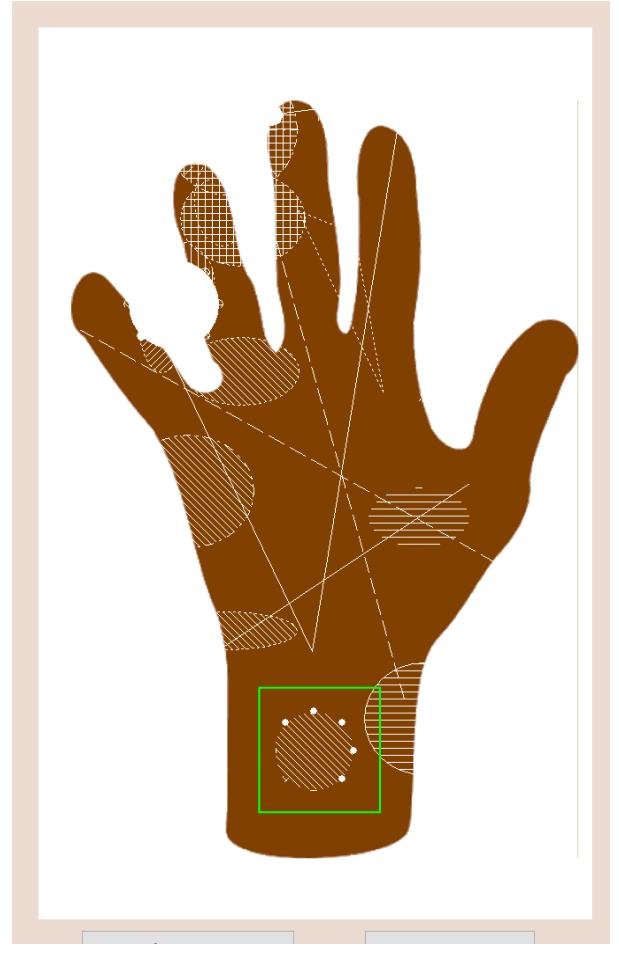


| | |
|-------------------------|--|
| Inner Reflection Circle |  A brown hand silhouette is shown against a white background. Two reflection circles are present: one at the bottom center of the palm containing a grid pattern, and another at the top center of the palm containing a similar grid pattern. A green square highlights the top reflection circle. |
| Mini Border Circle | Did not appear in any of the designs |

Hence when I looked at the code again, I realised that the random function was generating numbers from 0 -6, instead of 1-7. Hence I incremented the random element selection function by one.

```
element := random(7)+1;
```

Mini border Circle



Hence this fixed my issue.

Make changes to the program

I then needed to edit the run template function, so that template 3 was selected , the appropriate procedure was called.

```

procedure runTemplate();
begin
  case currentTemplate of
    1 : template1(handCentre) ;
    2 : template2(handCentre) ;
    3 : template3() ;
  end;
end;

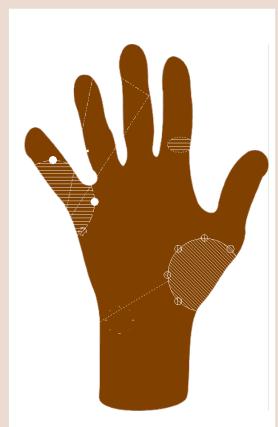
```

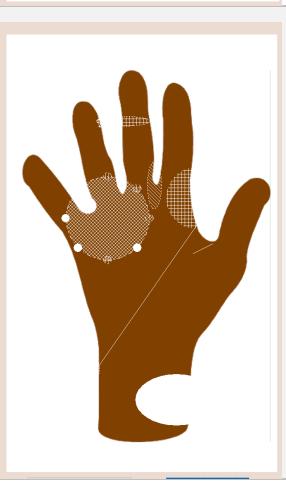
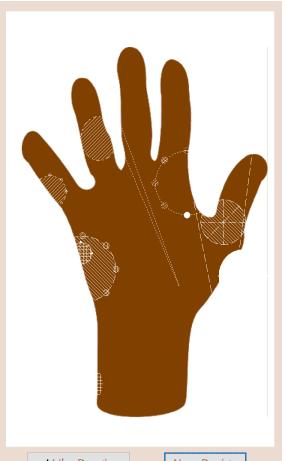
Make changes to the database

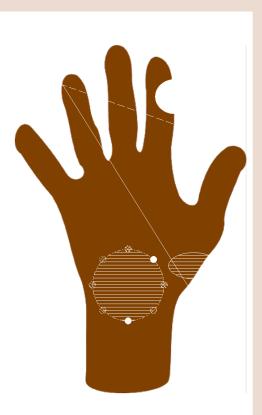
| TemplatesID | Template Name | South Asian | North African | Middle Eastern | Click to Add |
|-------------|-------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------|
| 1 | concentricCircles | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | |
| 2 | diagonalTriangles | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| 3 | abstract | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

I then added the abstract template to the database.

Testing

| Template | Tanvi | Neha | Sarah |
|---|-------|------|-------|
|  | Yes | Yes | Yes |

| | | | |
|---|-----|-----|-----|
|  | Yes | Yes | Yes |
| | Yes | Yes | Yes |
|  | Yes | Yes | Yes |

| | | | |
|---|-----|-----|-----|
|  | Yes | Yes | Yes |
|---|-----|-----|-----|

Stage 14 Review

User Feedback

- Tanvi – I love the abstract design, it's very minimalist and goes well with today's teenage fashion trends
- At this point I asked the users if there were any other features they would like to see in the code
 - Tanvi – A link on the basics of how to apply henna
 - As this is a plausible feature to add I will include it as an option in the menu bar
 - Neha – Nope ! I think the app could be a big hit with beginners even though I was sceptical at first
 - Sarah – Even though it is fairly intuitive to use, an application guide would be nice on how to use it (this I intended on adding in the next stage anyways)

Changes

- A link to instructions on applying henna should be included in the application
-

What has been done

- Randomisation feature have been added to template 1 and 2
 - Template 3 has been created
-

How has it been tested

- Module tests
 - User testing
-

Criteria Being met

- Designs being randomised each time
-

Functionality Checklist

- N/A
-

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

Templates can be saved to your computer.

The new design button randomly picks and draws a template.

All templates work and there are many randomisation features.

Stage 15 – Adding More menu Bar Options

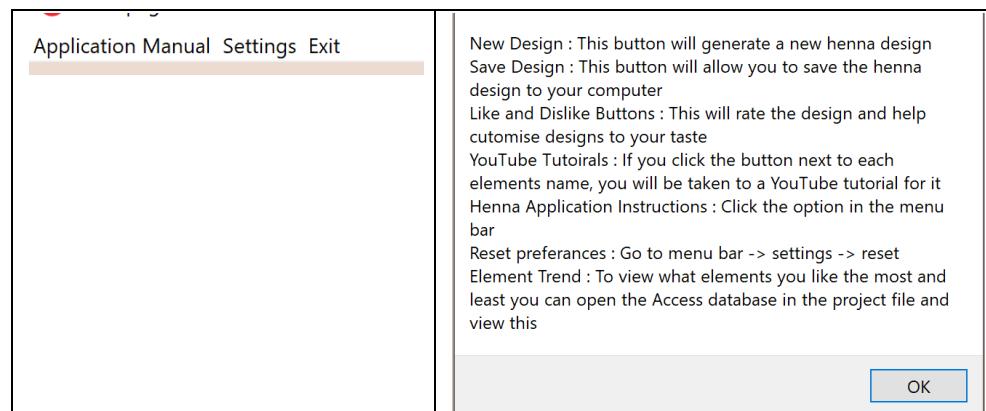
Application Manual

Coding

To help users navigate the application I added an application guide in the menu bar. This informs users about the full functionality of the application and makes it easier to use.

```
procedure Tform_homepage.ApplicationManualClick(Sender: TObject);
begin
showmessage('New Design : This button will generate a new henna design' +#13#10+
'Save Design : This button will allow you to save the henna design to your computer' +#13#10+
'Like and Dislike Buttons : This will rate the design and help customise designs to your taste' +#13#10+
'YouTube Tutorials : If you click the button next to each elements name, you will be taken to a YouTube tutorial for it' +#13#10+
'Henna Application Instructions : Click the option in the menu bar' +#13#10+
'Reset preferences : Go to menu bar -> settings -> reset' +#13#10+
'Element Trend : To view what elements you like the most and least you can open the Access database in the project file and view this');
end;
```

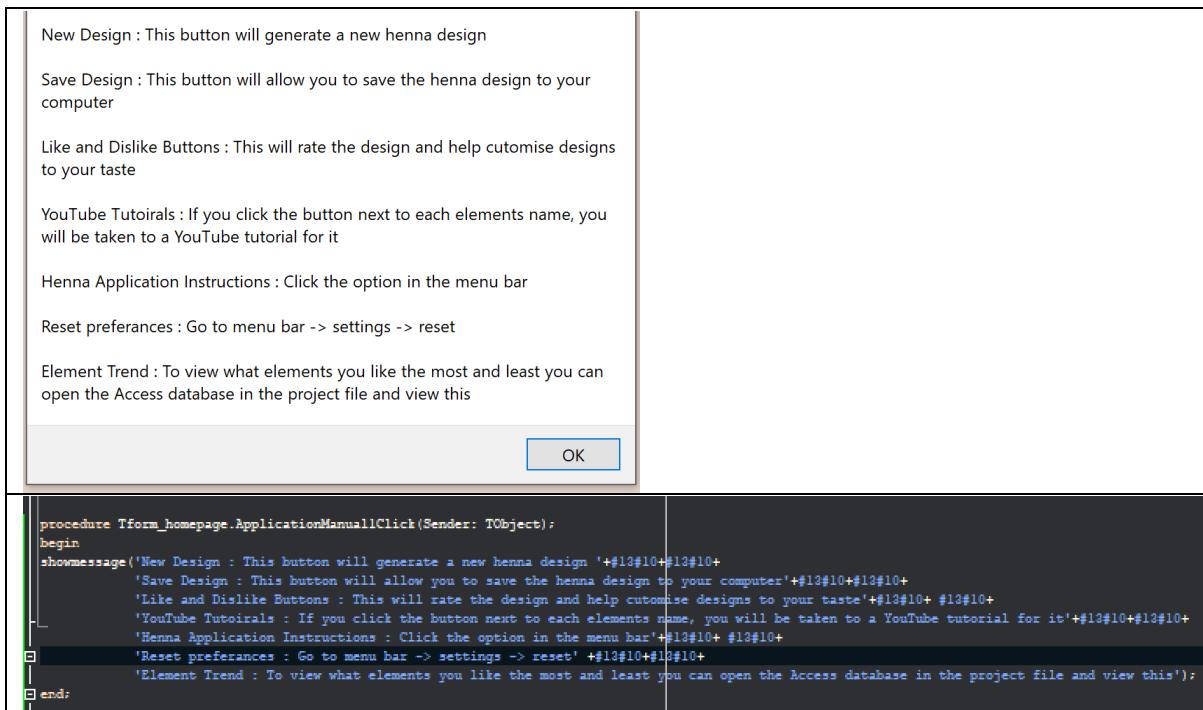
I laid the code out as above to help visualise how this would look in the message box.



User Feedback

I showed this function to Sarah, who seemed particularly keen for this feature to be included in the application. She said that whilst the information was useful, it visually appeared cramped and strained her eyes.

To ensure my application was friendly to those with visual impairments, I added an extra line break between each line.



When Sarah was shown this she made the following comments:

- This is a lot easier to read
- The instructions are short but really helpful

Henna Application Instructions

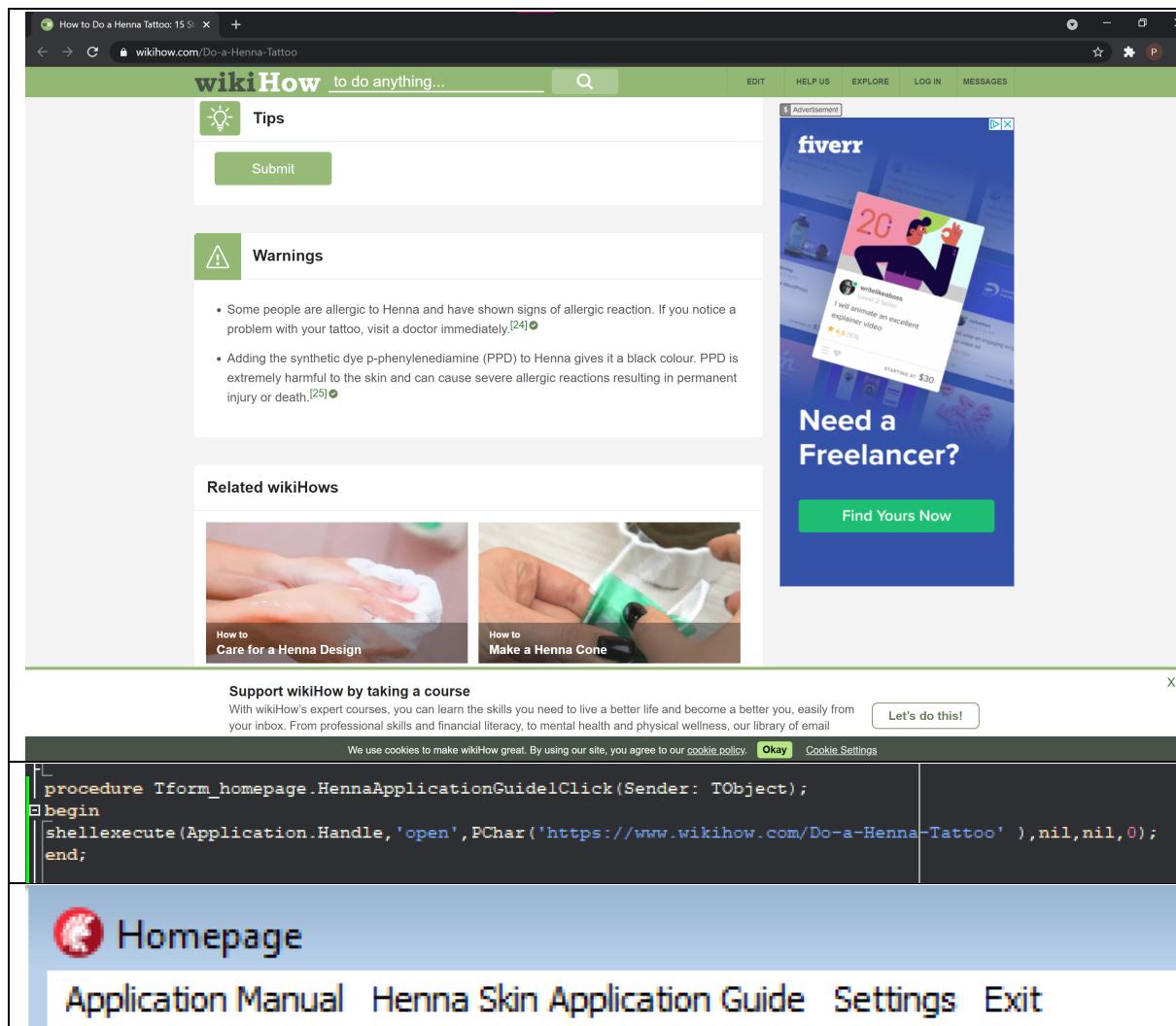
Choosing a webpage

I consulted Neha on which webpages she believed were good at giving instructions on how to apply henna on the skin. She recommended the following link : <https://www.wikihow.com/Do-a-Henna-Tattoo>.

When I looked into the webpage it had the following features:

- Instructions on making the paste and application
- Community Q&A
- A list of needed items
- Tips
- Warning
- Related articles

All of these features maybe useful for beginner henna artists and provide useful information. The instructions were also written in a simple and easily understandable way . Hence I decided to use this link.



I once again used the shellAPI library in order to open a webpage with the specified URL. As this took me to the appropriate webpage when clicked it is fully functional.

User Feedback

I showed this function to Tanvi, who suggested it should be included in the application. She had the following feedback:

- The webpage is really well explained and makes the fundamentals of henna clear

Stage 15 Review

User Feedback

At this point the users only had positive feedback about the application and did not suggest any changes.

Changes

N/A

What has been done

- An application manual has been included
- Instructions on how to apply henna on the skin have been included

How has it been tested

- Module tests
- User testing

Criteria Being met

- A clear instruction menu on how to use the application

Functionality Checklist

- N/A

Project Summary

The loading screen is fully functional.

The homepage user interface has been set up.

All elements can now be drawn.

Two templates can now be drawn.

The linked list is functional.

The henna database now exists.

The elements used in the templates are now displayed.

The Access database has been linked to the project.

YouTube tutorials can be opened for elements used in the templates.

Templates can be saved to your computer.

The new design button randomly picks and draws a template.

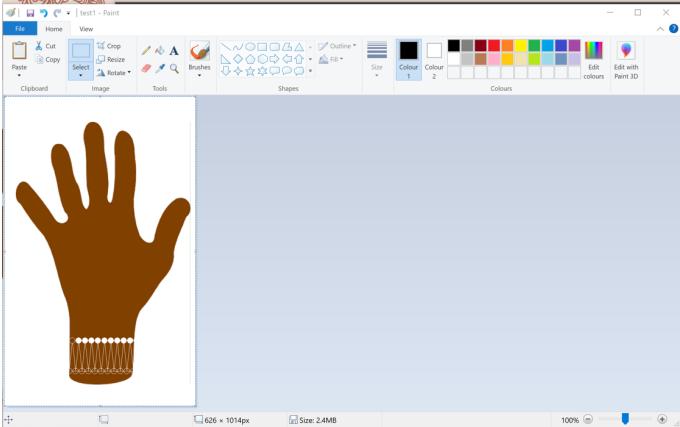
All templates work and there are many randomisation features.

An application guide has been included.

Basic henna application (on the skin) instructions have been included.

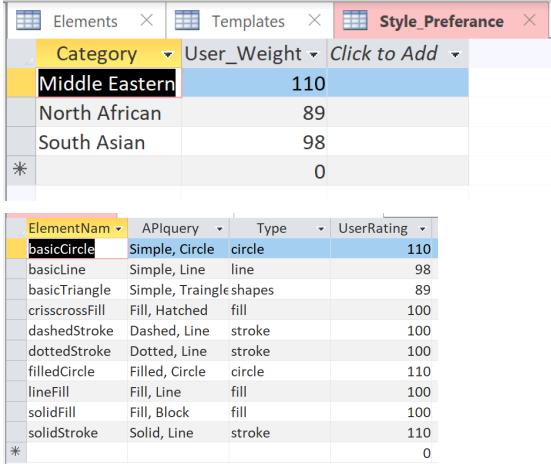
Stage 16 – Final Testing

Henna Design Generation

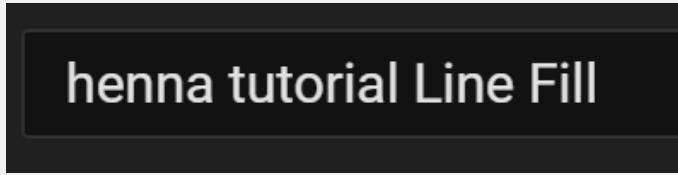
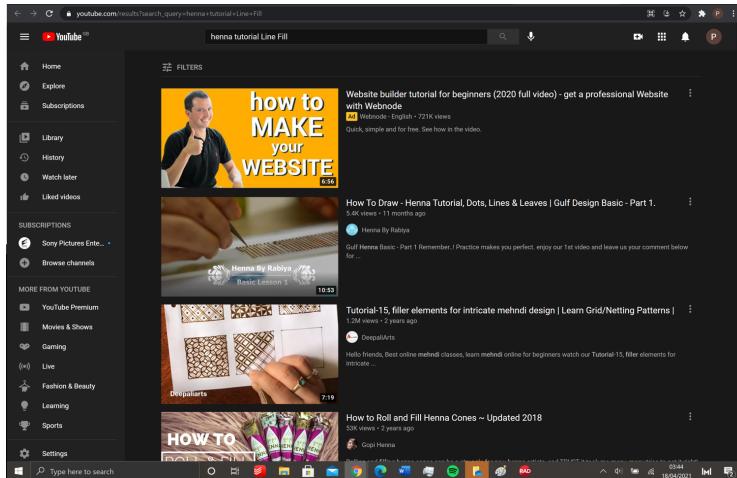
| To be tested | Evidence |
|---|--|
| Loading hand onto the canvas |  |
| | Homepage code – Line 215, Line 932 |
| Henna elements | Module tests were successful |
| Element's variation | Module tests were successful |
| Templates working | Module tests were successful |
| Templates acceptable as henna design | Positive stakeholder feedback |
| Saving templates |  |
| | Homepage code – Line 875 |

Design Recommendations

| To be tested | Evidence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|--|--------------|------------|------|------------|-------------|----------------|--------|-----|-----------|--------------|------|-----|---------------|------------------|--------|-----|----------------|---------------|------|-----|--------------|--------------|--------|-----|--------------|--------------|--------|-----|--------------|----------------|--------|-----|----------|------------|------|-----|-----------|-------------|------|-----|-------------|-------------|--------|-----|---|--|--|---|----------|-------------|--------------|----------------|-----|--|---------------|-----|--|-------------|-----|--|---|---|--|
| Resting database | <p>Settings Exit</p> <h2>Reset Preferences</h2> <pre> procedure Tform_homepage.ResetPreferences1Click(Sender: TObject); var i : integer ; begin with form_homepage.adoquery1 do begin SQL.Clear; SQL.Add('UPDATE Elements'); SQL.Add('SET UserRating = 100'); ExecSQL ; end; with form_homepage.adoquery1 do begin SQL.Clear; SQL.Add('UPDATE Style_Preference'); SQL.Add('SET User_Weighting = 100'); ExecSQL ; end; end; </pre> <p>Elements</p> <table border="1"> <thead> <tr> <th>ElementName</th> <th>APIquery</th> <th>Type</th> <th>UserRating</th> </tr> </thead> <tbody> <tr> <td>basicCircle</td> <td>Simple, Circle</td> <td>circle</td> <td>100</td> </tr> <tr> <td>basicLine</td> <td>Simple, Line</td> <td>line</td> <td>100</td> </tr> <tr> <td>basicTriangle</td> <td>Simple, Triangle</td> <td>shapes</td> <td>100</td> </tr> <tr> <td>crisscrossFill</td> <td>Fill, Hatched</td> <td>fill</td> <td>100</td> </tr> <tr> <td>dashedStroke</td> <td>Dashed, Line</td> <td>stroke</td> <td>100</td> </tr> <tr> <td>dottedStroke</td> <td>Dotted, Line</td> <td>stroke</td> <td>100</td> </tr> <tr> <td>filledCircle</td> <td>Filled, Circle</td> <td>circle</td> <td>100</td> </tr> <tr> <td>lineFill</td> <td>Fill, Line</td> <td>fill</td> <td>100</td> </tr> <tr> <td>solidFill</td> <td>Fill, Block</td> <td>fill</td> <td>100</td> </tr> <tr> <td>solidStroke</td> <td>Solid, Line</td> <td>stroke</td> <td>100</td> </tr> <tr> <td>*</td> <td></td> <td></td> <td>0</td> </tr> </tbody> </table> <p>Style_Preference</p> <table border="1"> <thead> <tr> <th>Category</th> <th>User_Weight</th> <th>Click to Add</th> </tr> </thead> <tbody> <tr> <td>Middle Eastern</td> <td>100</td> <td></td> </tr> <tr> <td>North African</td> <td>100</td> <td></td> </tr> <tr> <td>South Asian</td> <td>100</td> <td></td> </tr> <tr> <td>*</td> <td>0</td> <td></td> </tr> </tbody> </table> | ElementName | APIquery | Type | UserRating | basicCircle | Simple, Circle | circle | 100 | basicLine | Simple, Line | line | 100 | basicTriangle | Simple, Triangle | shapes | 100 | crisscrossFill | Fill, Hatched | fill | 100 | dashedStroke | Dashed, Line | stroke | 100 | dottedStroke | Dotted, Line | stroke | 100 | filledCircle | Filled, Circle | circle | 100 | lineFill | Fill, Line | fill | 100 | solidFill | Fill, Block | fill | 100 | solidStroke | Solid, Line | stroke | 100 | * | | | 0 | Category | User_Weight | Click to Add | Middle Eastern | 100 | | North African | 100 | | South Asian | 100 | | * | 0 | |
| ElementName | APIquery | Type | UserRating | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicCircle | Simple, Circle | circle | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicLine | Simple, Line | line | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicTriangle | Simple, Triangle | shapes | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| crisscrossFill | Fill, Hatched | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dashedStroke | Dashed, Line | stroke | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dottedStroke | Dotted, Line | stroke | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| filledCircle | Filled, Circle | circle | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| lineFill | Fill, Line | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| solidFill | Fill, Block | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| solidStroke | Solid, Line | stroke | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| * | | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Category | User_Weight | Click to Add | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Middle Eastern | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| North African | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| South Asian | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| * | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| To be tested | Evidence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|--|-------------|------------|------|------------|-------------|----------------|--------|-----|-----------|--------------|------|----|---------------|-------------------|--------|----|----------------|---------------|------|-----|--------------|--------------|--------|-----|--------------|--------------|--------|-----|--------------|----------------|--------|-----|----------|------------|------|-----|-----------|-------------|------|-----|-------------|-------------|--------|-----|
| Likes and disliked |  <p>The screenshot shows two tables. The top table has columns for Category, User_Weight, and Click to Add. It contains rows for Middle Eastern (110), North African (89), South Asian (98), and an empty row with a value of 0. The bottom table has columns for ElementName, APIquery, Type, and UserRating. It lists various shape types with their corresponding API queries, types, and user ratings.</p> <table border="1"> <thead> <tr> <th>ElementName</th> <th>APIquery</th> <th>Type</th> <th>UserRating</th> </tr> </thead> <tbody> <tr> <td>basicCircle</td> <td>Simple, Circle</td> <td>circle</td> <td>110</td> </tr> <tr> <td>basicLine</td> <td>Simple, Line</td> <td>line</td> <td>98</td> </tr> <tr> <td>basicTriangle</td> <td>Simple, Triangles</td> <td>shapes</td> <td>89</td> </tr> <tr> <td>crisscrossFill</td> <td>Fill, Hatched</td> <td>fill</td> <td>100</td> </tr> <tr> <td>dashedStroke</td> <td>Dashed, Line</td> <td>stroke</td> <td>100</td> </tr> <tr> <td>dottedStroke</td> <td>Dotted, Line</td> <td>stroke</td> <td>100</td> </tr> <tr> <td>filledCircle</td> <td>Filled, Circle</td> <td>circle</td> <td>110</td> </tr> <tr> <td>lineFill</td> <td>Fill, Line</td> <td>fill</td> <td>100</td> </tr> <tr> <td>solidFill</td> <td>Fill, Block</td> <td>fill</td> <td>100</td> </tr> <tr> <td>solidStroke</td> <td>Solid, Line</td> <td>stroke</td> <td>110</td> </tr> </tbody> </table> <p>Expect some values to be altered slightly as like and dislike clicked a few times Homepage code – Line 386, Line 393</p> | ElementName | APIquery | Type | UserRating | basicCircle | Simple, Circle | circle | 110 | basicLine | Simple, Line | line | 98 | basicTriangle | Simple, Triangles | shapes | 89 | crisscrossFill | Fill, Hatched | fill | 100 | dashedStroke | Dashed, Line | stroke | 100 | dottedStroke | Dotted, Line | stroke | 100 | filledCircle | Filled, Circle | circle | 110 | lineFill | Fill, Line | fill | 100 | solidFill | Fill, Block | fill | 100 | solidStroke | Solid, Line | stroke | 110 |
| ElementName | APIquery | Type | UserRating | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicCircle | Simple, Circle | circle | 110 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicLine | Simple, Line | line | 98 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| basicTriangle | Simple, Triangles | shapes | 89 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| crisscrossFill | Fill, Hatched | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dashedStroke | Dashed, Line | stroke | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dottedStroke | Dotted, Line | stroke | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| filledCircle | Filled, Circle | circle | 110 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| lineFill | Fill, Line | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| solidFill | Fill, Block | fill | 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| solidStroke | Solid, Line | stroke | 110 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

YouTube Videos

| To be tested | How to collect evidence |
|-------------------------|--|
| API Query Keywords |  <p>Expected “henna tutorial Line Fill”</p> |
| Calling the YouTube API |  <p>Homepage code – Line 400</p> |
| Elements displayed | Module tests successful |

Linked List

| To be tested | How to collect evidence |
|--------------|-------------------------|
|--------------|-------------------------|

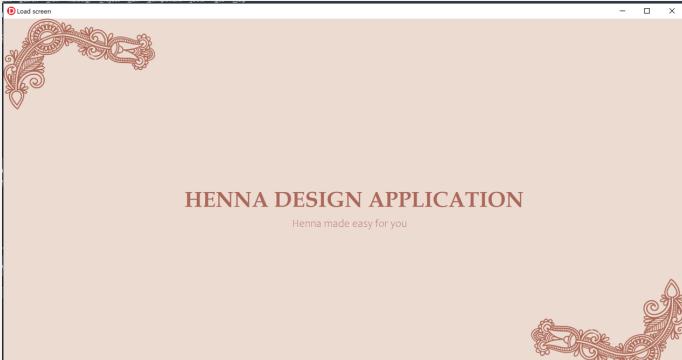
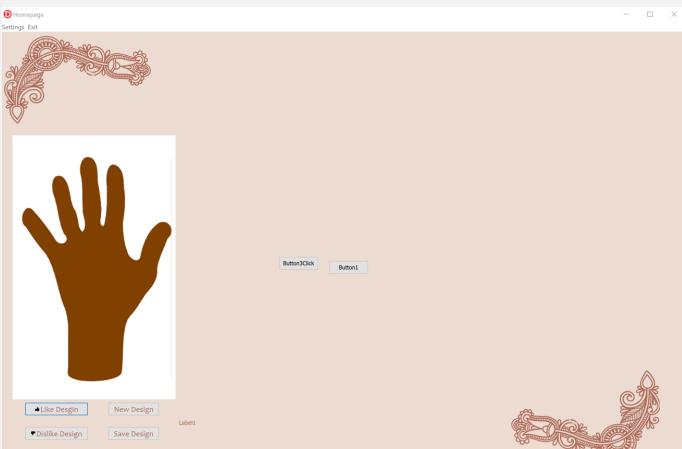
| | |
|---------------------------|-------------------------|
| Items stored in LL | Module tests successful |
| LL traversal | Module tests successful |

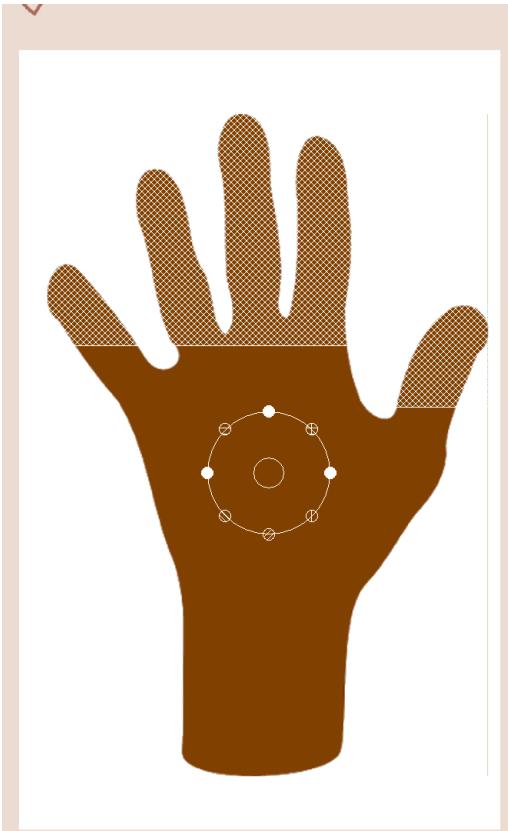
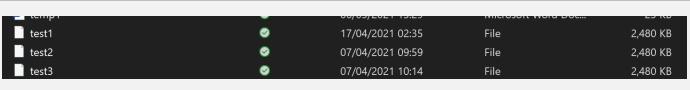
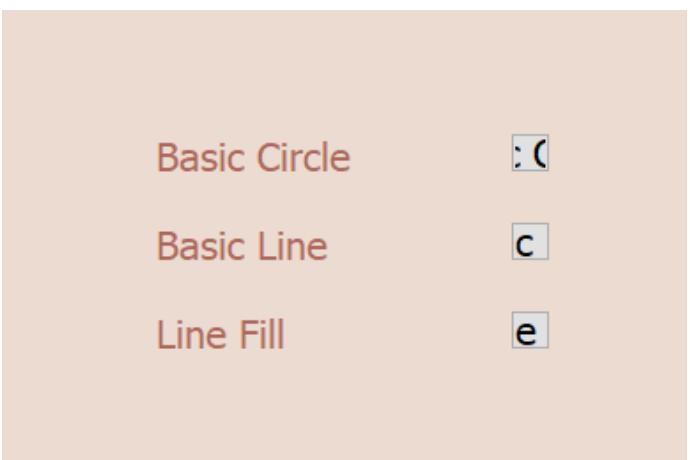
User Interface

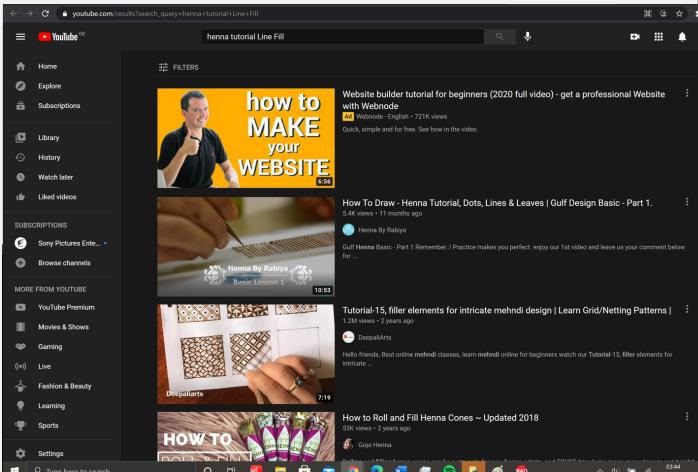
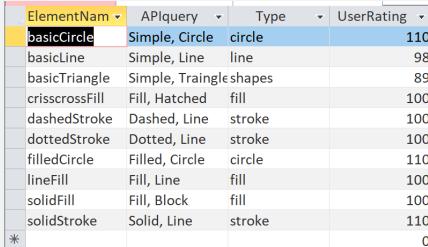
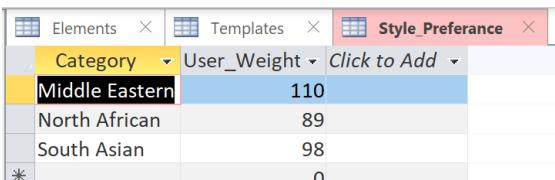
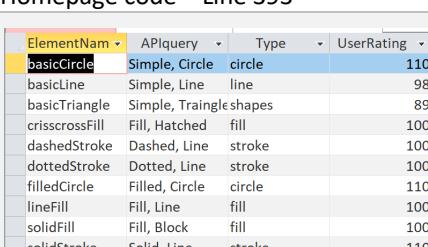
| | |
|---|-------------------------|
| To be tested | How to collect evidence |
| Technically correct | Module tests successful |
| Aesthetical and visually correct | User feedback positive |

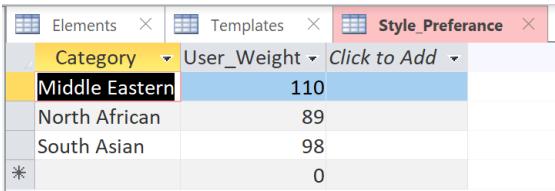
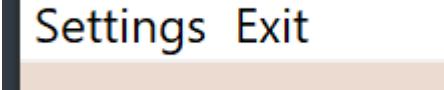
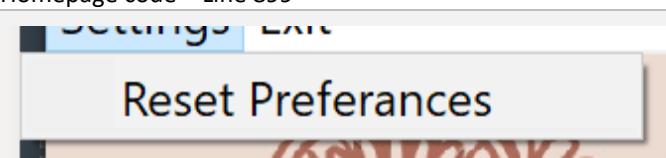
Functionality Checklist

My code should meet the following checklist to ensure it works properly and has full functionality :

| Action to test | Working? Y/N | Evidence |
|---|-----------------|--|
| Does the loading screen automatically appear? | Y |  |
| Does the loading screen automatically disappear after a set time? | Y | Load screen code – Line 36 |
| Does the homepage automatically appear? | Y |  Load screen code – Line 36 |

| Action to test | Working? Y/N | Evidence |
|--|-----------------|---|
| Does the new design button produce a new design? | Y |  <p>Home screen code – Line 998</p> |
| Can you save designs? | Y |  <p>Home screen code – Line 875</p> |
| Are the elements used displayed? | Y |  |

| Action to test | Working? Y/N | Evidence |
|--|-----------------|--|
| Can you go to the relevant YouTube pages for the elements? | Y |  <p>Homepage code – Line 400</p> |
| Are pictures of the isolated elements displayed? | N | |
| Does the like button increase weightings of styles? | Y |  <p>Homepage code – Line 393</p> |
| Does the like button increase the weighting of style? | Y |  <p>Homepage code – Line 393</p> |
| Does the dislike button decrease weightings of styles? | Y |  <p>Homepage code – Line 386</p> |

| Action to test | Working? Y/N | Evidence |
|--|-----------------|---|
| Does the dislike button decrease the weighting of style? | Y |  <p>Homepage code – Line 386</p> |
| Does the menu bar have a settings option? | Y |  |
| Does the menu bar have an exit option? | Y |  <p>Homepage code – Line 899</p> |
| Is there an option to reset the application? | Y |  <p>Homepage code – Line 914</p> |

Stage 17 – Final Stakeholder Feedback

Questions

I got the stakeholders to use my app for three henna sessions each. I then asked them the following questions and recorded their responses:

1. How did you find the process of creating a new design?
2. Did you like the design produced?
3. Did you feel the designs got more specialised to you the more you used the application?
4. Were the YouTube tutorials found useful?
5. How did you find the overall user journey?

Responses

1. How did you find creating a new design?

Tanvi :

This process was incredibly simple. I just clicked the new design button, and it generates a new design for me. I like the fact I didn't have to spend time trying to understand the app, like you sometimes have to with artsy applications, such as photoshop.

Neha:

It was fine, I just clicked the new design button until I found a design I liked.

Sarah:

Creating a new design was really easy, I still looked at the application guide to ensure I was taking the right steps and wasn't going to break something though.

2. Did you like the design produced

Tanvi :

Yup I loved the designs produced, especially the abstract ones! They were super cool, minimalistic and with current fashion trends.

Neha:

Honestly, I wasn't the biggest fan on the henna designs. I understand they may be good for beginners but for me I prefer more traditional ones or that of my own creation.

Sarah:

The designs were so so stunning ! I felt it gave me an insight into these cultures in a safe space and allowed to me to appreciate their beauty.

3. Did you feel the designs got more specialised to you the more you used the application?

Tanvi :

Yes definitely. I kept liking the abstract ones and disliking any other design type. Soon enough the majority of designs produced for me were those which were abstract.

Neha:

Yes they did, I only liked the traditional ones, and after liking a bunch of patterns it mostly gave me these.

Sarah:

I just liked all the designs ! So I didn't really end up using this function.

4. Were the YouTube tutorials found useful

Tanvi:

I kept getting stuck on how to draw the designs, so I regularly referred back to the tutorials on each section. I had to scroll a bit, but I always found a suitable tutorial

Neha:

I don't need help on henna designs anymore, so I didn't use this. I'm happy it was an optional feature.

Sarah:

Yes they were really helpful, as I learn best through auditory methods. However I did struggle with getting back to the application, I had to ask my grandson to help me.

5. How did you find the overall user journey?

Tanvi :

As I said before, I'm happy that I needed very little knowledge in order to use this application. Having used things like Photoshop before, I was worried I would have to spend hours trying to understand how to use it. It was really simple to use and an enjoyable experience overall!

Neha:

It was very intuitive to use, I only looked at the menu once to check I wasn't missing any interesting features.

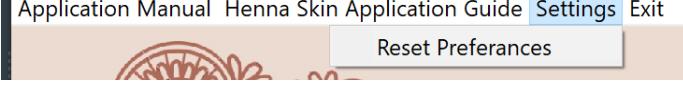
Sarah:

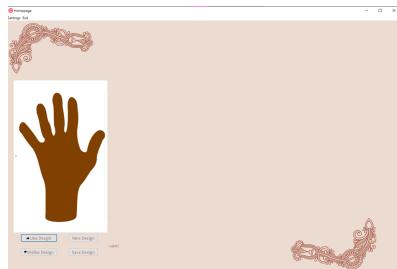
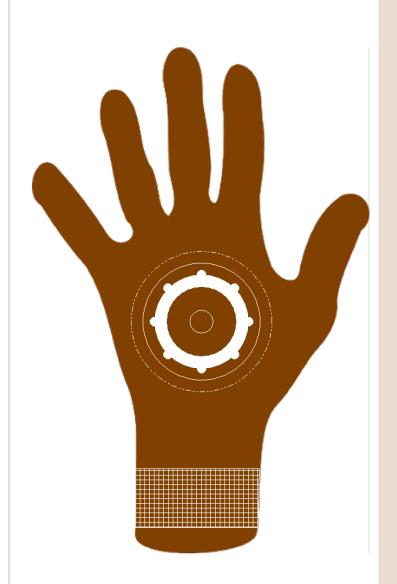
Despite my slight hiccup with using YouTube, overall the application was simple to use. As someone who struggles with technology , I'm so happy it catered for me and I feel comfortable using it.

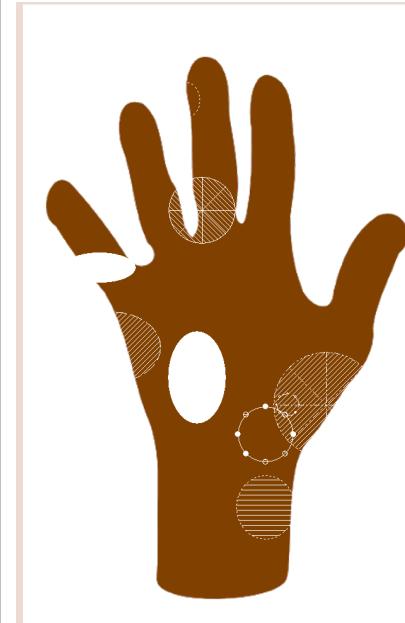
Reviews

Evaluation

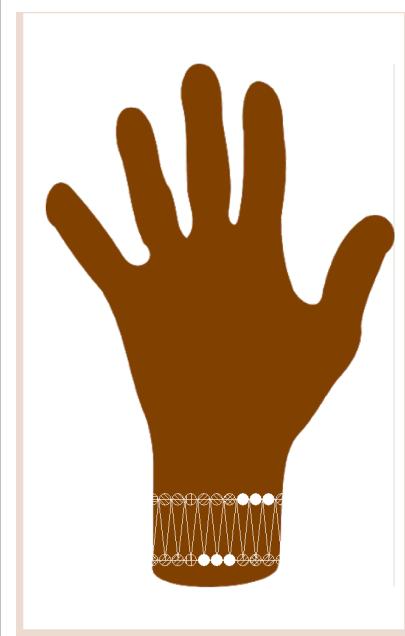
Success Criteria Met

| Success Criteria | How to Evidence | Evidence |
|---|---|--|
| An easy-to-use interface with limited typing | Screenshots of all input boxes (inc. buttons/ textboxes/ menus). Options should not require typing where possible (e.g. drop-down box instead of textbox) |  <p>These are all of the input boxes in my program. All of them are easily accessible and easy to read and no typing is required. Thus it meets my criteria.</p> |
| Menu bar | Screenshot of the menu bar and all of its options |  <p>Whilst the menu bar defers from my original design as it now has default formatting properties. As large proportion of apps already use this style, users are more used to it.</p> |
| My application has a gender-neutral interface | Screenshot and justification of the colour pallet used (with the precise colour hex codes) | <p>Main Background Colour: Tan(#ecdbd1)</p> <p>Main Body Colour: Dark Brown (#ab6f5f)</p> <p>Brown is a gender-neutral colour and is associated with henna, thus this is a suitable colour pallet for my application</p> |
| Lightweight Design | Screenshot of application pages which are not cluttered |  |

| | | |
|---|---|--|
| | |  <p>My application pages have a minimalist design, where elements are well spaced out. This makes it easy on the eye</p> |
| Colour pallet chosen in easy on the eye | <p>Screenshot of colour pallet used (with the precise colour hex codes) with light background colour and dark font colour</p> | <p>Body Colour : aa6d5b Background Colour : ecdbd1 These colours are high in contrast, but also are on the more muted side. This means the application is easier to read and doesn't strain the eye.</p>  |
| The program generates henna designs of all major cultures | <p>Screenshot of one generated henna design from each culture</p> | <p>South Asian</p>  <p>Middle Eastern</p> |



North African



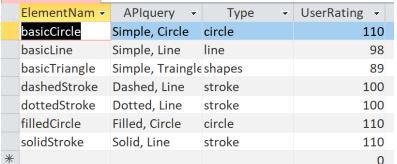
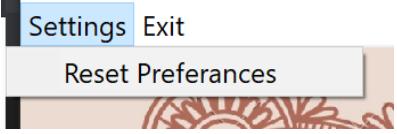
Using Neha's technical knowledge, I placed the designs into the above categories. This means my application is diverse and inclusive of all major cultures.

A clear instruction menu on how to use the application

Screenshot of the button to access the menu and the instructions

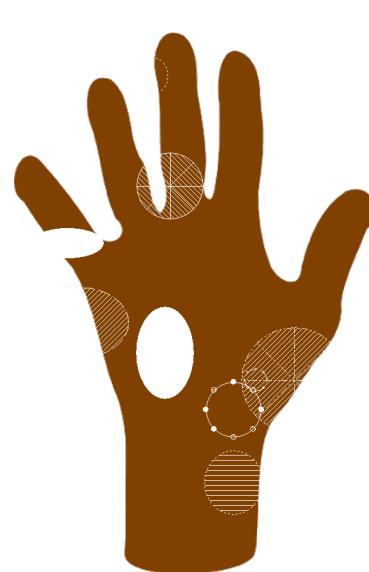
New Design : This button will generate a new henna design
Save Design : This button will allow you to save the henna design to your computer
Like and Dislike Buttons : This will rate the design and help customise designs to your taste
YouTube Tutorials : If you click the button next to each elements name, you will be taken to a YouTube tutorial for it
Henna Application Instructions : Click the option in the menu bar
Reset preferences : Go to menu bar -> settings -> reset
Element Trend : To view what elements you like the most and least you can open the Access database in the project file and view this

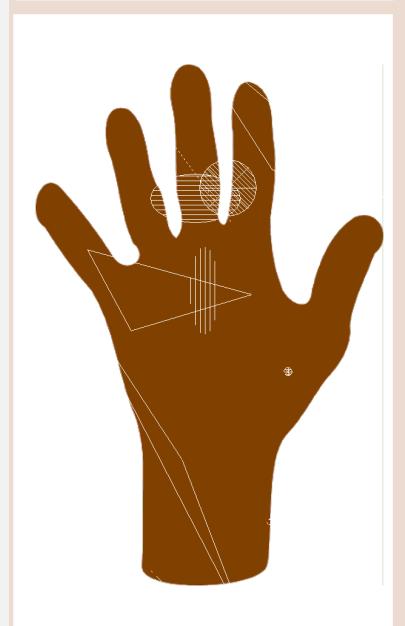
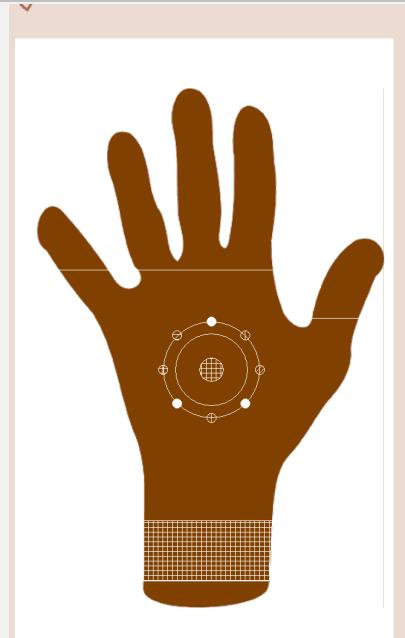
OK

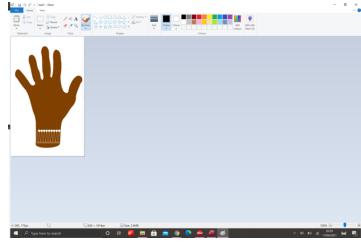
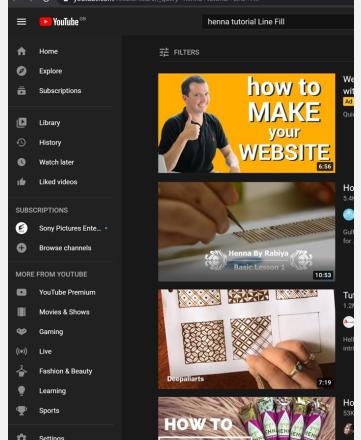
| | | |
|--|---|---|
| | | <h2>Application Manual</h2> <p>The application manual is an option in the menu bar, making it easily accessible. It provides clear, concise instructions on how to use the application.</p> |
| Designs being specialised to a user | Screenshot of database values changing according to like and dislike buttons , code and testing |  <p>The like and dislike button adapt the values in the database, changing the probability of each being selected.</p> <p>Coding and testing have been evidenced earlier in the document.</p> |
| The ability to like or dislike each design | Screenshots of the buttons |  <p>These buttons provide the users with the option to like and dislike each design. They are also graphical, which may make it easier for younger and older users to use.</p> |
| Option to reset preferences | Screenshots of the reset button |  <p>These buttons provide the users with the option to reset their preferences. It is located in the menu bar, making it easily accessible.</p> |

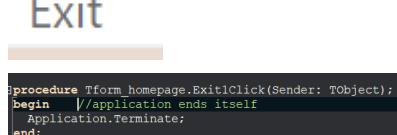
Designs being randomised each time

5 screenshots of different designs being produced consecutively and code





| | | |
|--|--|--|
| | |  |
| | | <p>Hence the new design button produces new, random designs.</p> |
| Ability to save the designs in an image format | Screenshot of the design open in a different application |  <p>The user can press save design which will allow the user to download the template. They can then open it into a different application, this will also allow them to send the henna designs to a device which doesn't have the app installed.</p> |
| Elements being linked to YouTube tutorials | Screenshot of buttons which open YouTube tutorials, screenshot of corresponding tutorial |  |

| | | | |
|--------------------------------|-------------------------------|---|--|
| | |  <p>Basic Circle</p> <p>Basic Line</p> <p>Line Fill</p> | <p>When the button is clicked the corresponding YouTube video is shown</p> |
| Option to exit the application | Screenshot of button and code |  <pre>procedure Tform_homepage.Exit1Click(Sender: TObject); begin //application ends itself Application.Terminate; end;</pre> | <p>When the button is clicked the application is exited.</p> |

Unmet Criteria or Tests

Success Criteria

My program met all the success criteria, and evidence has been provided above. Therefore this means that my project meets all the stakeholder requirements.

Functionality Testing

My program did not meet all of the functionally testing. It failed the following test:

- Are pictures of the isolated elements displayed?

Whilst initially I had believed this feature would have been useful I learnt otherwise in implementation:

- When I talked to my users they clearly understood what each element was from the label caption – hence it would only serve an aesthetic purpose
- Adding images to the database and any attempt to load them was causing the program to significantly slow down – which maybe an issue for people with older computers
- Adding images to the database was massively increasing the file size

Hence it became clear my original approach to this task was not suitable. It would require a different approach, such as remotely sorting these images in the cloud and loading them into the program. Give the time frame I had and the non-essential nature of it, I decided to get rid of this component.

Module Tests

My program now passes all the module tests, those that initially did have been modified and corrected to fix all the errors. Hence, I can be sure that my program works successfully.

An unexpected error I encountered was that if the user has clicked a YouTube tutorial and then immediately clicks the background, it opens the same YouTube tutorial again. I realised this was as the button stayed selected, hence when the background is clicked on it actually re-clicks the button. Thus it is not a technical error as the code performs the actions it should. Furthermore as it doesn't cause the program to crash , act in a majorly disruptive way or have a high likelihood of the user conducting this action, there was no need to look further into the issue.

Usability

To ensure my program is easily usable, it should meet the following criteria (this was established in design):

| Number | Requirement | Explanation | Met? | Evidence |
|--------|---|--|------|---|
| | | | Y/N | |
| 1 | An easy-to-use interface with limited typing | This will make my application intuitive and easy to use, which is especially important for older users who may already struggle with understanding new technology. | Y | No typing in any of the screen Input is via buttons or menu bars Look at homepage |
| 2 | Menu bar | A menu bar should be at the top of the page containing the essential links to navigate the application. This will allow a streamlined user journey and make the application simple to use. | Y | Look at homepage |
| 3 | My application has a gender-neutral interface | The colour pallet and design elements chosen should be relatively gender neutral in order to make my application feel welcoming to everyone. | Y | Main Background Colour: Tan(#ecdbd1) Main Body Colour: Dark Brown (#ab6f5f) Brown is a gender-neutral colour and is associated with henna, thus this is a suitable colour pallet for my application |
| 4 | Lightweight Design | The design should be simple ensuring it is not too complicated to use the application | Y | Minimalist design as can be seen below |
| 5 | Colour pallet chosen in easy on the eye | The colour palettes chosen should not be too busy or bold, and high in contrast. This will make my application more suitable for people with | Y | Body Colour : aa6d5b Background Colour : ecdbd1 |

| Number | Requirement | Explanation | Met? Y/N | Evidence |
|-------------------|-------------------|--|-------------|---|
| | | certain visual impairments . | | <p>These colours are high in contrast, but also are on the more muted side.</p>  |
| 6 (Additional) | Application Guide | The application manual is an option in the menu bar, making it easily accessible. It provides clear, concise instructions on how to use the application. | | <p>New Design : This button will generate a new henna design Save Design : This button will allow you to save the henna design to your computer Like and Dislike Buttons : This will rate the design and help customise designs to your taste YouTube Tutorials : If you click the button next to each element's name, you will be taken to a YouTube tutorial for it Henna Application Instructions : Click the option in the menu bar Reset preferences : Go to menu bar -> settings -> reset Element Trend : To view what elements you like the most and least, you can open the Access database in the project file and view this</p> <div style="text-align: right;"><input type="button" value="OK"/></div> |

Exhibit 26 : Final Load Screen

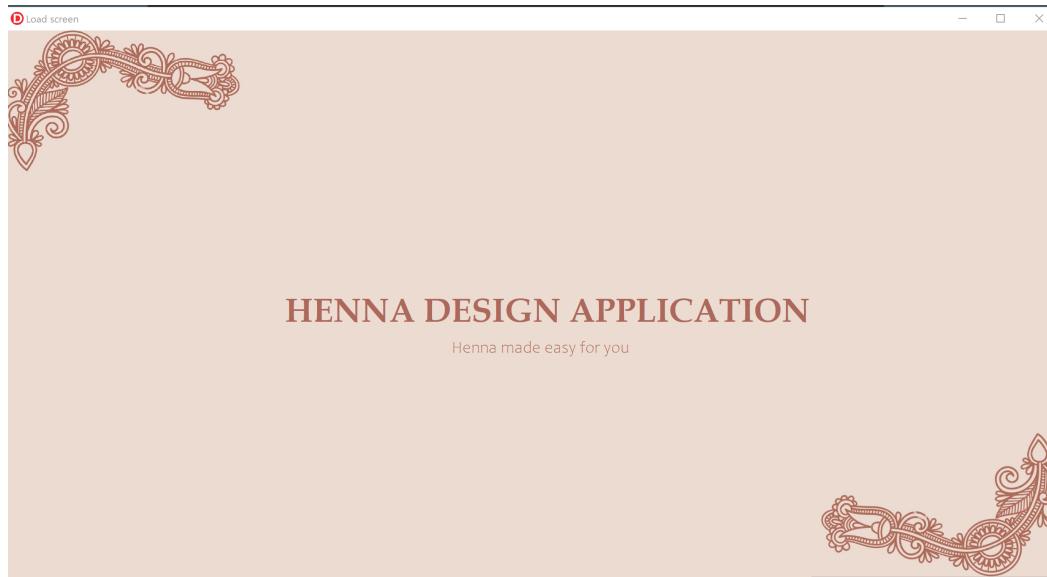


Exhibit 27 : Final Homepage



Limitations

Generating Designs

One of the limitations of my program is it only generates a certain amount of designs. For users like Neha, they wished for more templates and designs. Future versions of this application could allow users to add templates, in a similar way to Pinterest. This would allow a seemingly infinite number of henna designs to be generated.

As my application currently has the main focus of providing designs for practice to aid people in learning the basics of henna this is not the applications main focus. The user always has the option to download the design and make any adaptations they would like on word. Furthermore, the randomisation functions in my code allow a large combination of templates to be drawn.

Only for desktop

Currently my application is only available on desktop. This may be inconvenient for people who wish to access the app on a mobile device, for example if they are doing henna at a mehndi night.

Usually henna is practiced at home, as it requires a long time period for application and drying. This means users are likely to have access to a desktop device. Users also have the option to download designs and send it to a different device.

Designs produced only for one hand

My application only produces designs for the hand, not any other body part, which users may like to apply henna on, such as the feet.

As henna designs on the hand are easiest, it makes most sense to provide these designs to aid them to learn. Furthermore designs for each body part are specialised and adapted to that body part, hence it would require different elements and templates to the ones used by my program. It would make more sense to use/produce a totally separate application for different body parts.

Maintenance

Additional Changes

The following changes could be made with future maintenance to improve the application:

- Allow users to add designs
 - This would increase the number of available designs
 - This would provide the users with more practice material
- Allow hyperlinks to select design elements directly from canvas
 - This could be done via machine learning
 - This would make it easier for users to find specific tutorials
- Integrate YouTube videos into it
 - For users like Sarah who may have struggled with navigating back to the application after being on YouTube, this will create a simpler user journey
 - Instead of taking the user to an external YouTube page, the video could be integrated into the application
 - This would streamline the user journey

How is the application maintainable in the long term?

My code has a modular structure to it. Each procedure does one specific task. This would make changes to the code in future easy to find, implement and test, as it would have limited interaction without procedures.

My code is well annotated via comments. Using comments I have created section headers, for example for the linked list. This makes procedures easily findable. Furthermore the comments explain what the code does, allowing me or anyone else to easily understand it.

I have left the testing buttons in at the end of my code. These buttons were mainly used for module tests, thus if changes are made to any existing procedures, they can easily be tested.

My code has been well documented. This would allow someone to look through the documentation and find the explanations or information they are looking for in order to understand and modify the code.

Final Forms and interfaces

Load Screen

Exhibit 27: Load Screen in design view

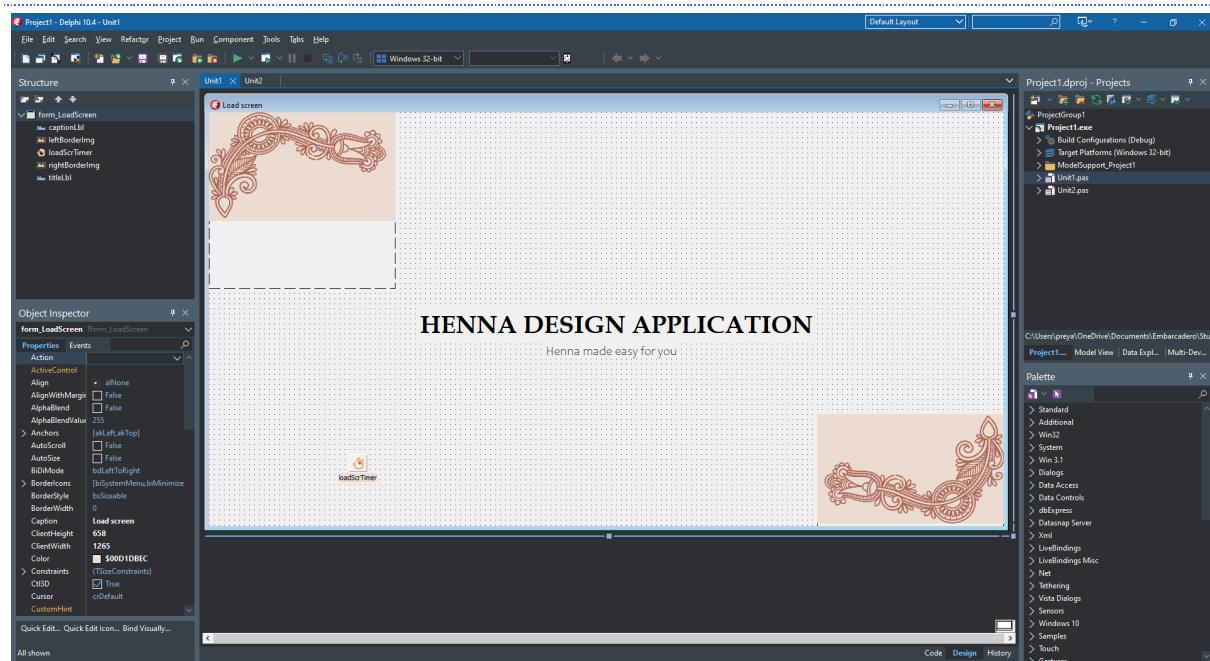
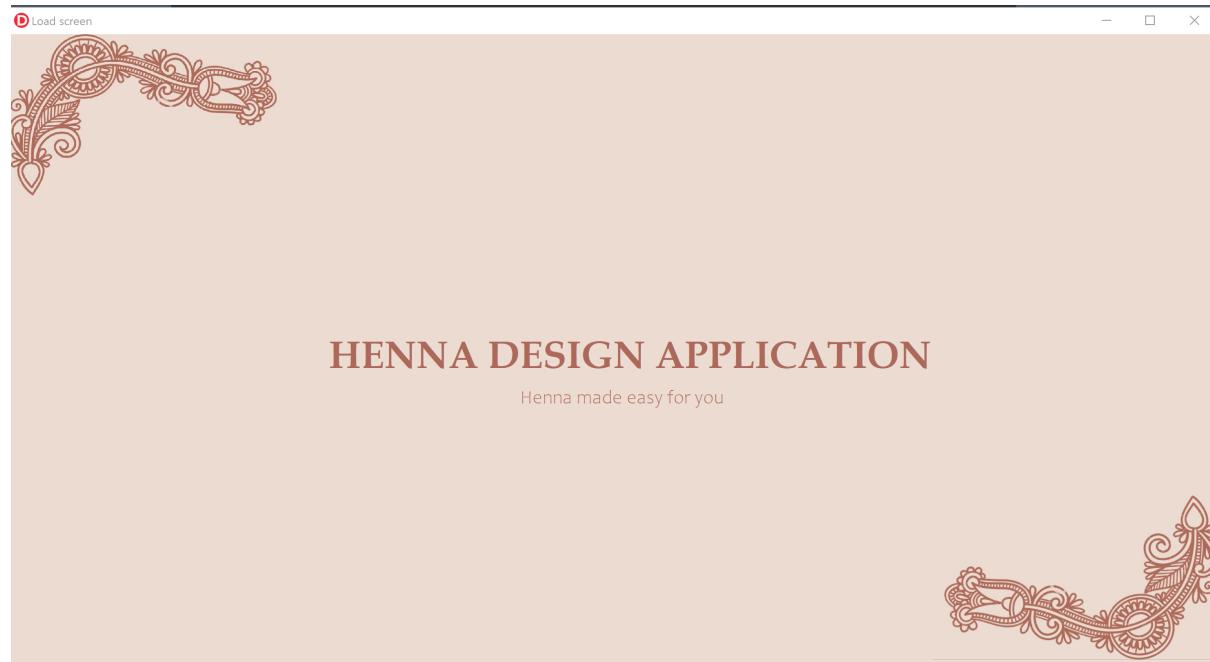


Exhibit 28: Load Screen when application is running



Homepage

Exhibit 29: Homepage in design view

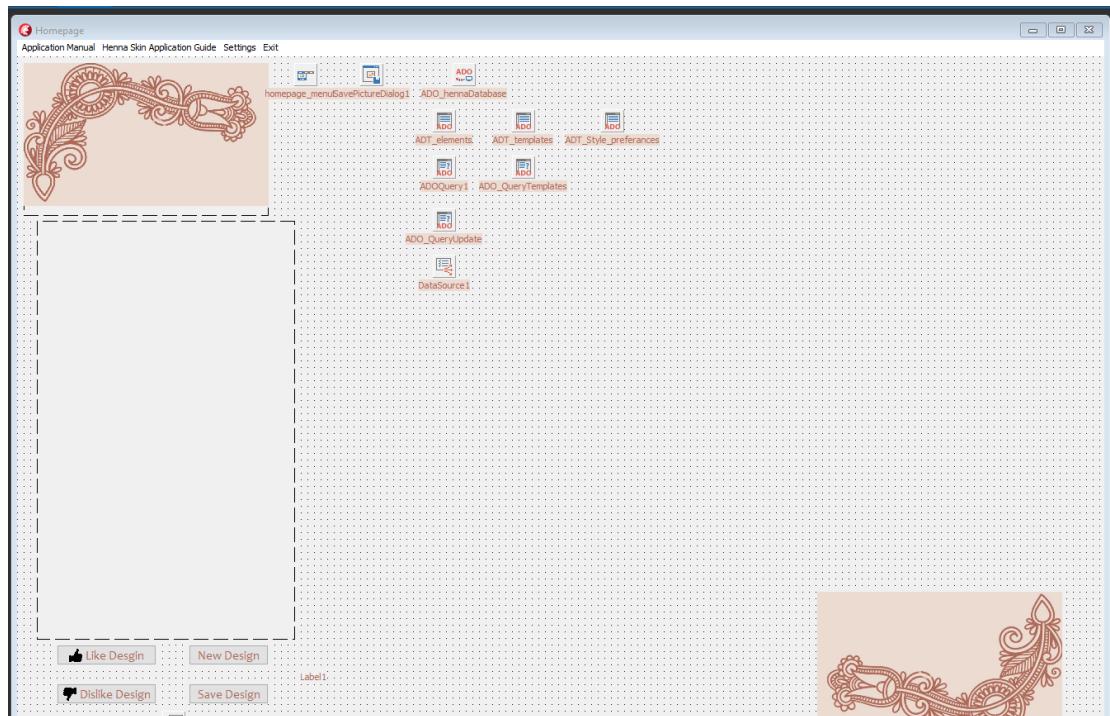
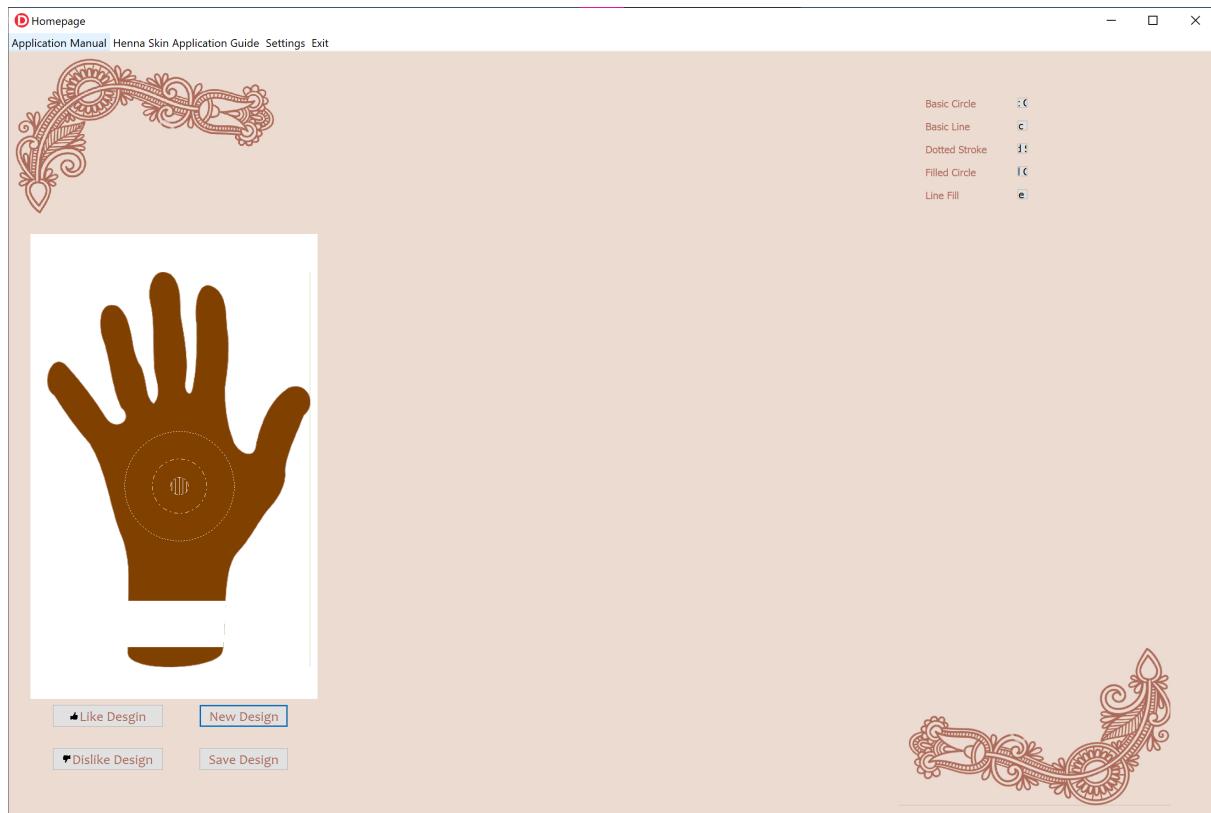


Exhibit 30: Homepage when application is running



Final code

Load Screen

Unit1.pas 18/04/2021 03:59:02

Page 1 of 1

```
1: unit Unit1;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
7:   System.Classes, Vcl.Graphics,
8:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls,
9:   Vcl.ExtDlgs,
10:  Vcl.Imaging.jpeg, Vcl.Imaging.pngimage;
11:
12: type
13:   TForm_LoadScreen = class(TForm)
14:     titleLbl: TLabel;
15:     leftBorderImg: TImage;
16:     rightBorderImg: TImage;
17:     captionLbl: TLabel;
18:     loadScrTimer: TTimer;
19:   procedure FormCreate(Sender: TObject);
20:   procedure LoadScrTimerActionClick(Sender: TObject);
21: private
22:   { Private declarations }
23: public
24:   { Public declarations }
25:
26: var
27:   form_LoadScreen: TForm_LoadScreen;
28:   mainTextColour : TColor;
29:
30: implementation
31:
32: {$R *.dfm}
33:
34: uses Unit2;
35:
36: procedure TForm_LoadScreen.LoadScrTimerActionClick(Sender: TObject);
37: begin
38:   //this function manages the timing and disappearance/transition of
39:   //the load screen
40:   form_LoadScreen.Hide;
41:   form_homepage.show;
42:   form_LoadScreen.loadScrTimer.Enabled := false;
43: end;
44: procedure TForm_LoadScreen.FormCreate(Sender: TObject);
45: begin
46:   //colour managment of the text on the page
47:   mainTextColour := RGB(172, 105, 91);
48:   titleLbl.Font.Color := mainTextColour;
49:   captionLbl.Font.Color := mainTextColour;
50: end;
51:
52:
53:
54: end.
```

```
1: unit Unit2;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.Math, System.SysUts,
7:   System.Variants, System.Classes, Vcl.Graphics,
8:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls, Vcl.Menus,
9:   Vcl.Imaging.jpeg, System.ImageList, Vcl.ImgList, Vcl.Buttons, Data.DB,
10:  Data.Win.ADODB, ShellAPI, Vcl.ExtDlgs;
11:
12: type
13:   TForm_homepage = class(TForm)
14:     rightBorderImg: TImage;
15:     leftBorderImg: TImage;
16:     homepage_menuBar: TMainMenu;
17:     Homepage2: TMenuItem;
18:     Exit1: TMenuItem;
19:     buttonIcons_ImgList: TImageList;
20:     likeDesign_btn: TBitBtn;
21:     dislikeDesign_btm: TBitBtn;
22:     newDesign_btn: TBitBtn;
23:     BitBtn1: TBitBtn;
24:     hennaDesign_canvas: TImage;
25:     ADO_hennaDatabase: TADOConnection;
26:     ADT_elements: TADOTable;
27:     DataSource1: TDataSource;
28:     ADOQuery1: TADOQuery;
29:     SavePictureDialog1: TSavePictureDialog;
30:     ADO_QueryUpdate: TADOQuery;
31:     ADT_templates: TADOTable;
32:     ADO_QueryTemplates: TADOQuery;
33:     ADT_Style_preferances: TADOTable;
34:     ResetPreferences1: TMenuItem;
35:     ApplicationManual1: TMenuItem;
36:     HennaApplicationGuide1: TMenuItem;
37:     procedure FormCreate(Sender: TObject);
38:     procedure Button1Click(Sender: TObject);
39:     procedure CircleClick(Sender: TObject);
40:     procedure Button2Click(Sender: TObject);
41:     procedure Button3Click(Sender: TObject);
42:     procedure Button4Click(Sender: TObject);
43:     procedure Button5Click(Sender: TObject);
44:     procedure Button6Click(Sender: TObject);
45:     procedure Button7Click(Sender: TObject);
46:     procedure Button8Click(Sender: TObject);
47:     procedure Button10Click(Sender: TObject);
48:     procedure Button11Click(Sender: TObject);
49:     procedure Button12Click(Sender: TObject);
50:     procedure Button13Click(Sender: TObject);
51:     procedure Button14Click(Sender: TObject);
52:     procedure Button15Click(Sender: TObject);
53:     procedure Button16Click(Sender: TObject);
```

```
53:     procedure Button17Click(Sender: TObject);
54:     procedure Button18Click(Sender: TObject);
55:     procedure Button19Click(Sender: TObject);
56:     procedure Button20Click(Sender: TObject);
57:     procedure Button21Click(Sender: TObject);
58:     procedure Button22Click(Sender: TObject);
59:     procedure APICall_buttonClick(Sender: TObject);
60:     procedure BitBtn1Click(Sender: TObject);
61:     procedure likeDesign_btnClick(Sender: TObject);
62:     procedure dislikeDesign_btmClick(Sender: TObject);
63:     procedure newDesign_btnClick(Sender: TObject);
64:     procedure makeNewLabels();
65:     procedure createButtons();
66:     procedure Exit1Click(Sender: TObject);
67:     procedure ResetPreferences1Click(Sender: TObject);
68:     procedure template3_btnClick(Sender: TObject);
69:     procedure ApplicationManual1Click(Sender: TObject);
70:     procedure HennaApplicationGuide1Click(Sender: TObject);
71: private
72:   { Private declarations }
73:     procedure youtubeAPI(Sender : TObject);
74: public
75:   { Public declarations }
76:
77: end;
78:
79: Type
80:   TCooridantes = record
81:     xCord : real ;
82:     yCord : real;
83:   end;
84:   TCooridantesBlock = record
85:     topLeft : TCooridantes ;
86:     topRight : TCooridantes ;
87:     bottomLeft : TCooridantes ;
88:     bottomRight: TCooridantes;
89:   end;
90:   TLinkedListItem = record
91:     item : string ;
92:     pointer : integer;
93:   end;
94:   TElementsDisplay = record
95:     labels : TLabel ;
96:     button : TButton ;
97:     image : TImage ;
98:   end;
99:
100:
101:
102: var
103:   form_homepage: TForm_homepage;
104:   //default form variables
105:   bodyTextColour,defaultColour, deafultPenColour: TColor;
106:   defaultBrushStyle : TBrushStyle;
107:   defaultPenStyle : TPenStyle;
```

```

108:  //coordiantes
109:  handCentre : TCooridantes;
110:  wristMaxPoints, fingerMaxPoints, thumbMaxPoints : TCoorida
      ntesBlock;
111:  //display elements
112:  elementsLabels : array[0..30] of TLabel ;
113:  elementsLl : array[0..30] of TLinkedListItem ;
114:  elementsLlSp: integer;
115:  elementsTop,test : Integer;
116:  dispalyArray : array[0..30] of TElementsDisplay ;
117:  //templates
118:  tempIndex : integer ;
119:  currentTemplate : integer;
120:  templates : array [1..100] of Tprocedure;
121:  implementation
122:
123: {$R *.dfm}
124:
125: uses Unit1;
126:
127: //Display elements
128:
129: procedure TForm_homepage.makeNewLabels(); //Dynamically crea
      tes new labels
130: var
131: i,j : integer;
132: begin
133: j := 0;
134: for i := 0 to 30 do
135: begin
136:   dispalyArray[i].labels := TLabel.Create(self);
137:   dispalyArray[i].labels.Parent := self;
138:   dispalyArray[i].labels.Top := 100 + j;
139:   dispalyArray[i].labels.Left := 2000;
140:   dispalyArray[i].labels.Cursor := crHandPoint ;
141:   j := 50 + j;
142: end;
143: end;
144:
145: procedure TForm_homepage.createButtons(); //Dynamically cre
      ates new buttons
146: var
147: i,j: integer;
148: begin
149: j := 0;
150: for i := 0 to 30 do
151: begin
152:   dispalyArray[i].button.Free;
153: end;
154:
155: for i := 0 to 30 do
156: begin
157:   dispalyArray[i].button := TButton.Create(Self);
158:   with dispalyArray[i].button DO
159: begin

```

```

160:      Visible := true;
161:      Caption := dispalyArray[i].labels.Caption;
162:      Parent := Self;
163:      Height := 23;
164:      Width := 23;
165:      Left := 2200;
166:      Top := 100 + j;
167:      if dispalyArray[i].labels.Caption <> '' then
168:      begin
169:          tempIndex := i;
170:          OnClick := youtubeAPI;
171:      end
172:      else
173:      begin
174:          visible := false;
175:      end;
176:      j := 50 + j;
177:  end;
178: end;
179: end;
180:
181:
182:
183: procedure populateElementLabels(); //Dynamically fills in the
   e labels captions
184: var
185: index,i,j,p, splitIndex: Integer;
186: tempString1, tempString2 : string ;
187: begin
188:     i := 0;
189:     p := elementsLlSp;
190:     while p <> -1 do //linked list traversal
191:     begin
192:         for index := 1 to (elementsLl[p].item.Length) do //gets item in correct format
193:             begin
194:                 if elementsLl[p].item[index] = upperCase(elementsLl[p].item[index]) then
195:                     begin
196:                         splitIndex := index;
197:                     end ;
198:                 end;
199:                 tempString1 := elementsLl[p].item.Substring(1,splitIndex-2) ;
200:                 tempString2 := elementsLl[p].item.Substring(splitIndex-1,
elementsLl[p].item.Length-1) ;
201:                 dispalyArray[i].labels.Caption := upperCase( elementsLl[p].item[1]) + tempString1 + ' ' + tempString2 ;
202:                 p := elementsLl[p].pointer ; //item which current one points to
203:                 i := i + 1;
204:             end;
205:             while i<=30 do //rest of labels are made 'invisible'
206:             begin
207:                 dispalyArray[i].labels.Caption := '';

```

```

208:     i := i +1;
209:   end;
210: end;
211:
212: //Canvas
213:
214: procedure clearCanvas();
215:
216: Var
217: bitmap : TBitmap;
218: i: Integer;
219: begin
220:   //Loads hand template into canvas
221:   bitmap := TBitmap.Create;
222:   try
223:     Bitmap.LoadFromFile('handOutlineV5.bmp');
224:     form_homepage.hennaDesign_canvas.canvas.Brush.Bitmap := B
      itmap;
225:     form_homepage.hennaDesign_canvas.canvas.FillRect(Rect(0,0
      ,1000,1000));
226:   finally
227:     form_LoadScreen.Canvas.Brush.Bitmap := nil;
228:     Bitmap.Free;
229:   end;
230: end;
231:
232: //New design specific
233:
234: procedure resetVariables();
235: Var
236: index: integer;
237: begin //sets deafult values for LL
238:   elementsLlSp:= -1;
239:   elementsTop := -1;
240:   for index := 0 to 30 do
241:   begin
242:     elementsLl[index].pointer :=-1;
243:     elementsLl[index].item := '';
244:   end;
245: end;
246:
247: procedure pickTemplate();
248: var
249: i, j,cummalativeWeighting, randomNumb,upperLimit, style : in
      teger ;
250: styleInfo : array[1..3,0..2] of integer ;
251: numbFound, templateFound : bool;
252: styleName : string;
253: begin
254:   cummalativeWeighting := 0;
255:   with form_homepage.ATD_Style_preferences do
256:   begin
257:     open;
258:     for i:=1 to 3 do    //populates array with styles and t
      heir data ranges

```

```

259:      begin
260:        recno:=i;
261:        styleInfo[i,0] := i;
262:        styleInfo[i,1] := cummalativeWeighting + 1;
263:        cummalativeWeighting := cummalativeWeighting + fieldv
264:          alues['User_Weighting'];
265:        styleInfo[i,2] := cummalativeWeighting;
266:      end;
267:    randomNumb := RandomRange(1,cummalativeWeighting);
268:    numbFound := false;
269:    j := 0;
270:    while numbFound = false do
271:    begin
272:      j := j + 1;
273:      upperLimit := styleInfo[j,2];
274:      if (upperLimit >= randomNumb) then
275:      begin
276:        style := j;
277:        numbFound := true;
278:      end;
279:    end;
280:    with form_homepage.ATD_Style_preferences do //selects a
281:      begin
282:        open;
283:        recno:=style;
284:        styleName := fieldvalues['Category'] + ' Weight';
285:      end;
286:      templateFound := false;
287:      with form_homepage.ATD_templates do
288:      begin
289:        open;
290:        while templateFound = false do //repeats until templat
e picked has picked style
291:        begin
292:          randomNumb := RandomRange(1, recordcount);
293:          recno:=randomNumb;
294:          if fieldvalues[styleName] = true then
295:          begin
296:            templateFound := true;
297:          end;
298:        end;
299:      end;
300:      currentTemplate := randomNumb;
301:    end;
302:
303:
304:
305:
306:
307: //Reccomendation
308:
309: procedure changeElementWeighting(dampaningFactor:string);
310:   var

```

```

311: currentElement : string ; //element weight changed according
   g to dampening factor
312: i : integer ;
313: begin
314: for i := 0 to 30 do
315: begin
316:   currentElement := elementsL1[i].item;
317:   with form_homepage.ADO_QueryUpdate do
318:   begin
319:     Active:=False;
320:     SQL.Clear;
321:     SQL.Add('UPDATE Elements');
322:     SQL.Add('SET UserRating = (UserRating*' + dampeningFactor
      +') WHERE ElementName = "' + currentElement + '"');
323:     ExecSQL ;
324:   end;
325:   form_homepage.ATD_Elements.open;
326:   form_homepage.ATD_Elements.Refresh;
327: end;
328: end;
329:
330:
331:
332: procedure changeStyleWeighting(dampeningFactor:string);
333: var      //style weight changed according to dampening factor
334: i : integer;
335: SA,NA,ME : bool;
336: begin
337:   with form_homepage.ATD_Templates do
338:   begin
339:     open;
340:     for i:=1 to recordcount do //checks what styles the
      template is
341:     begin
342:       recno:=i;
343:       if fieldvalues['TemplatesID'] = currentTemplate th
      en
344:       begin
345:         SA := FieldValues['South Asian Weight'];
346:         NA := FieldValues['North African Weight'];
347:         ME := FieldValues['Middle Eastern Weight'];
348:       end;
349:     end;
350:   end;
351:
352:   begin
353:   with form_homepage.ADO_QueryTemplates do //for all style
      s used in template dampening factor applied
354:   begin
355:     Active:=False;
356:     if SA = true then
357:     begin
358:       SQL.Clear;
359:       SQL.Add('UPDATE Style_Preferance');
360:       SQL.Add('SET User_Weighting = (User_Weighting*' + damp

```

```

        aningFactor +') WHERE Category = "South Asian";');
361:      ExecSQL ;
362:    end;
363:    if NA = true then
364:    begin
365:      SQL.Clear;
366:      SQL.Add('UPDATE Style_Preferance');
367:      SQL.Add('SET User_Weighting = (User_Weighting*' + damp
        aningFactor +') WHERE Category = "North African";');
368:      ExecSQL ;
369:    end;
370:    if ME = true then
371:    begin
372:      SQL.Clear;
373:      SQL.Add('UPDATE Style_Preferance');
374:      SQL.Add('SET User_Weighting = (User_Weighting*' + damp
        aningFactor +') WHERE Category = "Middle Eastern";');
375:      ExecSQL ;
376:    end;
377:  end;
378:  form_homepage.ATD_Style_preferences.open;
379:  form_homepage.ATD_Style_preferences.Refresh;
380: end;
381:
382:
383: end;
384:
385: procedure Tform_homepage.dislikeDesign_btmClick(Sender: TObject
ect);
386: begin //weightings decreased
387:   changeElementWeighting('0.9');
388:   changeStyleWeighting('0.9') ;
389: end;
390:
391: procedure Tform_homepage.likeDeign_btnClick(Sender: TObject)
;
392: begin //weightings increased
393:   changeElementWeighting('1.1');
394:   changeStyleWeighting('1.1') ;
395: end;
396:
397: //YouTube tutorials
398:
399: procedure Tform_homepage.youtubeAPI(Sender : TObject);
400: begin
401:   with (Sender as TButton) do
402:     shellExecute(Application.Handle,'open',PChar('https://www.
        youtube.com/results?search_query=henna+tutorial+' + caption
        ),nil,nil,0);
403: end;
404:
405: procedure Tform_homepage.APICall_buttonClick(Sender: TObject
);
406: var
407: tempString,test : string;

```

```

408: tempPWide,test2 : PWideChar;
409: temp3 : wideString ;
410: begin
411: temp3 := 'https://www.youtube.com/results?search_query=henn
a+henna+tutorial' + dispalyArray[tempIndex].labels.Caption ;
412: shellexecute(Application.Handle,'open',PChar('https://www.y
outube.com/results?search_query=henna+henna+tutorial'),nil,
nil,0);
413: end;
414:
415:
416: //Linked list
417: procedure addToLinkedList(item:string;var linkedList:array o
f TLinkedListItem;var lastIndexUsed : integer; var SP : inte
ger); //passed by var as value needs to be changed
418: var //adds an item to a specified linked list
419: tempIndex,p : integer;
420: placeFound, addItem : boolean;
421: begin
422: if SP = -1 then //if list is empty added to beggining
423: begin
424: lastIndexUsed := lastIndexUsed +1;
425: linkedList[lastIndexUsed].item := item;
426: SP := lastIndexUsed;
427: linkedList[lastIndexUsed].pointer := -1;
428: end
429: else if linkedList[SP].item = item then //if item is sam
e as SP item do not add
430: begin
431: addItem := false;
432: end
433: else if linkedList[SP].item > item then //if item is bigg
er than SP item add at start of LL
434: begin
435: lastIndexUsed := lastIndexUsed +1;
436: linkedList[lastIndexUsed].item := item;
437: tempIndex := SP;
438: SP := lastIndexUsed;
439: linkedList[lastIndexUsed].pointer := tempIndex;
440: end
441: else
442: begin
443: placeFound := false;
444: addItem := true;
445: p := SP ;
446:
447: while (placeFound = false) AND (linkedList[p].pointer <>
-1) AND(addItem = true) do
448: begin //loop through till item duplicate founds, plac
e is found or at end of LL
449: if item = linkedList[linkedList[p].pointer].item then
450: begin
451: addItem := false; //duplicate found then do
not add item to LL
452: end

```

```

453:      else if item > linkedList[linkedList[p].pointer].item
454:      then
455:          begin
456:              p := linkedList[p].pointer; //If item is bigger than current one move onto next item
457:          end
458:      else
459:          begin
460:              placeFound := true; //otherwise we have found our LL place to add item
461:          end;
462:      if addItem = true then //add item if no duplicate
463:          begin
464:              lastIndexUsed := lastIndexUsed +1;
465:              linkedList[lastIndexUsed].item := item;
466:              tempIndex := linkedList[p].pointer;
467:              linkedList[p].pointer := lastIndexUsed;
468:              linkedList[lastIndexUsed].pointer := tempIndex;
469:          end;
470:
471:
472:      end;
473:
474:
475:  end;
476:
477: procedure clearLinkedList(length : integer; var LL : array o
478:   f TLinkedListItem);
479: var index : integer;
480: begin //reset Ll to original values
481: elementsLlSP := -1;
482: for index := 0 to length do
483: begin
484:     elementsLl[index].pointer :=-1;
485:     elementsLl[index].item := '';
486: end;
487:
488: //the following procedure can be used to find the API query
489: //for a particular item
490: //procedure queryLLItems();
491: //var
492: //i,p : integer;
493: //begin
494: //with form_homepage.adt_elements do
495: //  begin
496: //    p := elementsLlSp;
497: //    while p <> -1 do
498: //        begin
499: //            for i := 1 to recordcount do
500: //                begin
501: //                    recno := i;
502: //                    if fieldvalues['ElementName'] = elementsLl[p].item

```

```

        : then
503: //           begin
504: //           //form2.label2.Caption:= form2.label2.Caption +
      fieldvalues['APIquery'];
505: //           end;
506: //           end;
507: //           p := elementsLl[p].pointer;
508: //           end;
509: //           end;
510: //end;
511:
512:
513: //randomisations
514: function toDraw(): bool ; //decides whether or not to draw
   an elements
515: begin
516:   if random(2) =1 then
517:     begin
518:       toDraw := True ;
519:     end
520:   else
521:     begin
522:       toDraw := false ;
523:     end;
524:   end;
525:
526:
527: function choosePenStyle(clearAllowed : bool): TPenStyle;
528: var //Picks a random pen style
529: numStyles,p, pInterval : integer;
530: test : string;
531: selectedStyle : TPenStyle;
532: begin
533:   if clearAllowed = true then
534:     begin
535:       p := 1 + Random(6);
536:     end
537:   else
538:     begin
539:       p := 1 + Random(5);
540:     end;
541:
542:   case p of
543:     1 : selectedStyle := psSolid;
544:     2 : selectedStyle := psDash;
545:     3 : selectedStyle := psDot;
546:     4 : selectedStyle := psDashDot;
547:     5 : selectedStyle := psDashDotDot;
548:     6 : selectedStyle := psClear;
549:   end ;
550:
551:   if selectedStyle = psSolid then //adds style to Ll
552:     begin
553:       addToLinkedList('solidStroke',elementsLl,elementsTop,
      elementsLlSp);

```

```

554:   end
555: else if selectedStyle = psDash then
556: begin
557:   addToLinkedList('dashedStroke',elementsLl,elementsTop
,elementsLlSp);
558: end
559: else if selectedStyle = psDot then
560: begin
561:   addToLinkedList('dottedStroke',elementsLl,elementsTop
,elementsLlSp);
562: end
563: else if selectedStyle <> psClear then
564: begin
565:   addToLinkedList('dashedStroke',elementsLl,elementsTo
p,elementsLlSp);
566:   addToLinkedList('dottedStroke',elementsLl,elementsTo
p,elementsLlSp);
567: end;
568:
569: choosePenStyle := selectedStyle;
570: end;
571:
572: function chooseStrokeStyle: TBrushStyle;
573: var //Picks a random brush style
574: numStyles,p, pInterval : integer;
575: test : string;
576: selectedStyle : TBrushStyle;
577: begin
578:   p := 1 + Random(8);
579:
580: case p of
581:   1 : selectedStyle := bsSolid;
582:   2 : selectedStyle := bsHorizontal;
583:   3 : selectedStyle := bsVertical;
584:   4 : selectedStyle := bsFDiagonal;
585:   5 : selectedStyle := bsBDiagonal;
586:   6 : selectedStyle := bsCross;
587:   7 : selectedStyle := bsDiagCross;
588:   8 : selectedStyle := bsClear;
589: end ;
590:
591: if selectedStyle = bsSolid then //adds style to Ll
592: begin
593:   addToLinkedList('solidFill',elementsLl,elementsTop,elem
entsLlSp);
594: end
595: else if (selectedStyle = bsCross) or (selectedStyle = bsDiag
Cross) then
596: begin
597:   addToLinkedList('crisscrossFill',elementsLl,elementsTop
,elementsLlSp);
598: end
599: else if selectedStyle <> bsClear then
600: begin
601:   addToLinkedList('lineFill',elementsLl,elementsTop,elem

```

```

        entsLlSp);
602:   end;
603:
604: chooseStrokeStyle := selectedStyle;
605:   end;
606:
607: //elements
608: procedure basicLine(x1,y1,x2,y2:integer);
609: begin
610: form_homepage.hennaDesign_canvas.Canvas.MoveTo(x1,y1); //Moves pen to start position
611: form_homepage.hennaDesign_canvas.canvas.lineto(x2,y2) ;//Draws line to end position
612: addToLinkedList('basicLine',elementsLl,elementsTop,elementsL
       lSp);
613: end;
614:
615: procedure basicCircle(topLeftX,topLeftY,bottomRightX,bottomR
      ightY:integer);
616: begin //Uses inbuilt ellipse function with equal x and y di
      stance to make a circle
617: form_homepage.hennaDesign_canvas.canvas.Ellipse(topLeftX,top
      LeftY,bottomRightX,bottomRightY);
618: addToLinkedList('basicCircle',elementsLl,elementsTop,element
      sLlSp);
619: end;
620:
621: procedure filledCircle(topLeftX,topLeftY,bottomRightX,bottom
      RightY:integer);
622: begin
623: with form_homepage.hennaDesign_canvas.Canvas do
624: begin
625: form_homepage.hennaDesign_canvas.Canvas.Brush.Style := choos
      eStrokeStyle;
626: brush.color := defaultColour; //changes brush colour to bla
      ck so whole shape drawn is in this colour
627: basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
628: brush.Color := clWhite; //changes brush colour back to deafu
      lt
629: form_homepage.hennaDesign_canvas.Canvas.Brush.Style := defau
      ltBrushStyle;
630: addToLinkedList('filledCircle',elementsLl,elementsTop,elemen
      tsLlSp);
631: end;
632: end;
633:
634: procedure dashedCircle(topLeftX,topLeftY,bottomRightX,bottom
      RightY:integer);
635: var
636: test : TPenStyle;
637: begin
638: with form_homepage.hennaDesign_canvas.Canvas do
639: begin
640: pen.style := (psDot); //chnages pen style - affects how bor
      der is draw

```

```

641: basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
642: test := psSolid;
643: pen.style := (test); // changed back to deafult
644: end;
645: addToLinkedList('basicCircle',elementsLl,elementsTop,element
   sLlSp);
646: end;
647:
648:
649: procedure circleEightPoints(topLeftX,topLeftY,bottomRightX,b
   ottomRightY:integer; var coordinates :array of TCooridantes)
   ;//passed by refeerence - so changes values
650: var //0 starts from the one directly above the centre, then
   counted clockwise
651: centre : TCooridantes;
652: radius : real;
653: begin
654:   addToLinkedList('basicCircle',elementsLl,elementsTop,eleme
   ntsLlSp);
655:   addToLinkedList('basicLine',elementsLl,elementsTop,element
   sLlSp);
656:   centre.xCord := (topLeftX + bottomRightX) / 2;
657:   centre.yCord := (topLeftY + bottomRightY) / 2;
658:   radius := bottomRightY - centre.yCord ;
659:   coordinates[0].xCord := centre.xCord;
660:   coordinates[0].yCord := topLeftY;
661:   coordinates[1].xCord := centre.xCord + radius * cos((7*PI)/
   4);
662:   coordinates[1].yCord := centre.yCord + radius * sin((7*PI)/
   4);
663:   coordinates[2].xCord := bottomRightX ;
664:   coordinates[2].yCord := centre.yCord;
665:   coordinates[3].xCord := centre.xCord + radius * cos((1*PI)/
   4);
666:   coordinates[3].yCord := centre.yCord + radius * sin((1*PI)/
   4);
667:   coordinates[4].xCord := centre.xCord ;
668:   coordinates[4].yCord := bottomRightY;
669:   coordinates[5].xCord := centre.xCord + radius * cos((3*PI)/
   4);
670:   coordinates[5].yCord := centre.yCord + radius * sin((3*PI)/
   4);
671:   coordinates[6].xCord := topLeftX ;
672:   coordinates[6].yCord := centre.yCord;
673:   coordinates[7].xCord := centre.xCord + radius * cos((5*PI)/
   4);
674:   coordinates[7].yCord := centre.yCord + radius * sin((5*PI)/
   4);
675: end;
676:
677: procedure innerReflectionCircle(topLeftX,topLeftY,bottomRigh
   tX,bottomRightY:integer) ;//passed by refeerence - so change
   s values
678: //0 starts from the one directly above the centre, then cou
   nted clockwise

```

```

679: Var
680: miniCircleCenters : array[0..7] of TCooridantes;
681: i, arrayHalfWay : integer;
682: begin
683:   basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
684:   circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY,miniCircleCenters) ;
685:   arrayHalfWay := (length(miniCircleCenters)DIV 2)-1 ;
686:   for i := 0 to arrayHalfWay do
687:   begin
688:     basicLine(trunc(miniCircleCenters[i].xCord),trunc(miniCircleCenters[i].yCord),trunc(miniCircleCenters[i+4].xCord),trunc(miniCircleCenters[i + 4].yCord))
689:   end;
690:   addToLinkedList('basicCircle',elementsLl,elementsTop,elementsLlSp);
691:   addToLinkedList('basicLine',elementsLl,elementsTop,elementsLlSp);
692: end;
693:
694: procedure miniCircleBorder(topLeftX,topLeftY,bottomRightX,bottomRightY:integer) //passed by reference - so changes values
695: //0 starts from the one directly above the centre, then counted clockwise
696: Var
697: miniCircleCenters : array[0..7] of TCooridantes;
698: i, arrayHalfWay : integer;
699: centre : TCooridantes;
700: radius,miniRadius : real;
701: begin
702:   centre.yCord := (topLeftY + bottomRightY) /2;
703:   radius := bottomRightY - centre.yCord ;
704:   miniRadius := radius/10;
705:   basicCircle(topLeftX,topLeftY,bottomRightX,bottomRightY);
706:   circleEightPoints(topLeftX,topLeftY,bottomRightX,bottomRightY,miniCircleCenters) ;
707:   arrayHalfWay := (length(miniCircleCenters)DIV 2)-1 ;
708:   for i := 0 to 7 do
709:   begin
710:     filledCircle(trunc(miniCircleCenters[i].xCord-miniRadius),trunc(miniCircleCenters[i].yCord-miniRadius),trunc(miniCircleCenters[i].xCord+miniRadius),trunc(miniCircleCenters[i].yCord+miniRadius))
711:   end;
712:   addToLinkedList('basicCircle',elementsLl,elementsTop,elementsLlSp);
713:   addToLinkedList('filledCircle',elementsLl,elementsTop,elementsLlSp);
714: end;
715:
716:
717: Procedure basicTriangle(x1,y1,x2,y2,x3,y3:integer);
718: begin //draws a basic triangle with corners x1,y1,x2,y2,x3,y3
719:   basicLine(x1,y1,x2,y2);

```

```

720:   basicLine(x1,y1,x3,y3);
721:   basicLine(x2,y2,x3,y3);
722:   addToLinkedList('basicTriangle',elementsLl,elementsTop,ele
    mentsLlSp);
723: end;
724:
725: //templates
726:
727: procedure template1(middleCord : TCooridantes); // so acutal
    data never changed - saftey measure
728: var
729: initialRadius,currentRadius : integer;
730: begin
731: currentTemplate := 1;
732: with form_homepage.hennaDesign_canvas.Canvas do
733:   begin
734:     initialRadius := 20;
735:     if toDraw() = true then //draws the middle circle
736:       filledCircle(trunc(middleCord.xCord-initialRadius),trunc(
      middleCord.yCord-initialRadius),trunc(middleCord.xCord+initial
      Radius),trunc(middleCord.yCord+initialRadius));
737:     currentRadius := initialRadius*3;
738:     if toDraw() = true then //draws the next circle
739:       basicCircle(trunc(middleCord.xCord-currentRadius),trunc(m
      iddleCord.yCord-currentRadius),trunc(middleCord.xCord+current
      tRadius),trunc(middleCord.yCord+currentRadius));
740:     currentRadius := initialRadius*4;
741:     if toDraw() = true then //draws the next circle
742:       miniCircleBorder(trunc(middleCord.xCord-currentRadius),t
      runc(middleCord.yCord-currentRadius),trunc(middleCord.xCord+
      currentRadius),trunc(middleCord.yCord+currentRadius));
743:     currentRadius := initialRadius*5;
744:     if toDraw() = true then
745:       begin //draws the next circle with a solis fill between t
         he prvious one
746:         basicCircle(trunc(middleCord.xCord-currentRadius),trunc(m
      iddleCord.yCord-currentRadius),trunc(middleCord.xCord+current
      tRadius),trunc(middleCord.yCord+currentRadius));
747:         Brush.Style := bsSolid;
748:         Brush.Color := clWhite;
749:         FloodFill(trunc((middleCord.xCord-currentRadius)+(initialR
      adius*2.50)),trunc((middleCord.yCord-currentRadius)+((initial
      Radius)*2.5)), clWhite, fsBorder);
750:         Brush.Style := defaultBrushStyle;
751:       end;
752:
753:     currentRadius := initialRadius*6;
754:     if toDraw() = true then
755:       begin //draws the next circle
756:         pen.Style := choosePenStyle(false);
757:         basicCircle(trunc(middleCord.xCord-currentRadius),trunc(m
      iddleCord.yCord-currentRadius),trunc(middleCord.xCord+current
      tRadius),trunc(middleCord.yCord+currentRadius));
758:       end;
759:       pen.Style := defaultPenStyle;

```

```
760:  if toDraw() = true then
761:    begin //fills in wrist section
762:      basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord),trunc(wristMaxPoints.topRight.xCord),trunc(wristMaxPoints.topRight.yCord));
763:      basicLine(trunc(wristMaxPoints.bottomLeft.xCord), trunc(wristMaxPoints.bottomLeft.yCord),trunc(wristMaxPoints.bottomRight.xCord),trunc(wristMaxPoints.bottomRight.yCord));
764:      basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(wristMaxPoints.topLeft.yCord),trunc(wristMaxPoints.bottomLeft.xCord), trunc(wristMaxPoints.bottomLeft.yCord));
765:      basicLine(trunc(wristMaxPoints.topRight.xCord), trunc(wristMaxPoints.topRight.yCord),trunc(wristMaxPoints.bottomRight.xCord),trunc(wristMaxPoints.bottomRight.yCord));
766:      Brush.Style := chooseStrokeStyle;
767:      Brush.Color := clWhite;
768:      FloodFill(trunc((wristMaxPoints.topLeft.xCord)+10),trunc((wristMaxPoints.topLeft.yCord+10)), clWhite, fsBorder);
769:    end;
770:    if toDraw() = true then
771:      begin //fills in finger section
772:        pen.Color := clBlack;
773:        repeat
774:          Brush.Style := chooseStrokeStyle;
775:          until (Brush.Style <> bsSolid) ;
776:          rectangle(trunc(fingerMaxPoints.topLeft.xCord),trunc(fingerMaxPoints.topLeft.yCord) ,trunc(fingerMaxPoints.bottomRight.xCord),trunc(fingerMaxPoints.bottomRight.yCord ));
777:          FloodFill(trunc((fingerMaxPoints.topLeft.xCord)+10),trunc((fingerMaxPoints.topLeft.yCord+10)), clBlack, fsBorder);
778:          pen.Color := defaultColour;
779:          rectangle(trunc(fingerMaxPoints.topLeft.xCord),trunc(fingerMaxPoints.topLeft.yCord) ,trunc(fingerMaxPoints.bottomRight.xCord),trunc(fingerMaxPoints.bottomRight.yCord ));
780:          pen.Color := clBlack;
781:          rectangle(trunc(thumbMaxPoints.topLeft.xCord),trunc(thumbMaxPoints.topLeft.yCord) ,trunc(thumbMaxPoints.bottomRight.xCord),trunc(thumbMaxPoints.bottomRight.yCord ));
782:          FloodFill(trunc((thumbMaxPoints.topLeft.xCord)+10),trunc((thumbMaxPoints.topLeft.yCord+10)), clBlack, fsBorder);
783:          pen.Color := defaultColour;
784:          rectangle(trunc(thumbMaxPoints.topLeft.xCord),trunc(thumbMaxPoints.topLeft.yCord) ,trunc(thumbMaxPoints.bottomRight.xCord),trunc(thumbMaxPoints.bottomRight.yCord ));
785:          Brush.Style := defaultBrushStyle;
786:        end;
787:      end;
788:    end;
789:
790: procedure template2(middleCord : TCooridantes);
791: var
792:   heightDif,widthDif,widthInterval : real;
793:   i : integer;
794: begin
795:   currentTemplate := 2;
```

```

796:   with form_homepage.hennaDesign_canvas.Canvas do
797:     begin
798:       widthDif := wristMaxPoints.topRight.xCord - wristMaxPoints.topLeft.xCord ;
799:       widthInterval := widthDif / 10 ;
800:       if toDraw() = true then
801:         begin //does wrist border
802:           basicLine(trunc(wristMaxPoints.topLeft.xCord), trunc(
803:             wristMaxPoints.topLeft.yCord), trunc(wristMaxPoints.topRight.
804:               xCord), trunc(wristMaxPoints.topRight.yCord));
805:           basicLine(trunc(wristMaxPoints.bottomLeft.xCord), trunc(
806:             wristMaxPoints.bottomLeft.yCord), trunc(wristMaxPoints.bot-
807:               tomRight.xCord), trunc(wristMaxPoints.bottomRight.yCord));
808:         end;
809:         if toDraw() = true then
810:           begin //does wrist triangle fill
811:             for i := 0 to 9 do
812:               begin
813:                 basicTriangle((trunc( wristMaxPoints.topLeft.x
814:                   Cord + (widthInterval *i))),trunc(wristMaxPoints.bottomLeft.
815:                     yCord),trunc((widthInterval/2)+( wristMaxPoints.topLeft.xCor-
816:                       d )+ (widthInterval *i)),trunc(wristMaxPoints.topLeft.yCord)
817:                         ,trunc(wristMaxPoints.topLeft.xCord + (widthInterval *(i+1))
818:                           ),trunc(wristMaxPoints.bottomLeft.yCord)) ;
819:               end;
820:             end;
821:             if toDraw() = true then
822:               begin //does filled circles on traingle border
823:                 for i := 0 to 10 do
824:                   begin
825:                     filledCircle((trunc( wristMaxPoints.topLeft.xCord
826:                       + (widthInterval *i)-10)),trunc(wristMaxPoints.bottomLeft.yC-
827:                         ord-10),(trunc( wristMaxPoints.topLeft.xCord + (widthInterv-
828:                           al *i)+10)),trunc(wristMaxPoints.bottomLeft.yCord+10));
826:                     filledCircle((trunc( wristMaxPoints.bottomLeft.xCord
827:                       + (widthInterval *i)-10)),trunc(wristMaxPoints.topLeft.yC-
828:                         ord - 10),(trunc( wristMaxPoints.bottomLeft.xCord + (widthIn-
829:                           terval *i)+10)),trunc(wristMaxPoints.topLeft.yCord)+10);
827:                   end;
828:                 end;

```

```

829:
830:   end;
831: end;
832:
833: procedure template3();
834: var
835:   i, element,tempNum : integer;
836:   middleCord : TCooridantes ;
837: begin
838:   for i := 1 to 20 do //does max 20 elements
839:   begin
840:     if toDraw() = true then
841:     begin
842:       form_homepage.hennaDesign_canvas.canvas.pen.style := choosePenStyle(true) ;
843:       form_homepage.hennaDesign_canvas.canvas.brush.Style := chooseStrokeStyle() ;
844:       middleCord.xCord := Random(form_homepage.hennaDesign_canvas.Width-200)+100; // random number but ensures wont be outside canvas
845:       middleCord.yCord := Random(form_homepage.hennaDesign_canvas.height-200)+100;
846:       element := random(7)+1;
847:       tempNum := random(100) ;
848:       case element of //picks a random element
849:         1 : basicLine(Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height));
850:         2 : basicCircle(trunc(middleCord.xCord-random(100)),trunc(middleCord.yCord-random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
851:         3 : filledCircle(trunc(middleCord.xCord-random(100)),trunc(middleCord.yCord-random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
852:         4 : dashedCircle (trunc(middleCord.xCord-random(100)),trunc(middleCord.yCord-random(100)),trunc(middleCord.xCord+random(100)),trunc(middleCord.yCord+random(100)));
853:         5 : basicTriangle(Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height),Random(form_homepage.hennaDesign_canvas.Width),Random(form_homepage.hennaDesign_canvas.height));
854:         6 : innerReflectionCircle(trunc(middleCord.xCord-tempNum),trunc(middleCord.yCord-tempNum),trunc(middleCord.xCord+tempNum),trunc(middleCord.yCord+tempNum));
855:         7 : miniCircleBorder(trunc(middleCord.xCord-tempNum),trunc(middleCord.yCord-tempNum),trunc(middleCord.xCord+tempNum),trunc(middleCord.yCord+tempNum));
856:       end;
857:     end;
858:
859:   end;
860:
```

```

861: end;
862:
863: procedure runTemplate();
864: begin
865:   case currentTemplate of
866:     1 : template1(handCentre) ;
867:     2 : template2(handCentre);
868:     3 : template3();
869:   end;
870: end;
871:
872: procedure Tform_homepage.BitBtn1Click(Sender: TObject);
873: begin
874:   messagedlg('Do not use a prexsiting filename',mtWarning ,
mbOKCancel , 0);
875:   if SavePictureDialog1.Execute then
876:   begin
877:     if SavePictureDialog1.FileName<>'' then
878:     begin
879:       hennaDesign_canvas.Picture.SaveToFile(SavePictureDialo
g1.FileName);
880:     end;
881:   end;
882: end;
883:
884: //menu bar
885:
886: procedure Tform_homepage.ApplicationManual1Click(Sender: TObject);
887: begin //loads a pop up with the instruction guide
888:   showmessage('New Design : This button will generate a new
henna design '+#13#10+#13#10+
889:             'Save Design : This button will allow you to s
ave the henna design to your computer'+#13#10+#13#10+
890:             'Like and Dislike Buttons : This will rate the
design and help cutomise designs to your taste'+#13#10+ #13
#10+
891:             'YouTube Tutoirals : If you click the button n
ext to each elements name, you will be taken to a YouTube tu
torial for it'+#13#10+#13#10+
892:             'Henna Application Instructions : Click the op
tion in the menu bar'+#13#10+ #13#10+
893:             'Reset preferences : Go to menu bar -> setting
s -> reset' +#13#10+#13#10+
894:             'Element Trend : To view what elements you lik
e the most and least you can open the Access database in the
project file and view this');
895: end;
896:
897: procedure Tform_homepage.Exit1Click(Sender: TObject);
898: begin //application ends itself
899:   Application.Terminate;
900: end;
901:
902: procedure Tform_homepage.HennaApplicationGuide1Click(Sender:

```

```

        TObject);
903: begin
904: shellexecute(Application.Handle,'open',PChar('https://www.wi
      kihow.com/Do-a-Henna-Tattoo'),nil,nil,0);
905: end;
906:
907: procedure Tform_homepage.ResetPreferences1Click(Sender: TObject);
908: var
909: i : integer ; //resets to initial values
910: begin
911:   with form_homepage.adoquery1 do
912:     begin
913:       SQL.Clear;
914:       SQL.Add('UPDATE Elements');
915:       SQL.Add('SET UserRating = 100');
916:       ExecSQL ;
917:     end;
918:
919:   with form_homepage.adoquery1 do
920:     begin
921:       SQL.Clear;
922:       SQL.Add('UPDATE Style_Preference');
923:       SQL.Add('SET User_Weighting = 100');
924:       ExecSQL ;
925:     end;
926: end;
927:
928:
929: //form loading
930:
931: procedure Tform_homepage.FormCreate(Sender: TObject);
932: Var
933: bitmap : TBitmap;
934: i,index: Integer;
935: begin
936:   makeNewLabels();
937:   //Loads hand template into canvas
938:   bitmap := TBitmap.Create;
939:   try
940:     Bitmap.LoadFromFile('handOutlineV5.bmp');
941:     form_homepage.hennaDesign_canvas.canvas.Brush.Bitmap := B
      itmap;
942:     hennaDesign_canvas.canvas.FillRect(Rect(0,0,1000,1000));
943:   finally
944:     form_LoadScreen.Canvas.Brush.Bitmap := nil;
945:     Bitmap.Free;
946:   end;
947:
948: //sets values of global coordinate values
949: handCentre.xCord := 325;
950: handCentre.yCord := 550;
951: wristMaxPoints.topLeft.xCord := 212;
952: wristMaxPoints.topLeft.yCord := 800;
953: wristMaxPoints.bottomLeft.xCord := 212;

```

```

954:   wristMaxPoints.bottomLeft.yCord :=900;
955:   wristMaxPoints.topRight.xCord :=425;
956:   wristMaxPoints.topRight.yCord :=800;
957:   wristMaxPoints.bottomRight.xCord :=420;
958:   wristMaxPoints.bottomRight.yCord :=900;
959:
960:
961:   fingerMaxPoints.topLeft.xCord :=2;
962:   fingerMaxPoints.topLeft.yCord :=75;
963:   fingerMaxPoints.bottomLeft.xCord :=2;
964:   fingerMaxPoints.bottomLeft.yCord :=385;
965:   fingerMaxPoints.topRight.xCord :=438;
966:   fingerMaxPoints.topRight.yCord :=75;
967:   fingerMaxPoints.bottomRight.xCord :=438;
968:   fingerMaxPoints.bottomRight.yCord :=385;
969:
970:   thumbMaxPoints.topLeft.xCord :=471;
971:   thumbMaxPoints.topLeft.yCord :=330;
972:   thumbMaxPoints.bottomLeft.xCord :=471;
973:   thumbMaxPoints.bottomLeft.yCord :=445;
974:   thumbMaxPoints.topRight.xCord :=617;
975:   thumbMaxPoints.topRight.yCord :=350;
976:   thumbMaxPoints.bottomRight.xCord :=617;
977:   thumbMaxPoints.bottomRight.yCord :=465;
978:
979: //sets deafult styles
980: defaultBrushStyle := bsClear;
981: defaultPenStyle := psSolid;
982: defaultColour := ClWhite;
983: form_homepage.hennaDesign_canvas.Canvas.Brush.Color := deafaultColour;
984: form_homepage.hennaDesign_canvas.Canvas.pen.Color := deafultColour;
985: form_homepage.hennaDesign_canvas.Canvas.Brush.Style := deafaultBrushStyle;
986:
987: //sets deafult values for BST
988: elementsLlSp:= -1;
989: elementsTop := -1;
990: for index := 0 to 30 do
991: begin
992:   elementsLl[index].pointer:=-1;
993: end;
994: end;
995:
996: //new deisgn - at end as requires many other procedures
997: procedure Tform_homepage.newDesign_btnClick(Sender: TObject)
998: ;
999: begin
1000: clearCanvas();
1001: resetVariables();
1002: clearLinkedList(30,elementsLl) ;
1003: pickTemplate();
1004: runTemplate();
1005: populateElementLabels();

```

```
1005: createButtons();  
1006: end;  
1007:  
1008:  
1009:  
1010:  
1011: //test buttons - these buttons were used for unit testing  
1012: //they have been left in case of the need to debug future er  
rors  
1013:  
1014: procedure TForm_homepage.Template3.BtnClick(Sender: TObject)  
1015: ;  
1016: begin  
1017: Template3();  
1018: end;  
1019: procedure TForm_homepage.Button10Click(Sender: TObject);  
1020: begin  
1021: Template1(handCentre);  
1022: end;  
1023:  
1024: procedure TForm_homepage.Button11Click(Sender: TObject);  
1025: begin  
1026: Template2(handCentre);  
1027: end;  
1028:  
1029: procedure TForm_homepage.Button12Click(Sender: TObject);  
1030: var  
1031: i : integer;  
1032: begin  
1033: //label1.Caption := '';  
1034: for i := 0 to 30 do  
1035: begin  
1036: //label1.Caption := label1.Caption + ' ' + elementsLl[i]  
     .item;  
1037: end;  
1038:  
1039: end;  
1040:  
1041: procedure TForm_homepage.Button13Click(Sender: TObject);  
1042: begin  
1043: clearLinkedList(30,elementsLl) ;  
1044: end;  
1045:  
1046: procedure TForm_homepage.Button14Click(Sender: TObject);  
1047: begin  
1048: if 'filledCircle' > 'basicLine' then  
1049: begin  
1050: //label1.Caption := 'works';  
1051: end  
1052: else  
1053: begin  
1054: //label1.Caption := 'nope';  
1055: end;  
1056: end;
```

```

1057:
1058: procedure Tform_homepage.Button15Click(Sender: TObject);
1059: var
1060:   i,p: Integer;
1061: begin
1062:   //label1.caption:= '';
1063:   p := elementsLlSp;
1064:   while p <> -1 do
1065:     begin
1066:       //label1.Caption := label1.Caption + ' ' + elementsLl[p]
1067:       .item;
1068:       p := elementsLl[p].pointer
1069:     end;
1070:   end;
1071: procedure Tform_homepage.Button16Click(Sender: TObject);
1072: var
1073: testLabel : TLabel ;
1074: begin
1075:   testLabel := TLabel.Create(self);
1076:   testLabel.Parent := self;
1077:   testLabel.Top := 20;
1078:   testLabel.Left := 20;
1079:   testLabel.Caption := 'test label';
1080: end;
1081:
1082: procedure Tform_homepage.Button17Click(Sender: TObject);
1083: var
1084: index,i,p, splitIndex: Integer;
1085: tempString1, tempString2 : string ;
1086: begin
1087:   i := 0;
1088:   p := elementsLlSp;
1089:   while p <> -1 do
1090:     begin
1091:       for index := 1 to (elementsLl[p].item.Length) do
1092:         begin
1093:           if elementsLl[p].item[index] = upperCase(elementsLl[p]
1094:             .item[index]) then
1095:               begin
1096:                 splitIndex := index;
1097:               end ;
1098:               tempString1 := elementsLl[p].item.Substring(1,splitInde
1099:               x-2) ;
1100:               tempString2 := elementsLl[p].item.Substring(splitIndex-
1101:               1,elementsLl[p].item.Length-1) ;
1102:               dispalyArray[i].labels.Caption := upperCase( elementsLl
1103:               [p].item[1]) + tempString1 + ' ' + tempString2 ;
1104:               p := elementsLl[p].pointer ;
1105:               i := i + 1;
1106:             end;
1107:             while i<=30 do
1108:               begin
1109:                 dispalyArray[i].labels.Caption := '';

```

```

1161: end;
1162:
1163: procedure Tform_homepage.Button22Click(Sender: TObject);
1164: begin
1165: var
1166: i,j: integer;
1167: begin
1168: j := 0;
1169: for i := 0 to 30 do
1170:   begin
1171:     dispalyArray[i].button := TButton.Create(Self);
1172:     with dispalyArray[i].button DO
1173:       begin
1174:         Visible := true;
1175:         Caption := dispalyArray[i].labels.Caption;
1176:         Parent := Self;
1177:         Height := 23;
1178:         Width := 100;
1179:         Left := 2200;
1180:         Top := 100 + j;
1181:         if dispalyArray[i].labels.Caption <> '' then
1182:           begin
1183:             tempIndex := i;
1184:             OnClick := youtubeAPI;
1185:           end;
1186:         end;
1187: //   dispalyArray[i].button.OnClick := shellexecute(handle,
1188: //   'open','https://www.youtube.com/results?search_query=simple+
1189: //   circle+henna',nil,nil,0);
1190: //   //try assinging it to an invisible buttons code
1191: //   // dispalyArray[i].button.OnClick := (Handle, 'open', PC
1192: //   har('http://www.google.com/'), nil, nil, SW_SHOW);
1193: //   // dispalyArray[i].button.Click := shellexecute(handle,
1194: //   'open','https://www.youtube.com/results?search_query=simple+
1195: //   circle+henna',nil,nil,0);
1196:
1197: //procedure TForm2.Button23Click(Sender: TObject);
1198: //var
1199: //i,j,k,p: integer;
1200: //begin
1201: //j := 0;
1202: //  with form2.adt_elements do
1203: //    begin
1204: //      open ;
1205: //      p := elementsLlSp;
1206: //      while p <> -1 do
1207: //        begin
1208: //          for i := 1 to recordcount do
1209: //            begin
1210: //              recno := i;

```

```

1211: //      if fieldvalues['ElementName'] = elementsL1[p].item
1212: //      : then
1213: //          begin
1214: //              dispalyArray[i].image:= fieldvalues['Images'];
1215: //          end;
1216: //          p := elementsL1[p].pointer;
1217: //      end;
1218: //  end;
1219: //end;
1220: //
1221: //  dispalyArray[i].image := TImage.Create(self) ;
1222: //  with dispalyArray[i].image DO
1223: //  begin
1224: //      dispalyArray[i].image := TImage.Create(self);
1225: //      dispalyArray[i].image.Picture := ;
1226: //      dispalyArray[i].image.Parent := PageControl1.ActivePa
ge;
1227: //      if dispalyArray[i].labels.Caption <> '' then
1228: //      begin
1229: //          tempIndex := i;
1230: //          OnClick := youtubeAPI;
1231: //      end;
1232: //  end;
1233: //
1234: //  j := 50 + j;
1235: // end;
1236: //end;
1237: //end;
1238:
1239:
1240: procedure TForm_homepage.Button2Click(Sender: TObject);
1241: begin
1242:     filledCircle(10,10,200,200);
1243: end;
1244:
1245: procedure TForm_homepage.Button3Click(Sender: TObject);
1246: begin
1247:     dashedCircle(10,10,200,200);
1248: end;
1249:
1250: procedure TForm_homepage.Button4Click(Sender: TObject);
1251: begin
1252:     form_homepage.hennaDesign_canvas.canvas.pen.Style := choose
PenStyle(true);
1253:     basicLine(10,10,10,1000) ;
1254:     form_homepage.hennaDesign_canvas.canvas.pen.Style := psSoli
d;
1255:
1256: end;
1257:
1258: procedure TForm_homepage.Button5Click(Sender: TObject);
1259: var
1260: coordinates : array[0..7] of TCooridantes;
1261: begin

```

```
1262: circleEightPoints (10,10,10,1000,coordinates);  
1263:  
1264: end;  
1265:  
1266: procedure Tform_homepage.Button6Click(Sender: TObject);  
1267: begin  
1268: innerReflectionCircle(10,10,200,200);  
1269: end;  
1270:  
1271: procedure Tform_homepage.Button7Click(Sender: TObject);  
1272: begin  
1273: miniCircleBorder(10,10,200,200);  
1274: end;  
1275:  
1276: procedure Tform_homepage.Button8Click(Sender: TObject);  
1277: begin  
1278: //skeletonLeaf(100,800,300,100);  
1279: end;  
1280:  
1281: procedure Tform_homepage.CircleClick(Sender: TObject);  
1282: begin  
1283: basicCircle(10,10,200,200)  
1284: end;  
1285:  
1286:  
1287:  
1288:  
1289: end.
```

Bibliography

1. <https://www.kohinoorfoods.co.uk/the-importance-of-henna/#:~:text=In%20the%20ancient%20era%2C%20henna,to%20help%20with%20nervous%20feelings>
2. <https://www.superprof.co.uk>
3. <https://www.amazon.co.uk>
4. <https://apps.apple.com/us/app/henna-mehendi-design-temporary-tattoos/id1161831032s>