Dominik Subocz

Small Scale Apps / Testing Project

Coin Flip App

Overview

Contents

[Purpose of Test Strategy 2](#_Toc152334537)

[Objective 2](#_Toc152334538)

[Test Scope 2](#_Toc152334539)

[Test Strategy 2](#_Toc152334540)

[White-box testing 2](#_Toc152334541)

[Black-box testing 2](#_Toc152334542)

[Functional Testing 3](#_Toc152334543)

[Unit Testing 3](#_Toc152334544)

[System Testing 3](#_Toc152334545)

[Tests & Results 4](#_Toc152334546)

[Unit Testing 4](#_Toc152334547)

[System Testing 7](#_Toc152334548)

[Functionality Testing 7](#_Toc152334549)

[App in Action 8](#_Toc152334550)

[Overview of the user interface design 8](#_Toc152334551)

[Design principles 8](#_Toc152334552)

[Navigation and menu structure 8](#_Toc152334553)

[Visual elements 8](#_Toc152334554)

[Designs 9](#_Toc152334555)

[Flip Mode (MainPage) 9](#_Toc152334556)

[Menu (SplitView) 9](#_Toc152334557)

[Guess Mode (GuessFlip) 9](#_Toc152334558)

[Interaction Design 10](#_Toc152334559)

[Class Diagram 11](#_Toc152334560)

[Site Map 11](#_Toc152334561)

[Trello Boards (Progression) 12](#_Toc152334562)

# Purpose of Test Strategy

The test strategy document serves as a comprehensive guide to evaluate the functionality, usability, and performance of the Coin Flip application.

The document provides a detailed test plan, covering various aspects such as navigation, menu clarity, error handling and overall functionality. It aims to provide a structured approach to testing, ensuring that the application meets user expectations, runs smoothly, and deals effectively with potential problems.

This document only contains the test plan, the actual tests are documented and located in a separate folder in the documents folder.

Through a systematic testing strategy, this document aims to ensure a robust and user-friendly experience for those using the Coin Flip app.

# Objective

The main objective of this test plan is to test and ensure the functionality and reliability of the Coin Flip app so that it works correctly, and the user has a flawless experience.

# Test Scope

The tests will cover the basic functionality of the application, including coin flipping, guessing mode and user interface and responsiveness.

The tests will also cover the scoring system to check that there are no logic errors and that the score is correct.

The app uses no input validation and yet no user errors can be made. The error prevention is done by setting minimum and maximum values, disabling buttons until method finished running, combo boxes not having empty entries, users cannot make any errors, no matter how hard they try.

# Test Strategy

### White-box testing

The testing strategy for the Coin Flip App mostly focus on white-box testing. In this testing approach, the tester has knowledge of the application’s architecture, scrutinizing individual lines of code, methods, and pathways. By understanding the intricacies of software’s logic and structure, the main objective is to uncover any hidden bugs, logical errors, and other vulnerabilities. This process allows us to ensure that each component performs as intended and that the different parts of the system interact seamlessly.

### Black-box testing

This involves seeking feedback from friends, fellow students, and lecturers who engage with the application without delving into its underlying code. Their insights provide valuable external viewpoints, helping evaluate the user experience, functionality, and overall usability of the application. This dual approach ensures a comprehensive evaluation, combining the meticulous examination of code internals with real-world user perspectives.

The test strategy will include the following types of testing:

## Functional Testing

This test will focus on verifying that the coin toss button works as expected. Confirming that the application displays the correct output and correctly increments the heads and tails score.

The functionality of the menu and navigation will also be tested, as well as the function of the guess mode; to check that the correct message and sound is played when guessing.

Functional testing is crucial to ensure that the core functions of the Coin Flip app work as intended. This includes verifying the operation of the coin flip button, confirming the accurate display of results, and ensuring the correct score increment for heads and tails. Functional testing also includes menu navigation and correct execution of the Guess Mode function, ensuring a reliable and accurate user experience.

## Unit Testing

The unit testing phase focuses on the systematic evaluation of individual components and functions of the Coin Flip application.

This includes rigorous testing of core functions such as the coin flip mechanism, score calculation and data handling.

Unit tests verify that each method and class work as expected, ensuring accurate results during coin flips, correct recording of historical data and seamless integration of audio and visual elements.

Unit testing focuses on isolating and evaluating individual components and functions within the Coin Flip application. This rigorous testing approach ensures that each method and class work as expected, contributing to the overall stability and quality of the application. Unit tests will cover core functionality, including the coin flip mechanism, score calculation, data handling and audiovisual integration, addressing potential bugs and edge cases.

## System Testing

System testing for our application involves evaluating the entire integrated system to ensure that it works as intended and meets the specified requirements. This comprehensive testing phase examines the interactions between the various components and modules, checking for correct data flow, user interface responsiveness and overall system behaviour.

The aim is to verify that the software runs smoothly in the intended environment, adapting to different inputs and usage scenarios. System testing goes beyond individual functions, focusing on how different functions work together and identifying potential issues that may arise in a real-world environment.

Through rigorous testing protocols, the aim is to detect and fix any faults, ensuring the robustness and reliability of our application before it reaches end users. This phase is crucial to guarantee a smooth user experience and overall system stability.

# Tests & Results

## Unit Testing

The unit tests in Coin Flip app might show different outcomes at times because some features, like coin flips, involve a bit of randomness. These tests can pass or fail based on the luck of the draw during each run. It's a bit like rolling a dice, it could land on different numbers each time.

While this randomness mimics real-life situations, it makes the tests a bit unpredictable.

This time only 3 out of 4 tests passed, the test failed because randomly generated result wasn’t right with the guess criteria.

A screenshot of a computer

Description automatically generated

This time both guesses were incorrect.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

This time however, all the tests have passed.

A screenshot of a computer

Description automatically generated

This is the code for each test:

A screenshot of a computer

Description automatically generated

A black rectangle with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

VideoMaster Unit tests (testing if correct video is playing)

A screen shot of a computer

Description automatically generated

Code for VideoMaster tests (each test is identical except filepath is different)

A screenshot of a computer

Description automatically generated

## System Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subject to Test** | **Scenario** | **Test Type** | **Expected Result** | **Actual Result** | **Comments** |
| Coin flip button | User spam smashes coin flip button | Extreme | The application becomes unresponsive or slow. | Button is disabled on click until flip is finished |  |
| FlipCoin() Method | Clicking the "Flip Coin" button with standard parameters | Normal | Flip performs and the right video plays. | Flip performs and the right video plays. |  |
| GuessFlip()Method | User guesses the flip, they chose Heads | Normal | Random result is generated, user either guessed or not. | Random result was generated, user either guessed or not. | The pop up showed up before the actual method finished |
| Duration slider | User tries to drag duration slider above 2 | Exceptional | Video not displayed because it wasn't found. | Application prevented user from sliding the slider above 2, slider's maximum value is 2. |  |
| Coin type combo box | User tries to pick something else than gold, silver and bronze | Exceptional | User won't be able because only coins are gold, silver and bronze | User won't be able because only coins are gold, silver and bronze | Option to add custom coins might be implemented in future versions |
| Volume Slider | User sets the volume lower than 10 | Normal | Volume will decrease | Volume decreases |  |
| Volume Slider | User tries to set volume below 0 | Exceptional | Error pops up as it cannot be negative | User can't set volume lower than 0, minimum value is 0 |  |
| VideoMaster() Class | User chooses gold coin | Normal | Coin switches to gold, video is updated before flip. | Coin switches to gold, video is updated before flip. |  |
| VideoMaster() Class | User chooses silver coin | Normal | Coin switches to silver, video is updated before flip. | Coin switches to silver, video is updated before flip. |  |
| VideoMaster() Class | User chooses bronze coin | Normal | Coin switches to bronze, video is updated before flip. | Coin switches to bronze, video is updated before flip. |  |
| Guess flip button | User spam smashes guess flip button | Exceptional | The application becomes unresponsive or slow. | Button is disabled on click until flip is finished |  |
| Menu button | User spam smashes the menu button | Extreme | Menu opens and closes, no performance issues | Menu opens and closes, no performance issues |  |
| GuessFlipPage Button | User clicks the guess flip page button | Normal | User is redirected to guessFlip page, their previous results are transferred | User is redirected to guessFlip page, their previous results are transferred |  |
| CoinFlipPage Button | User clicks the coin flip page button | Normal | User is redirected to coinFlip page, their previous results are transferred | User is redirected to coinFlip page, their previous results are transferred |  |

## Functionality Testing

# App in Action

The videos of application running can be found under the videos folder. This folder contains two .mp4 video files of the Coin Flip App running. Flip-mode.mp4 shows the default flip mode in action, showcasing the app’s customization features. GuessMode.mp4 shows the app’s guess mode in action.

# Overview of the user interface design

The Coin Flip’s UI design isn’t very complex, it also isn’t too professional, it probably has some faults here and there, but I think it’s not too bad. While it may look quite bad, it’s quite easy to use.

This section provides a comprehensive overview of the design principles, layout and interactive elements that shape the visual identity of the application.

# Design principles

The user interface design prioritises a visually appealing, minimalist, and user-friendly experience.

Although some design guidelines were considered during development, the focus is on achieving a modern, elegant look that is consistent with the concept of simplicity in the application.

The design aims for an intuitive layout, focusing on clarity and efficiency, meeting the needs of users who appreciate a clean and simple aesthetic.

The final user interface reflects the deliberate choice of a distinctive and stylish design, moving away from strict adherence to specific rules or design guidelines.

# Navigation and menu structure

A clear and concise navigation structure has been implemented to facilitate the exploration of the application's functions. Intuitive menu layout enhances user accessibility, ensuring that users can effortlessly navigate between different sections, including coin flip, guess mode and history tracking.

# Visual elements

The visual elements have been carefully designed to reflect the identity of the Coin Flip app. A consistent colour palette, well-chosen icons and visually appealing graphics contribute to an aesthetically pleasing interface. The app's branding is seamlessly integrated, creating a recognisable and memorable user experience.

# 

# Designs

### Flip Mode (MainPage)

A screenshot of a game

Description automatically generated

### Menu (SplitView)

A screenshot of a phone

Description automatically generated

### Guess Mode (GuessFlip)

A screenshot of a game

Description automatically generated

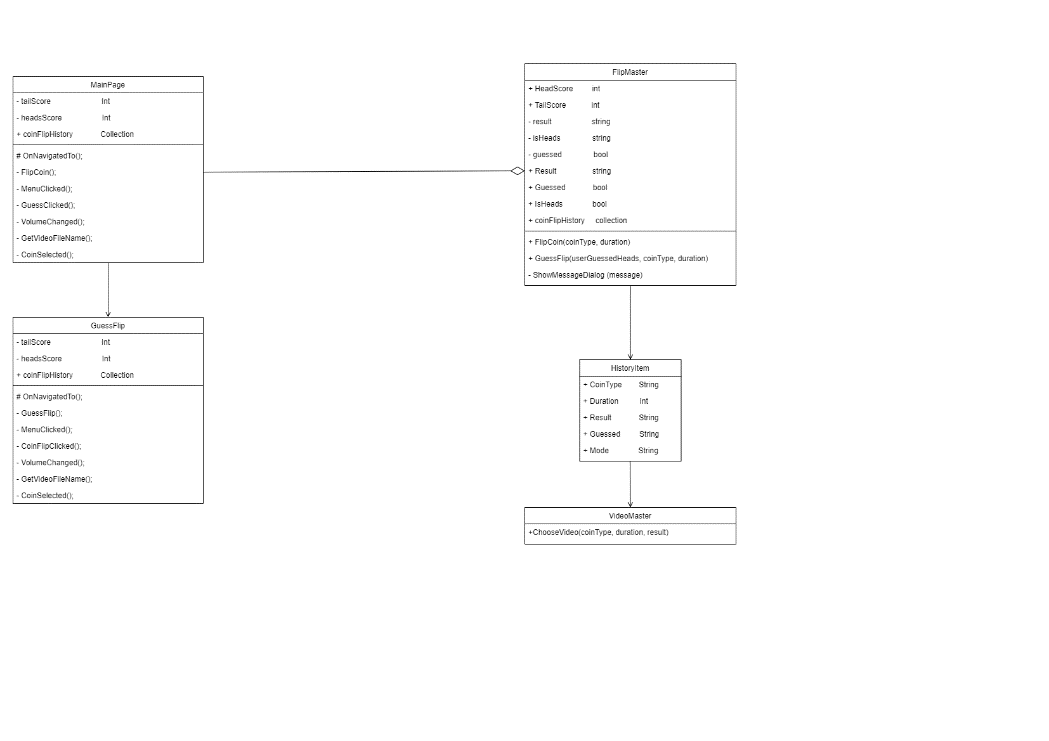
# Interaction Design

A diagram of a diagram

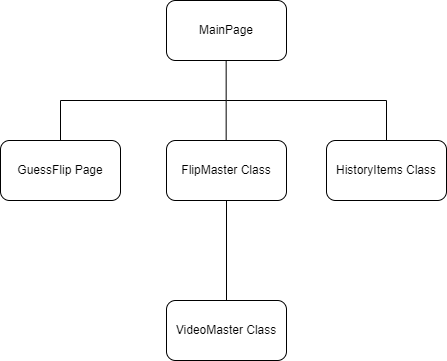
Description automatically generated

This similar looking to a site map design shows the interactions between different elements of the app. Highlighted in red is the text that gives a rough idea of what will be going on behind the visuals in the app, in the code. Arrows point to different parts of the app; the dashed arrow is just there to keep things less clustered and easy to read.

## Class Diagram



## Site Map



## Trello Boards (Progression)

A screenshot of a computer

Description automatically generated 6th November 2023

20th November 2023

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated 24th November 2023