Table of Contents

[Introduction 2](#_Toc166615670)

[10 Heuristics 2](#_Toc166615671)

[Visibility of System Status 2](#_Toc166615672)

[Match Between System and Real World 2](#_Toc166615673)

[User Control and Freedom 2](#_Toc166615674)

[Consistency and Standards 2](#_Toc166615675)

[Error Prevention 3](#_Toc166615676)

[Recognition Rather Than Recall 3](#_Toc166615677)

[Flexibility and Efficiency of Use 3](#_Toc166615678)

[Aesthetic and Minimalist Design 3](#_Toc166615679)

[Recognize, Diagnose and Recover from Errors 3](#_Toc166615680)

[Help and Documentation 4](#_Toc166615681)

[User Guide 4](#_Toc166615682)

[Design 4](#_Toc166615683)

[Project Design Wireframe 5](#_Toc166615684)

[Final Product 5](#_Toc166615685)

[Interaction Style 6](#_Toc166615686)

[GOMS 6](#_Toc166615687)

[Method 1 - Adding to Shopping List 6](#_Toc166615688)

[Method 2 – Clear Shopping List 6](#_Toc166615689)

[Method 3 – Close Program 6](#_Toc166615690)

[Goms Evaluation 7](#_Toc166615691)

[Cognitive Walkthrough 7](#_Toc166615692)

[Usability Testing 7](#_Toc166615693)

[Blackbox Testing 8](#_Toc166615694)

[Tester #1 – James MacKenzie 8](#_Toc166615695)

[Tester #2 – Jayden Robertson 9](#_Toc166615696)

[Tester #3 – Liam Ramsay 10](#_Toc166615697)

[Usability Testing Conclusion 11](#_Toc166615698)

# Introduction

This evaluation will be built around my application List-Off! A simple shopping List Application built in Windows Forms C#. This Evaluation will go over the 10 Heuristics and how they relate to my application design along with an evaluation on the overall usability of my application.

# 10 Heuristics

The 10 Heuristics of Usability is a neatly created set of good practice guidelines that were created by Jakob Nielsen. They are applied to an application, system or product and though of behind the end user, how they may use the system or product, how to make it as effective as possible to allow for users to have the upmost best experience.

## Visibility of System Status

Within the end product of the List-Off Application, using a very minimalist design the application has little room or need for system status, the user already knows what is going on within the application due to how quickly the system operates however, the system is built on windows forms which includes the built in mouse cursor spinning wheel when something is processing in the background of an application or system, this icon will appear for a nanosecond a most. I deem the system to be built well for how fast and efficient it is when inputting data or clearing the list.

## Match Between System and Real World

This guideline focuses on what the user will see within text in the application or product, always ensuring that the developer is using real world jargon that the user will know and understand. Within my application I’ve made use of this, keeping all forms of text, error codes and overall input requests as simple as possible for all users to understand when using the application.

## User Control and Freedom

When using an application, you always want a clear exit out of it, allowing the user to close the application if they feel they have done something incorrect or just want to leave the application, having that clear exit will guide them. In my application, I make use of the built-in windows forms app buttons within the open application window. My application blocks the use of maximising the window or changing its dimensions to allow for the text and boxes to appear as intended on screen. If a user were to close the application all they would have to do is click the built-in windows “X” button and the application will close.

## Consistency and Standards

Always ensuring that your design is the same throughout an application is crucial not only for looks but to minimise user confusion when using your system you created. If you have a big application that changes its design for every page, your user will become frustrated, lost and confused which can deter them from your application. My application uses the same design throughout the whole system. Same colour scheme and layout which supports easy learnability and a design that the user will remember for the next time they want to use the system.

## Error Prevention

It is important when designing an application or system to try and minimise the room for error, ensuring that if an error were to occur, to try and make it as simple as possible with a clear intention to the end user of what went wrong and how to correct it. My application has little room for error, other than input errors that have been programmed to appear if the user were to attempt to enter the wrong character input into their shopping List. These errors are simple for the user to understand such as too many characters entered, Try Again. Prompting the user to learn from the mistakes will support the application and the users knowledge on how to use it.

## Recognition Rather Than Recall

When designing the layout of an application, you should always try and make it as simple as possible. Allowing the user to use the application freely without them needing to fully remember the complicated layout of an application. This also helps when having clear buttons or menus within the application, keeping them close by will aid the users experience. My application’s design is minimal allowing for the user to know where everything is due to each part of the application being labelled to what its purpose is.

## Flexibility and Efficiency of Use

Pushing off from the guideline above, shortcuts are always a good option for the end user, it can allow them if more experienced to complete tasks quicker and allows for the less experienced end users to complete tasks using the slower built in buttons within the system. Giving end users freedom of use on the application will not only increase productivity but boost the user gap. My application doesn’t have many shortcuts to it rather than the built-in windows shortcuts. The experienced user can use tab to swap to other content boxes and click enter to use them. It can slowly decrease the time of use for tasks to be completed, they don’t have to use this and are encouraged to use the mouse to navigate the application however it’s an option, nonetheless.

## Aesthetic and Minimalist Design

When designing an application, you should always consider what content will be shown on app or system. Always keep in mind that keeping useless information that are not needed or not related to the application in any way isn’t a good thing to have to an application. Keeping all content on an app or system that relates to the app will keep user confusion minimal, keeping the design minimal is also a factor to this, a simple design is the best design for the end user. My application design is clean and doesn’t have any content not related to and or needed for the application to function. The Design offers a list, and 2 buttons with a label. That is all that is needed for the functionality to the application therefore that is all that is added.

## Recognize, Diagnose and Recover from Errors

If an error were to occur within a system, you should always keep the error codes in a jargon the user will understand and know what to do if faced with it. If a user is scared when an error appears it will frustrate the user into them not using the application. An error code should be short, understandable and have a clear reasoning on why it happened and how to move on from it. My applications error codes are clear for the user to understand, allowing them to learn from it and prevent them from occurring again.

## Help and Documentation

When it comes to a well-designed system, if done correctly there shouldn’t be any need on documentation/guides on how to use the application/system at hand, but in most cases its always good to have as you don’t truly know who will be using your application. My application in question is designed so lightly and minimal to a point where it is super easy to understand the use of the application although I also have created a User Guide that will show the end user a simple step by step guide in how to use the application. This will hopefully remove any situation where the user is feeling lost from not knowing some information on the use of the end product.

### User Guide

A screenshot of a computer screen

Description automatically generated

# Design

My application has gone through a complete overhaul when comparing it to the initial application wireframe in the design stage. This was due to the initial wireframe was based on a bigger project scope, the application was planned to hold more features that ended up being too big for the small-scale application. The Project was dialled down to a state that was feasible for the project deadline. Find the Initial Project Wireframe compared to the end result:

### Project Design Wireframe

A screenshot of a computer

Description automatically generated

### Final Product

A screenshot of a shopping list

Description automatically generated

# Interaction Style

## GOMS

I Will be using a GOMS (Goals, Operators, Methods, and Selection) Calculation to determine how long it would take for the end user to use the application Via a select task, each step of a task is assigned a letter to which has a time to be calculated from the end result of filling it out from the GOMS Rules.

The calculation is below:

### Method 1 - Adding to Shopping List

Move Mouse to “Add” Button (P)

Click Left Mouse Button (K)

Move Mouse to text box (P)

Click Left Mouse Button (K)

Type on Keyboard to add an item to the shopping List (K)

PKPKK

Rule 0 - - PKMPKMKM

Rule 1 - - PKPKK

1.1 + 0.28 + 1.1 + 0.28 + 0.28 = 3.04 Seconds

### Method 2 – Clear Shopping List

Move Mouse to “Clear” Button (P)

Click Left Mouse Button (K)

Move Mouse to “Ok” Button (P)

Click Left Mouse Button (K)

PKPK

Rule 0 - - MPMKMPMK

Rule 1 - - MPKMPK

1.35 + 1.1 + 0.28 + 1.35 + 1.1 + 0.28 = 5.46 Seconds

### Method 3 – Close Program

Move Mouse to “X” Button (P)

Click Left Mouse Button (K)

PK

Rule 0 - - PMK

Rule 1 - - PK

1.1 + 0.28 = 1.38 Seconds

## Goms Evaluation

Overall I am pleased with how quick a user can use the application based on the time given to me from the calculation due to if an application in this case a list would require a lot of time to create it may frustrate users and deter them from the application to use other services.

# Cognitive Walkthrough

This Section will cover the overall walkthrough on using the List-Off Application, from start to finish:

1. Start List-Off! Application
2. Click the “Add” Button
3. Enter data to input box to add items to the shopping List
4. Click “Ok” to submit item or click “Cancel” to not add said item to the shopping List.
5. If an item is picked up, click the check box next to said item within the list to mark it as collected/complete.
6. Click “Clear” to remove all items from the shopping list if everything on the list is collected.

This is a very short and simple application that anyone can use, it doesn’t take much knowledge of technology to use therefore is open to a wider audience.

# Usability Testing

Using Microsoft Forms I have created a short list of tasks for individuals I have to test my application. This is using a testing strategy known as Black Box Testing, giving the program to someone who doesn’t have knowledge of the system at hand. The Form contains small tasks for them to complete with a scale of 1 – 5 1 being easy and 5 being extremely difficult to perform. Find Below an image of the Form Questions:

A screenshot of a online survey

Description automatically generated

## Blackbox Testing

### Tester #1 – James MacKenzie

Reviewing my First Tester’s Form Response, James has completed all the tasks provided on the form for the application with no problems, totalling all questions with Very Easy Rating. James completed all tasks which adds to the flexibility and ease of use for the application.

##### Feedback on what can be done better

James mentions the possible functionality of a Remove button that will allow the user to remove certain items on a list at their own leisure. This would be a great idea to implement to the application project in the future.

###### James MacKenzie’s Form Response

A screenshot of a computer

Description automatically generated

### Tester #2 – Jayden Robertson

Based on my Second Tester’s Form Response, Jayden has managed to complete all provided tasks within my application with ease, each task was given Very Easy as a difficulty rating which I’m delighted with.

##### Feedback on what can be done better

Jayden given some feedback on a possible feature to be added to the application if more time were added to the development, he believes that adding functionality for multiple lists to be created and edited would be a great asset to the application. To this I am inclined to agree, giving the application more functionality and purpose will not only give the application more meaning but open it to a wider audience allowing users to create any kind of list they want to.

###### Jayden Robertson’s Form Response

A screenshot of a shopping list

Description automatically generated

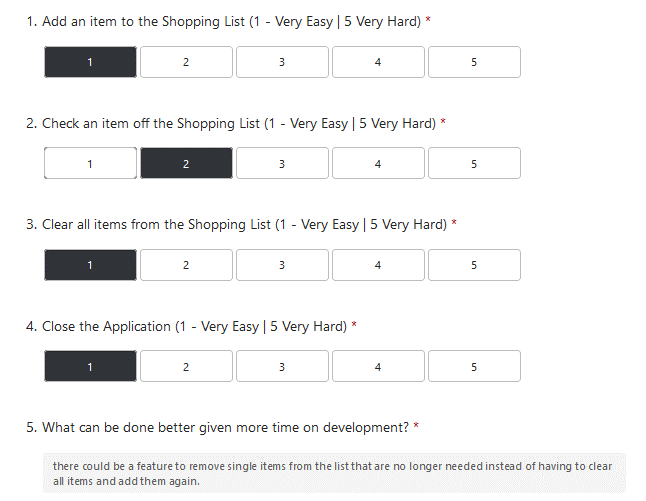
### Tester #3 – Liam Ramsay

Based on my final tester’s Form Response, I am relatively happy with it. It shows that my application is easy and simple to use, fast and efficient. Using the feedback attached from this from will help me improve areas such as checking an item off the list.

##### Feedback on what can be done better

Just like James, Liam stated that a feature that consists of allowing for singular items to be removed from the shopping List could possibly be added to the application, this would prove useful if the end user doesn’t want to fully clear their list and start again if they don’t want an item on there. Given more time on development I feel would be a great feature to which I didn’t consider at time of development.

###### Liam Ramsay’s Form Response



## Usability Testing Conclusion

In the end, I am happy with the end state and my feedback given to me for my small-scale application via Microsoft forms. Given more time on development I would like to return and implement some of these features mentioned in the feedback such as the ability of removing specific items within a shopping list rather than all list content. I would also like to expand the application to add functionality for multiple lists at once, this was the proposed purpose at the start of the project which was cut down due to time constraints and feasibility.