Team Members:
-Leeya Howley rlh9398
-Ons Taktak ot503
-Jiacheng Xia jx2467
-Mirette Dahab md5341

Project Deliverable 4: Testing Report

Section 1: Introduction

1.1 Test Project Name: Specify the name of the test project.
ClearMyMind

1.2. Summary of the Rest of the Test Plan:
We compiled all the test cases we could perform (such as testing inputs and behavior for a journal entry, prescription creating, and pill tracking as part of our mental health website, database functions,...).
We divided our test cases based on categories (including functionality, performance, security, …). We also grouped them based on their main features. We created a traceability matrix to make sure all of our relevant requirements (to the 3 use cases) have been tested.
Finally, we executed all the test cases and inspected the results over multiple rounds resolving any issues that did not pass the testing in each round.

Section 2: Feature Description

Describe the features of the selected application.
- A secure account for a user that other users cannot access
- A diary space where user can see all their journal entries
- Space for user to create journal entries with title, main text, and date
- Space for user to monitor and keep track of their available medications with name, expiry date, and number of pills, as well as add new prescriptions
- Displays alerting the user if a medication has expired
- Space for user to log their intakes of active medications for current or past dates
- Space for user to check their intake history log (which is a displayed list of the medication name they took, number of pills taken and the date of intake)

Section 3: Assumptions

3.1 Test Case Exclusions: List any test cases or scenarios that are excluded from testing.
- Wifi failures, connection failures
- Not testing if entered text in the entries is actual text vs gibberish
- Not testing for the existence of sensitive words (such as suicide, …)
- Load testing/stress testing (because our web app has not been deployed yet: we can only test for 1 user at a time)
- Interoperability testing: our system does not interact with other systems

- Reliability testing: we won't measure the ability of our system to keep operating for a long time without developing failures. Not applicable because our system is not deployed and we cannot measure its performance over a long period of time.

3.2 Test Tools, Formats, and Organizational Schemes: List the testing tools, formats, and organizational schemes used.
- We used Google Sheets to
    - Stay organized with our test cases, keep track of which ones passed and which ones failed
    - Keep track of the bugs resulting from failed test cases
    - Categorize our test cases
    - Create our traceability matrix

Section 4: Test Approach
4.1 Special Testing Considerations: Highlight any special considerations for testing
- Our web app is not a deployed product at the moment. You have to have a local database on your computer to run the system.

4.2 Test Strategy: Testing techniques used and the rationale for the selection.

Testing techniques	Rationale behind using the technique
Black Box - Boundary Value (Robustness Testing)	It allowed us to test inputs that are in the form of a range, for any different value the user could input
Black Box - Equivalence Class Partitioning	Classes allow us to test for all the different possible input values the user can attempt to put in
Black Box- Decision Table	It allowed us to make sure that any rules we set for the functionality of the website work based on conditions and appropriate response actions
White Box - Control Flow Graph	It allowed us to make sure that all paths of the while loop in a section of our code are identified, feasible and executed correctly
White Box - Data Flow Testing	It allowed us to identify if we had unnecessary variable definitions which makes our code more efficient
System Testing - Performance	This was an appropriate technique to test the speed of some functionalities of our system
System testing - Functionality Testing - Security	This allowed to us to make sure that our system's security features are well implented
System testing - Functionality Testing - Graphical User Interface Tests	This allowed us to make sure that the system is accessible and easily understood by users and that our design choices were well-received
System Testing - Regression	It allowed us to make sure that fixing our identified bugs did not create more bugs
Exploratory Testing - FedEx Tour	This was an appropriate technique to test for data addition, modifying and deleting within a database
Exploratory Testing - After Hours	We used this because we had a functionality that updates an information in the database, based on the current date at a time when the user is not interacting with the system

4.3 Test Categories: Categorize the test cases, e.g., functional, performance, security, etc.

Number	Test Case	Category	
1	Conditions on Logging in	Functional	Security
2	Conditions on Signing up	Functional	Security
3	Attempting Website Access	Security	
4	Conditions on Journal Entry Title	Functional	
5	Conditions on Journal Entry Date	Functional	
6	Conditions on Journal Entry Content	Functional	
7	Cancelling Journal Entry Confirmation	Functional	
8	Conditions on Prescription Name	Functional	
9	Conditions on Prescription Expiry Date	Functional	
10	Conditions on Prescription's Number of pills	Functional	
11	Conditions on Intake Log chosen prescription	Functional	
12	Conditions on Intake Log Date	Functional	
13	Conditions on Intake's number of pills taken	Functional	
14	Displaying journal entries in correct order	Functional	Interface
15	Measuring Performance of Large Journal Table in The Database	Performance	
16	Scanning for Data Anomalies	Performance	
17	Checking for Real Time Expiry Status Message	Functional	
18	External Feedback on GUI	Usability	Interface
19	Checking Other Component Functionalities after Fixing a Bug	Functional	
20	Checking Database Functionalities	Logical Database	
21	Verifying Password is Hashed Correctly	Security	

Section 5: Test Cases

5.1 Test Group and Subgroup Definition:

Number	Test Case	Group	Subgroup	Group Objective
1	Conditions on Logging in	Access to Website	Login	Ensure account access is restricted to appropriate corresponding users with valid credentials
2	Conditions on Signing up		Signup	
3	Attempting Website Access		Login	
4	Conditions on Journal Entry Title	Diary	Title for Submitting Diary	Consider all possible inputs for creating a diary entry and handle each situation appropriately
5	Conditions on Journal Entry Date		Date for Submitting Diary	
6	Conditions on Journal Entry Content		Content for Submitting Diary	
7	Cancelling Journal Entry Confirmation		Cancelation	
8	Conditions on Prescription Name	Prescription	Name for Submitting Prescription	Consider all possible inputs for creating a prescription and handle each situation appropriately
9	Conditions on Prescription Expiry Date		Date for Submitting Prescription	
10	Conditions on Prescription's Number of pills		Num Pills for Submitting Prescription	
11	Conditions on Intake Log chosen prescription	Intake	Presc choice for Submitting Intake	Consider all possible inputs for creating an intake log and handle each situation appropriately
12	Conditions on Intake Log Date		Date for Submitting Intake	
13	Conditions on Intake's number of pills taken		Num Pills for Submitting Intake	
14	Displaying journal entries in correct order	Database	Comprehensive Display	Ensure all database functionalities and displays are working as intended for different operations
15	Measuring Performance of Large Journal Table in The Database		Performance	
20	Checking Database Functionalities		Functionalities	
21	Verifying Password is Hashed Correctly		Hashing	
16	Scanning for Data Anomalies	Coding	Efficient Variables	Ensure code is efficient
17	Checking for Real Time Expiry Status Message	Miscellaneous	Hands off	Ensure everything else not related to a specific other group is working as intended
18	External Feedback on GUI		User Interface	
19	Checking Other Component Functionalities after Fixing a Bug		Regression	

5.2 Test Cases:
Link to all test cases:
https://docs.google.com/document/d/1xHWAKf6j6TvjTCw3yt9VmNP37Wul_VmgWz8MWvsAsEg/edit?usp=sharing

5.3 Traceability Matrix:

Requirement	Test case																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1. System shall allow users to create an account using their name, date of birth, gender, email address and a password.		x																			
1.1. User must provide a valid email address		x																			
1.2. User must set a password containing at least 8 characters, including at least one uppercase letter and one number		x																			
4. The system shall provide a dedicated diary space for the user to submit a journal entry.				x	x	x	x														
8. System shall allow users to log their medical prescriptions in the pill tracking interface.								x	x	x											
8.1. System shall allow users to add the expiry date, the total amount of pills procured, and the name of the prescription								x	x	x							x				
9. System shall allow users to log when they take their medication in the pill tracking interface.											x	x	x								
9.1 System shall allow users to choose a prescription, the amount of pills taken, and the date of intake											x	x	x								
15. The system shall be able to display information from the database under 25 milliseconds.														x	x		x				
16. The system shall be be able to add an entry to the database successfully under 25 milliseconds.															x						
17.The system shall use the font Helvetica as a default to accommodate users with ADHD and the neurodivergent audience.																		x			
18.The system shall be accessible to people with dyslexia by using a dyslexia-friendly sans-serif font.																		x			
19.The system shall have a high-contrast default theme using light gray background and dark colored font to accommodate people with different abilities.																		x			
24.The system shall encrypt user's password to secure their account.																					x
25. The system shall permit access only to authorized users (users with accounts).	x		x																		
Requirements related to the section Logical Database Requirements in our SRS																				x	

Section 6: Test Environment

6.1 Multiple Test Environments:
- Different browsers (Firefox, Safari, Google Chrome)
- Different OS's (Mac, Windows)
- Web vs Mobile

6.2 Schematic Diagram of the test environment setup if applicable.
- N/A

6.3 Test Architecture Overview if Applicable.
- N/A

6.4 Equipment Table:
- N/A

Section 7: Testing Results

Summary of testing results, including passed, failed, and unresolved issues:

Test case	Results					
	Round 1		Round 2		Round 3	
	Status	Failure Reason	Status	Failure Reason	Status	Failure Reason
1	Passed		Passed		Passed	
2	Passed		Passed		Passed	
3	Passed		Passed		Passed	
4	Passed		Passed		Passed	
5	Passed		Passed		Passed	
6	Passed		Passed		Passed	
7	Passed		Passed		Passed	
8	Passed		Passed		Passed	
9	Passed		Passed		Passed	
10	Passed		Passed		Passed	
11	Passed		Passed		Passed	
12	Passed		Passed		Passed	
13	Passed		Passed		Passed	
14	Failed	any submitted form is displayed twice after page refresh	Failed	recently submitted entry shows up at the bottom of the list	Passed	
15	Passed		Passed		Passed	
16	Failed	a "du" anomaly was detected	Passed		Passed	
17	Passed		Passed		Passed	
18	Passed		Passed		Passed	
19	Passed		Passed		Passed	
20	Failed	any submitted form is added twice to the database after page refresh	Passed		Passed	
21	Passed		Passed		Passed	

- Unresolved issue: In the pill management section of the website, when the "Prescriptions" tab is selected out of the two tabs "Intake Log" & "Prescriptions", and you submit the form to add a new prescription, the website automatically takes you back to the default "Intake Log" tab, instead of staying on the "Prescriptions" tab.
  We deemed this issue as not important to fix, especially since fixing it requires the use of languages that we have no experience with and were unable to use correctly despite trying.

Section 8: Recommendations on Software Quality

Recommendations on improving the quality of the software based on testing results
- Fixing the unresolved issue
- Deploying the web app to be able to test for more things such as load testing, stress testing and reliability testing
- Adding communication with external systems (such as weather APIs, notification systems, calendar apps) for extra functionalities to the website as well as being able to test for interoperability