

Team Three: Test Plan

Michele Cook, Chelsea Nieves, Valerie Rudich, Constance Sturm, and Hoi Lam Wong

University of Maryland Global Campus

CMSC 495-7382: Capstone in Computer Science

Professor David Castillo

December 12, 2023

Team Three: Test Plan

Table of Contents

1. Introduction
 - 1.1. Purpose
 - 1.2. Scope
 - 1.3. Objectives
 - 1.4. Approach
2. Test Items
 - 2.1. Features to be Tested
 - 2.2. Features not to be Tested
3. Test Deliverables
 - 3.1. Test Types
 - 3.2. Test Cases
 - 3.3. Test Data
 - 3.4. Test Reports
4. Test Responsibilities
 - 4.1. Testing Team
 - 4.2. Development Team
5. Defect Management
 - 5.1. Defect Logging
 - 5.2. Defect Resolution
6. Approval Criteria
 - 6.1. Test Sign-off

Team Three: Test Plan

Introduction

1.1 Purpose

The purpose of the test plan is to ensure that each part of the application works correctly with user interaction. The test plan will test each menu option, including the rules of the fortune teller game, registering a user, the option to sign the user in and out, the option to choose a fortune from a desired category, and to exit the game. The fortune option will then be further tested with the different fortune categories and a loop for picking another fortune, saving a fortune, or to exit back to the main menu. The application will be tested for the ability to read .txt files to generate the display for the fortunes.

1.2 Scope

The features to be tested include:

- GUI interface for ease of use.
- The menu bar will display the rules or exit the application.
- The welcome menu with the options to register, login, or play as a guest.
- The user menu with the options to get fortune or view saved fortunes.
- The portion of application where fortune choice is made will be tested to ensure proper fortune is shown.
- After the fortune is revealed the user will have the option to choose another fortune from the menu, save the fortune (if the user is signed in), or exit to the menu screen.
- The ability for logged in users to view saved fortunes.
- The Random Generator will be tested multiple times to ensure that the fortunes are displayed randomly.
- The application will be tested to ensure that .txt files are being read

1.3 Objectives

The objective of testing the application is to ensure that the user's interaction with the application works properly. The user can make several decisions during the use of the application and each part needs to be tested for correctness and usability. The application will also be tested for ease of use, likability, the login and registration forms, the use of the database, and for the reading of .txt files that hold the different fortunes.

1.4 Approach

Testing will occur with each code deliverable prior to the end of deployment to ensure potential bugs/vulnerabilities are caught and mitigated as early as possible in the development cycle.

2. Test Items

2.1 Features to be Tested

The predominant features to be tested are the GUI interface which are used for the interactive portion of the application. A database function that will include user information and saved fortunes. A random number generator will also be created and tested for functionality and randomness. This number generator will be used to select the fortune category in instances where the user selects "Random" from the Fortune Menu. Each menu option will be tested for correctness based upon its category. For example, if the user selects to return to the main menu, the program should return them to the welcome screen. The exit function will be tested to ensure the program ceases to run. Input validation functions will be tested to ensure that no potentially malicious user input is accepted. The ability to read .txt files will also be tested to ensure the proper fortunes are displayed.

2.2 Features not to be Tested

Error logging, welcome message.

3. Test Deliverables

3.1 Test Types

Test types: Manual

- Welcome Menu:
 - **Login/Register:** Will be tested for validation and connection to the database.
 - **Choose a fortune:** Will be tested to ensure that this option takes the user to the next menu to choose a fortune.
- Fortune Menu:
 - Tested to prompt the user to choose another fortune, save the fortune (if logged in), or exit back to the menu.
 - **Love:** Will be tested to ensure a fortune about love is displayed.
 - **Career:** Will be tested to ensure a fortune about the user's career is displayed.
 - **Health:** Will be tested to ensure a fortune about the user's health is displayed.
 - **General:** Will be tested to ensure a fortune about general life is displayed to the user.
 - **Random:** Will be tested to ensure the random generator is working properly and displays a random fortune to the user.
- User Menu
 - **Choose a fortune:** Will be tested to ensure that this option takes the user to the next menu to choose a fortune.
 - **View Saved Fortunes:** Will be tested to ensure previous saved fortunes are displayed to the user.
- Menu Bar
 - **View Rules:** Will be tested to ensure rules are displayed and the user is returned to the menu screen once closed.

- **Sign Out:** Will be tested to ensure the user is signed out and returned to the welcome menu.
- **Exit the program:** Will be tested to ensure that the program is closed properly.
 - Will also test that the program is closed when the user exits through the title bar in both the welcome and user menu.

3.2 Test Cases

Actual Test Results:  CMSC 495 - Actual Test Cases

The following table shows what one overall test may look like to cover all aspects of the application:

<i>Test Case</i>	<i>Description of Test</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/Fail</i>
1: Main Menu					
1.1: - Rules					
1.1.1	Menu Bar: User wishes to view “Rules” of Zoltar game	User Input: Selects rules from the menu bar.	User is displayed the rules of the fortune teller game.		
1.1.2	User chooses to exit rules	User Input: Selects “Close” button	User is returned to main menu		
1.1.3		User closes rules from title bar	User is returned to main menu		
1.2: - Register					
1.2.1		User Input: Clicks “Register” button	User is displayed registration form		
1.2.2	User chooses to exit registration	User Input: Selects “Close” button	User is returned to main menu		
1.2.3		User closes registration	User is returned to		

		from title bar	main menu		
1.3: - Login					
1.3.1		User Input: Clicks “Login” button	User is displayed login form		
1.3.2	User chooses to exit login	User Input: Selects “Cancel” button	User is returned to main menu		
1.3.3		User closes login from title bar	User is returned to main menu		
1.4: - Play as Guest					
1.4.1		User Input: Selects “Play as Guest” button	Fortune menu is displayed. Users are prompted to click another button from the selection		
1.4.1	User chooses to exit fortune	User closes fortune and fortune menu from title bar	User is returned to main menu		
1.5: - Exit Program					
1.5.1		User input: Selects “Exit” from the menu bar.	Application exits		
1.5.2		User closes program from title bar	Application exits		
2: Fortune Menu					

2.1	Option 1 - Love	User Input: Clicks “Love” button	User is displayed a fortune about love and prompted to choose another fortune.		
2.1.1	User chooses another fortune	User Input: Clicks “New Fortune” and selects another fortune category	Fortune menu is displayed. User is prompted to click another button from the selection.		
2.1.2	User chooses to exit	User closes fortune menu from title bar	Main menu is displayed.		
2.2	Option 2 - Career	User Input: Clicks “Career” button	User is displayed a fortune for career. Users are prompted to choose another fortune.		
2.2.1	User chooses another fortune	User Input: Clicks “New Fortune” and selects another fortune category	Fortune menu is displayed. User is prompted to click another button from the selection.		
2.2.2	User chooses to exit	User closes fortune menu from title bar	Main menu displayed.		
2.3	Option 3 - Health	User Input: Clicks “Health” button	User is displayed a fortune for health. Users are prompted to choose another fortune.		

2.3.1	User chooses another fortune	User Input: Clicks “New Fortune” and selects another fortune category	Fortune menu is displayed. User is prompted to click another button from the selection.		
2.3.2	User chooses to exit	User closes fortune menu from title bar	Main menu displayed.		
2.4	Option 4: General Life	User Input: Clicks “General” button	User is displayed a fortune for general life. Users are prompted to choose another fortune.		
2.4.1	User chooses another fortune	User Input: Clicks “New Fortune” and selects another fortune category	Fortune menu is displayed. User is prompted to click another button from the selection.		
2.4.2	User chooses to exit	User closes fortune menu from title bar	Main menu displayed.		
2.5	Option 5: Random	User Input: Clicks “Random” button	User is displayed a random fortune. Users are prompted to choose another fortune.		
2.5.1	User chooses another random fortune	User Input: Clicks “New Fortune” and selects “Random” button	User is shown another fortune from a random category. The user is prompted to choose another fortune.		

2.5.3	User chooses to exit	User closes fortune menu from title bar	User is displayed the main menu and prompted to click another button for selection.		
3: User Menu					
3.1	User selects fortune	User Input: Clicks “Get a Fortune” button	User is displayed the fortune menu and prompted to click another button for selection.		
3.1.2		User Input: Clicks any button	User is shown a fortune from the chosen category. User is prompted to choose another fortune or save fortune.		
3.1.3		User Input: Clicks “Save Fortune” button	Fortune saved.		
3.1.4	User chooses to exit	User closes fortune screen and fortune menu from title bar	User menu is displayed and user is prompted to make another selection.		
3.2		User Input: Clicks “View Past Fortunes” button	The user is shown a list of saved fortunes.		
3.3 - Sign Out		User Input: Clicks “Sign Out” from menu bar	User is returned to main menu.		
4: Input Validation Validate inputs the user enters for login/registration page.					

4.1: Invalid Input					
4.1.1	User attempts to register account	User Input: Selects "Register" button			
Username					
4.1.1.1 Empty Input / Null		User Input:	Error Displayed: "Invalid Username" User directed back to registration screen.		
4.1.1.2 Already Registered		User Input: test	Error Displayed: "Username Unavailable" User directed back to registration screen.		
4.1.1.3 Outside of Acceptable Range		User Input: qwerrtyuiopas dfg hjklzxcvbn m	Error Displayed: "Invalid Username" User directed back to registration screen.		
3.3 Test Data		Data required for testing will include a list of fortunes to be displayed to the user. Additionally, user input will be required for login and registration forms. A test will be considered a "pass" if the output is, as expected. For example, if the user selects the menu option to view the rules, the rules should be displayed. A test will be considered a failure if the output is not as expected, such as, if the menu were to allow a command or string outside of the expected range as input and crash the program. Additionally, another example of a failure would be if the user were to select the "Love" category for a fortune but a "Career" fortune was output instead.			
4.1.1.4 Special char		User Input: #	Error Displayed: "Invalid Username" User directed back to registration screen.		
First Name					
3.4 Test Reports					
4.1.1.5 Empty Input / Null		User Input:	Error Displayed: "Invalid First Name" User		
Testing and the scope of the test report will be conducted based on different scenarios that will be described and then conducted to see if the results match the expected results. The frequency of testing and					

reporting will be dependent on the different stages of development. When a new menu option is developed it should be thoroughly tested before further development happens.

4.0 Test Responsibilities

4.1 Testing Team

Chelsea Nieves, Constance Sturm, Valerie Rudich, Michele Cook

4.2 Development Team

Constance Sturm, Chelsea Nieves, Hoi Lam Wong, Valerie Rudich

5.0 Defect Management

5.1 Defect Logging

Defects will be well documented on what part of the application has the defects, what scenario causes the defect to occur, and what part of the code is causing the defect. Each defect will be reproducible, and needed details will be logged into a shared google sheet to-do list based on priority level. Included details will be: Date, Task (Defect Description), Assignee, Status, Date Updated, and Date Completed. Team will regularly check the to-do list, and updates will be mentioned in the team discord channel.

5.2 Defect Resolution

Once a bug or defect has been identified, the code in question will be revised or reformatted to achieve the desired results. Code review and PyUnit testing of the new code will be conducted to ensure that the mitigated program runs without error. Developers will update defect tracker with status and information on fixes. Testers will execute the original test case(s) associated with the defect to verify the fix. If the defect is not fixed or if new issues are identified, the cycle repeats. When resolved, the team

will ensure that the resolution did not introduce new issues by testing program functionality to check for unintended side effects, broken functionalities, or performance degradation.

6.0 Approval Criteria

6.1 Test Sign-off

The test completion and sign-off is dependent on the success of the application without bugs or errors present. The test will have to prove that each portion of the application is working as expected and displays the corresponding fortune for each category.