Software Test Plan

Harkify Application

Version 1.0.0

Team Bravo

SWEN 670

Test Plan Approvals

Name	Signature	Date
Approved by:		
Approved by: "Stakeholder"		
Approved by: "Project Manager"		

Revision History

Author	Date	Reason For Changes	Version
Team Bravo	06/07/2021	Initial version	1.0
Roy Gordon	06/11/2021	Recommended mention of saving PII as risk	1.0.1
Team Bravo	06/11/2021	General revisions. Added PII to security requirements.	1.0.2
Team Bravo	06/19/2021	General revisions. Added encrypt documents, notifications, and favorites page to use cases.	1.0.3

Contents

Introduction

Purpose

This document details the plans for test the functional requirements of Harkify, a mobile application designed to assist people with short-term memory issues. The document will comprehensively cover all aspects of the plan for testing Harkify. By formally defining the test plan, team members and informed stakeholders will all have access to the same information on how testing will be conducted and come to a consensus on how testing will be conducted. This document aims to improve the quality and comprehensiveness of Bravo Team's testing and shall elicit feedback from stakeholders and management to improve the process and methods of testing the application.

Test Scope

The test plan will be used to define and facilitate the effective testing of the application's input and output to determine if the application meets both in-scope and out-of-scope requirements.

In-Scope

The tests required for this project will ensure that Bravo Team's Harkify application is functional and meets all requirements set by stakeholders. Harkify is designed with the Flutter framework testing it effectively is achieved by using:

- Unit/Widget Testing
- Integration Testing
- User Acceptance Testing
- Regression Testing

Out-of-Scope

Performance testing, stress testing and system testing are out of scope for this project.

Test Objective

The objectives of testing Harkify are to determine whether the design meets all the requirements of the project. It needs to be determined if users of Harkify will be able to complete the tasks in the application successfully and reliably.

Intended Audience

The intended audience for this document is the members of Bravo team, Project Managers, DevSecOps, and the Product Owner. Relevant stakeholders are also the intended audience. Bravo is committed to synergy between development teams and invite the other teams to critique and take inspiration from this test plan, for the purposes of improving the quality of each team's product.

Roles and Responsibilities

Name	Role	Responsibilities
Raul Benavides	Test Manager	Assess the progress and effectiveness of testing effort. Plans and assigns testing activities.
Tyler Puschinsky Ben Cushing Elshaday Mesfin	Test Analyst	Identify and define the necessary tests. Collection, monitoring, and management of test data in each test cycle. Assess quality of product based on testing activity.
Entire Team	Tester	Executes test cases. Identify and report defects within software. Test resolutions from developers and verify defect no longer exists.
Entire Team	Developer	Will implement resolutions to defects in the software.

Testing Schedule & Key Milestones

- 28 June 2021 - Test Plan Complete
    - This document will be completed and ready for internal review and delivery.
- 28 June 2021 - Test Development Begins
    - Development of automated tests will begin, including Unit, Widget, and Integration tests.
- 5 July 2021 - Continuous Test Execution Begins
    - Execution of automated tests begin as soon as they are written, and continue throughout the project.
- 19 July 2021 - Test Development Complete
    - All automated tests will be completed.
- 26 July 2021 - Test Execution Complete
    - Execution of automated tests is completed, execution of user acceptance tests is completed. Results are collected for the test report.
- 2 Aug 2021 - Test Report Complete

- The Test Report document will be completed and ready for internal review and delivery.

Referenced Documents

Document	Location
Project Plan	https://umuc365-my.sharepoint.com/:w:/g/personal/rbenavides8_student_umgc_edu/Efh0P0BhigJJg4xvM827N1sBF-lEvEnZ_WfwTjJEMnu2sA
Software Requirement Specification	https://umuc365-my.sharepoint.com/:w:/g/personal/rbenavides8_student_umgc_edu/EbYbtJOPQZpHi1g52UInCOcBcIaiI33nrvbwA25gWO9hgA
PM Requirements	Appendix A.
Flutter Unit Testing	https://flutter.dev/docs/cookbook/testing/unit/introduction
Flutter Widget Testing	https://flutter.dev/docs/cookbook/testing/widget/introduction

Test Approach

Unit/Widget Testing

Unit tests will provide the foundation of testing for this project, representing the largest number of tests, as well as the most complete level of code coverage. Each software developers will be responsible for produce unit tests covering the code they develop, with any necessary guidance from the project tester.

Widget tests are UI specific tests supported by Flutter. Portions of code that are not suitable for unit tests, such as the UI, may be covered by Widget tests. Widget tests are expected to ensure that necessary functionality is tested, they are not intended to enforce UI design decisions.

A minimum level of combined test coverage will be required between unit and widget tests, at 60% of lines of code covered by tests.

Integration Testing

The team will develop a suite of automated integration tests to verify the developed application is able to integrate with the local device, or external applications, as required. These tests will cover a larger portion of functionality than Unit and Widget tests, with the goal of ensuring that individually tested units and widgets work correctly together.

User Acceptance Testing

User Acceptance tests are tests performed from the perspective of a user of the application. These tests are focused on ensuring the application meets the requirements outlined in the SRS, and that the application operates according to expectations. The test analysts are responsible for producing a series of acceptance tests to verify the completion of project requirements.

Regression Testing

Tests developed early in the development lifecycle will be executed as they are completed, in order to ensure defects are detected as soon as possible. These tests will be re-executed throughout the duration of development regardless of their previous pass/fail state, in order to ensure that later development does not introduce defects in previously tested portions of the application.

Test Cases

Test Case 1: Start Application

Requirements under test: 3.1.1 Start Application & 3.1.12 Welcome New User

PM Requirement: 19

Application first start

Description:	This test verifies that the system performs all first start functions correctly.
Test Type:	User Acceptance
Pre-conditions:	• The application is in a new install state.

Test Approach:	1. Tester launches the application from the base OS. 2. Tester verifies the application launches. 3. Tester verifies the application begins the process for training voice recognition and saving triggers. 4. Tester completes the following test cases as prompted by the application: 2.1, 3.1, 4.1, 5.1, 6.1 5. Tester verifies they were guided through each listed test case by the application.

Application first start aborted

Description:	This test verifies that the system completes all first start functions, if the original first start is aborted.
Test Type:	User Acceptance
Pre-conditions:	• The application is new install state.
Test Approach:	1. Tester launches the application from the base OS. 2. Tester verifies the application launches. 3. Tester verifies the application begins the process for training voice recognition and saving triggers. 4. Tester begins the following test cases as prompted by the application: 2.1, 3.1, 4.1, 5.1; but DOES NOT complete all of them. 5. Tester closes the application. 6. Tester opens the application. 7. Tester verifies the application launches. 8. Tester verifies the application begins the process for training voice recognition and saving triggers. 9. Tester completes the following test cases as prompted by the application: 2.1, 3.1, 4.1, 5.1, 6.1 (Note: The application may skip cases that were completed during step 4) 10. Tester verifies they were guided through each listed test case by the application.

Application non-first start

Description:	This test verifies that the system does not launch the welcome user process after the first start.
Test Type:	User Acceptance

Pre-conditions:	• The application has previously been started, and the welcome user process has been completed.
Test Approach:	1. Tester launches the application from the base OS. 2. Tester verifies the application launches. 3. Tester verifies the application does not begin the process for training voice recognition and saving triggers.

Application purges old notes on start

Description:	This test verifies that the application removes notes that have been persisted longer than the configured persistence duration.
Test Type:	User Acceptance
Pre-conditions:	• The application has previously been started, and the welcome user process has been completed.
Test Approach:	1. Tester launches the application from the base OS. 2. Tester saves a note according to test case 6.1 3. Tester updates the note persistence setting to 0 minutes, according to test case 8.? 4. Tester stops the application. 5. Tester launches the application. 6. Tester confirms the previously saved note has been deleted.

Test Case 2: Train Voice Recognition

Requirement under test: 3.1.2 Train Voice Recognition

PM Requirement: 12

Successful training attempt

Description:	This test verifies that a user is able to train voice recognition within the application.
Test Type:	User Acceptance
Pre-conditions:	• "View Settings" screen is open.

Test Approach:	1. Tester selects the “Train Voice Recognition Option” 2. Tester verifies the system requests a series of phrases be repeated. 3. Tester repeats the series of phrases **correctly**. 4. Tester verifies a confirmation of successful training is displayed.

Error in first training attempt

Description:	This test verifies that a user is able to train voice recognition if the first attempt at training fails.
Test Type:	User Acceptance
Pre-conditions:	• “View Settings” screen is open.
Test Approach:	1. Tester selects the “Train Voice Recognition Option” 2. Tester verifies the system requests a series of phrases be repeated. 3. Tester repeats the series of phrases **incorrectly**. 4. Tester verifies the system requests the series of phrases be repeated again. 5. Tester repeats the series of phrases **correctly.** 6. Tester verifies a confirmation of successful training is displayed.

Error in repeated training attempts

Description:	This test verifies that a user is able to train voice recognition if any number of training attempts fail.
Test Type:	User Acceptance
Pre-conditions:	• “View Settings” screen is open.
Test Approach:	1. Tester selects the “Train Voice Recognition Option” 2. Tester verifies the system requests a series of phrases be repeated. 3. Tester repeats the series of phrases **incorrectly**. 4. Tester verifies the system requests the series of phrases be repeated again. 5. Tester repeats steps 3 & 4 a number of times to verify the system will keep attempting training. 6. Tester repeats the series of phrases **correctly.** 7. Tester verifies a confirmation of successful training is displayed.

Training data is saved

Description:	This test verifies that training data is saved to the device after training has been completed.
Test Type:	Integration
Pre-conditions:	• Voice recognition training has been successfully completed.
Test Approach:	1. Tester verifies training data is saved to the device.

Test Case 3: Record Save Trigger

Requirement under test: 3.1.3 Record Save Trigger

PM Requirements: 9, 10

Successful save trigger recording

Description:	This test verifies that a user is able to save a recording trigger within the application.
Test Type:	User Acceptance
Pre-conditions:	• "View Settings" screen is open.
Test Approach:	1. Tester selects the "Record Save Trigger Option" 2. Tester verifies the system requests a trigger phrase be spoken. 3. Tester speaks a trigger phrase **correctly**. 4. Tester verifies a confirmation of successful recording is displayed.

Error in first save trigger recording attempt

Description:	This test verifies that a user is able to save a recording trigger if the first attempt at saving fails.
Test Type:	User Acceptance
Pre-conditions:	• "View Settings" screen is open.

Test Approach:	1. Tester selects the “Record Save Trigger Option” 2. Tester verifies the system requests a trigger phrase be repeated. 3. Tester does NOT repeat a trigger phrase. 4. Tester verifies the system again requests a trigger phrase be spoken. 5. Tester speaks a trigger phrase **correctly**. 6. Tester verifies a confirmation of successful recording is displayed.

Save trigger data is saved

Description:	This test verifies that save trigger data is saved to the device after recording has been completed.
Test Type:	Integration
Pre-conditions:	• Save trigger recording has been successfully completed.
Test Approach:	1. Tester verifies save trigger data is saved to the device.

Test Case 4: Record End Trigger

Requirement under test: 3.1.4 Record End Trigger

PM Requirement: 9, 10, 20

Successful end trigger recording

Description:	This test verifies that a user is able to save an end recording trigger within the application.
Test Type:	User Acceptance
Pre-conditions:	• “View Settings” screen is open.
Test Approach:	1. Tester selects the “Record End Trigger Option” 2. Tester verifies the system requests a trigger phrase be spoken. 3. Tester speaks a trigger phrase **correctly**. 4. Tester verifies a confirmation of successful recording is displayed.

Error in first end trigger recording attempt

Description:	This test verifies that a user is able to save an end recording trigger if the first attempt at saving fails.
Test Type:	User Acceptance
Pre-conditions:	• "View Settings" screen is open.
Test Approach:	1. Tester selects the "Record End Trigger Option" 2. Tester verifies the system requests a trigger phrase be repeated. 3. Tester does NOT repeat a trigger phrase. 4. Tester verifies the system again requests a trigger phrase be spoken. 5. Tester speaks a trigger phrase **correctly**. 6. Tester verifies a confirmation of successful recording is displayed.

End trigger data is saved

Description:	This test verifies that end trigger data is saved to the device after recording has been completed.
Test Type:	Integration
Pre-conditions:	• End trigger recording has been successfully completed.
Test Approach:	1. Tester verifies end trigger data is saved to the device.

Test Case 5: Record Search Trigger

Requirement under test: 3.1.5 Record Search Trigger

PM Requirement: 9, 10

Successful search trigger recording

Description:	This test verifies that a user is able to save a search trigger within the application.
Test Type:	User Acceptance
Pre-conditions:	• "View Settings" screen is open.

Test Approach:	1. Tester selects the “Record Search Trigger Option” 2. Tester verifies the system requests a trigger phrase be spoken. 3. Tester speaks a trigger phrase **correctly**. 4. Tester verifies a confirmation of successful recording is displayed.

Error in first search trigger recording attempt

Description:	This test verifies that a user is able to save a search trigger if the first attempt at saving fails.
Test Type:	User Acceptance
Pre-conditions:	• “View Settings” screen is open.
Test Approach:	1. Tester selects the “Record Search Trigger Option” 2. Tester verifies the system requests a trigger phrase be repeated. 3. Tester does NOT repeat a trigger phrase. 4. Tester verifies the system again requests a trigger phrase be spoken. 5. Tester speaks a trigger phrase **correctly**. 6. Tester verifies a confirmation of successful recording is displayed.

Search trigger data is saved

Description:	This test verifies that search trigger data is saved to the device after recording has been completed.
Test Type:	Integration
Pre-conditions:	• Search trigger recording has been successfully completed.
Test Approach:	1. Tester verifies search trigger data is saved to the device.

Test Case 6: Save Note

Requirement under test: 3.1.6 Save Note

PM Requirements: 1, 8, 11, 13, 14, 16

Successfully Saving a Note

Description:	This test verifies that a user is able to save a text note through vocal commands.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, a Save trigger has been recorded, and an End trigger has been recorded.
Test Approach:	1. Tester speaks the Save trigger followed by the sentence, "The quick brown fox jumps over the lazy dog 1 2 3 4 5 6 7 8 9 0." Then Tester speaks the End trigger. 2. Tester verifies that the system shows a confirmation that the text note has been saved. 3. Tester verifies the system shows this sentence as the only content of a newly saved text note: "The quick brown fox jumps over the lazy dog 1 2 3 4 5 6 7 8 9 0."

Saving a Note Without an End Trigger

Description:	This test verifies that a user is able to save a text note through vocal commands without supplying an End trigger.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, and a Save trigger has been recorded.
Test Approach:	1. Tester speaks the Save trigger followed by the sentence, "This text note is for testing purposes only." Then Tester is quiet for the number of seconds specified in the application settings to trigger the end of note saving. 2. Tester verifies that the system shows a confirmation that the text note has been saved. 3. Tester verifies the system shows this sentence as the only content of a newly saved text note: "This text note is for testing purposes only."

Error in Saving a Note

Description:	This test verifies that an error occurs when the user's speech cannot be interpreted while saving a text note.
Test Type:	User Acceptance

Pre-conditions:	• Application is open, tester's voice has been trained, a Save trigger has been recorded, and an End trigger has been recorded.
Test Approach:	1. Tester speaks the Save trigger followed by speech which is muffled by the microphone being covered with a thick blanket. 2. Tester verifies that the system shows an error message that the text note could not be saved. 3. Tester verifies the system has not saved a new text note.

No Note to Save

Description:	This test verifies that no note is saved if the user immediately says the End trigger after saying the Save trigger.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, a Save trigger has been recorded, and an End trigger has been recorded.
Test Approach:	1. Tester speaks the Save trigger. Then Tester immediately speaks the End trigger. 2. Tester verifies that the system does not save a new text note.

Successfully Saving a Long Note

Description:	This test verifies that a user is able to save a long text note through vocal commands.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, a Save trigger has been recorded, and an End trigger has been recorded.

Test Approach:	1. Tester speaks the Save trigger followed by this text: Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere. 2. Tester speaks the End trigger. 3. Tester verifies that the system shows a confirmation that the text note has been saved. 4. Tester verifies the system shows the correct content of a newly saved text note.

Test Case 7: View Note

Requirement under test: 3.1.7 View Note

PM Requirement: 2

Successfully Viewing a Note

Description:	This test verifies that the user can view a text note after it has been saved.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Notes” option.
Test Approach:	1. Tester selects a text note from the View Notes list. 2. Tester verifies that the system shows the full content of the selected note on the screen. 3. Tester verifies that the system also shows Edit and Delete options for that note.

Editing a Note

Description:	This test verifies that the user can edit an existing text note.
Test Type:	User Acceptance

Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.
Test Approach:	1. Tester selects the Edit option. 2. Tester verifies that the system shows the full content of the selected note in an editable format. 3. Tester verifies that the system shows Save and Cancel options. 4. Tester adds the word “TEST” at the beginning of the text note and again at the end of the text note. 5. Tester selects the Save option. 6. Tester verifies that the system shows the View Note screen again with “TEST” at the beginning of the text note and again at the end of the text note.

Deleting a Note

Description:	This test verifies that the user can delete an existing text note.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.
Test Approach:	1. Tester selects the Delete option. 2. Tester verifies that the system asks for a confirmation that the note should be deleted. 3. Tester selects Yes. 4. Tester verifies that the system shows a confirmation that the note has been deleted. 5. Tester verifies that the note has been deleted from the View Notes list.

Abandoning Edits to a Note

Description:	This test verifies that the user can abandon edits to an existing text note.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.

Test Approach:	1. Tester selects the Edit option. 2. Tester verifies that the system shows the full content of the selected note in an editable format. 3. Tester verifies that the system shows Save and Cancel options. 4. Tester adds the word “TEST” at the beginning of the text note and again at the end of the text note. 5. Tester selects the Cancel option. 6. Tester verifies that the system shows the View Note screen again without “TEST” at the beginning of the text note or at the end of the text note. 7. Tester selects the View Notes screen. 8. Tester verifies that the system does not show “TEST” at the beginning of the text note from the View Notes screen.

Abandoning Deletion of a Note

Description:	This test verifies that the user can abandon the deletion of an existing text note.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.
Test Approach:	1. Tester selects the Delete option. 2. Tester verifies that the system asks for a confirmation that the note should be deleted. 3. Tester selects No. 4. Tester verifies that the system shows the View Note screen with the complete text note and options for Edit and Delete. 5. Tester selects the View Notes screen. 6. Tester verifies that the note has not been deleted from the View Notes list of notes.

Error in Editing a Note

Description:	This test verifies that an error occurs when the user edits an existing text note when the application does not have permission.
Test Type:	User Acceptance

Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.
Test Approach:	1. Tester selects the Edit option. 2. Tester verifies that the system shows the full content of the selected note in an editable format. 3. Tester verifies that the system shows Save and Cancel options. 4. Tester adds the word “TEST” at the beginning of the text note and again at the end of the text note. 5. Tester changes application permissions for Harkify so that it no longer has access to documents. 6. Tester selects the Save option. 7. Tester verifies that the system shows an error message indicating that the operation could not be completed.

Error in Deleting a Note

Description:	This test verifies that an error occurs when the user deletes an existing text note when the application does not have permission.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the “View Note” option on the text note.
Test Approach:	1. Tester selects the Delete option. 2. Tester verifies that the system asks for a confirmation that the note should be deleted. 3. Tester changes application permissions for Harkify so that it no longer has access to documents. 4. Tester selects Yes. 5. Tester verifies that the system shows an error message indicating that the operation could not be completed.

Test Case 8: View Settings

Requirement under test: 3.1.8 View Settings

PM Requirement: 19

Successfully Viewing Application Settings

Description:	This test verifies that the user can view current application settings.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the “View Settings” option.
Test Approach:	1. Tester verifies that the system shows the text note purge setting and the end save note setting on the screen. 2. Tester verifies that the system also shows the following options: Edit, Train Voice Recognition, Record Save Trigger, Record End Trigger, Record Search Trigger.

Changing Note Purge Setting

Description:	This test verifies that the user can edit the note purge application settings.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the “View Settings” option and then the Edit option.
Test Approach:	1. Tester verifies that the system shows the text note purge setting and the end save note setting on the screen in an editable format with Save and Cancel options. 2. Tester changes the text note purge setting to 1 day. 3. Tester selects the Save option. 4. Tester verifies that the system shows the changed note purge setting value. 5. Tester verifies that the system still shows the original end save note setting value.

Changing End Save Note Setting

Description:	This test verifies that the user can edit the end save note application settings.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the “View Settings” option and then the Edit option.

Test Approach:	1. Tester verifies that the system shows the text note purge setting and the end save note setting on the screen in an editable format with Save and Cancel options. 2. Tester changes the end save note setting to 1 second. 3. Tester selects the Save option. 4. Tester verifies that the system shows the changed end save note setting value. 5. Tester verifies that the system still shows the original note purge setting value.

Abandoning Changes to Settings

Description:	This test verifies that the user can abandon changes to the application settings.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the "View Settings" option and then the Edit option.
Test Approach:	1. Tester changes the text note purge setting to 1 day and changes the end save note setting to 1 second. 2. Tester selects the Cancel option. 3. Tester verifies that the system still shows the original application setting values.

Error in Changing Note Purge Setting

Description:	This test verifies that an error occurs when the user edits the note purge setting when the application does not have permission.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the "View Settings" option and then the Edit option.

Test Approach:	1. Tester selects the Edit option. 2. Tester verifies that the system shows the note purge setting value in an editable format. 3. Tester verifies that the system shows Save and Cancel options. 4. Tester updates the note purge setting to 99 days. 5. Tester changes application permissions for Harkify so that it no longer has access to documents. 6. Tester selects the Save option. 7. Tester verifies that the system shows an error message indicating that the operation could not be completed.

Error in Changing End Save Note Setting

Description:	This test verifies that an error occurs when the user edits the end save note setting when the application does not have permission.
Test Type:	User Acceptance
Pre-conditions:	• Application is open and the tester selects the "View Settings" option and then the Edit option.
Test Approach:	1. Tester selects the Edit option. 2. Tester verifies that the system shows the end save note setting in an editable format. 3. Tester verifies that the system shows Save and Cancel options. 4. Tester updates the end save note setting to 99 seconds. 5. Tester changes application permissions for Harkify so that it no longer has access to documents. 6. Tester selects the Save option. 7. Tester verifies that the system shows an error message indicating that the operation could not be completed.

Test Case 9: Search Notes By Voice

Requirement under test: 3.1.9 Search Notes By Voice

PM Requirements: 5, 17, 18

Successfully Searching for a Note

Description:	This test verifies that a user is able to search for a text note through vocal commands.

Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, a Search trigger has been recorded, and test case 6.1 has been executed.
Test Approach:	1. Tester speaks the Search trigger followed by, "brown fox." 2. Tester verifies that the system speaks the following text note contents, "The quick brown fox jumps over the lazy dog 1 2 3 4 5 6 7 8 9 0" through the phone's speaker.

Searching With No Results

Description:	This test verifies that a user sees no results when searching for a non-existent text note through vocal commands.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained and a Search trigger has been recorded.
Test Approach:	1. Tester speaks the Search trigger followed by, "twas brillig and the slithy toves." 2. Tester verifies that the system speaks through the phone's speaker an indication that no text notes were found that matched that phrase.

Searching Without a Filter

Description:	This test verifies that a user does not trigger a text note search through vocal commands when no search phrase is given.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained and a Search trigger has been recorded.
Test Approach:	1. Tester speaks the Search trigger followed by a number of seconds of silence beyond the end save note application setting. 2. Tester verifies that the system does not speak through the phone's speaker or show any search results on the screen.

Error in Searching for a Note

Description:	This test verifies that an error occurs when the user's speech cannot be interpreted while searching for a text note.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, tester's voice has been trained, a Search trigger has been recorded, and test case 6.1 has been executed.
Test Approach:	1. Tester speaks the Search trigger followed by "brown fox" while muffling the phone's microphone with a thick blanket. 2. Tester verifies that the system speaks through the phone's speaker to indicate that an error has occurred while trying to interpret the user's speech. 3. Tester verifies that no text note search has occurred.

Test Case 10: View Notes

Requirement under test: 3.1.10 View Notes

Successfully Viewing All Notes

Description:	This test verifies that the user can view text note excerpts after they have been saved.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, a text note was previously saved, and the tester selects the "View Notes" option.
Test Approach:	1. Tester verifies that the system shows the first 100 characters of all saved notes, the date/time saved for each note, and whether the text note is favorited for each note.

Successfully Viewing No Notes

Description:	This test verifies that the user shall see no text notes when none have been saved.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, no text notes were previously saved, and the tester selects the "View Notes" option.

Test Approach:	1. Tester verifies that the system shows a message that no text notes were found.

Successfully Searching for a Note

Description:	This test verifies that a user is able to search for a text note through the View Notes page search bar.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, test case 6.1 has been executed, and the View Notes option has been selected.
Test Approach:	1. Tester selects the search bar button, types in "brown fox," and then selects the Search button. 2. Tester verifies that the system shows the following text note as a matching result, "The quick brown fox jumps over the lazy dog 1 2 3 4 5 6 7 8 9 0". 3. Tester verifies that the first 100 characters of that text note show, that the date/time the note was recorded shows, and that the correct favorite status shows. Tester also verifies that the text include includes a View Note option.

Searching With No Results

Description:	This test verifies that a user sees no results when searching for a non-existent text note through the View Notes page search bar.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, test case 6.1 has been executed, and the View Notes option has been selected.
Test Approach:	1. Tester selects the search bar button, types in "twas brillig and the slithy toves," and then selects the Search button. 2. Tester verifies that the system shows an indication that no text notes were found that matched that phrase and that the user should search again.

Searching Without a Filter

Description:	This test verifies that a user does not trigger a text note search through the View Notes page search bar when no search phrase is given.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, test case 6.1 has been executed, and the View Notes option has been selected.
Test Approach:	1. Tester selects the search bar button, types in no text, and then selects the Search button. 2. Tester verifies that the system shows the full list of current text notes without filtering them through a search.

Viewing More Than a Page of Notes

Description:	This test verifies that the user can view more than a page of text note excerpts after they have been saved.
Test Type:	User Acceptance
Pre-conditions:	• Application is open, 10 text notes were previously saved, and the tester selects the “View Notes” option.
Test Approach:	1. Tester verifies that the system shows the first 100 characters of all saved notes, the date/time saved for each note, and whether the text note is favorited for each note. 2. Tester verifies that the system allows them to scroll to see text notes that do not fit on the initial screen.

Test Case 11: Stop Application

Requirement under test: 3.1.11 Stop Application

Successfully Stop application

Description:	This test verifies that a user is able to stop/end the session of the open application and the system performs all stop functionalities as expected.
Test Type:	User Acceptance

Pre-conditions:	• User has the application installed and currently trained to specific user • User is ready to end the session. • The system has checked and retained any unsaved changes.
Test Approach:	1. Tester clicks on the stop icon from the main menu. 2. Tester verifies that the system has saved any previous changes since the current secession. 3. Tester verifies the system automatically then closes the application. 4. Tester verifies the system only starts back up when the recognized voice trigger is heard or manually launched by the user with the intent of starting the application.

Test Case 12: Encrypt Documents

Requirement under test: 3.1.13 Encrypt Documents

PM Requirement: 22

Successfully Encrypt documents

Description:	This test verifies the user's PII provided to the application by the user is secure. Since there is no transfer of data, the only encryption needed is data at rest on the local storage of the device.
Test Type:	Security testing, Integration testing
Pre-conditions:	• User must have begun recording a note. • User selects to save the note to the application • System has enabled file-based encryption.
Test Approach:	1. Tester initiates the *Record Save Trigger* and verifies the notes are saved. 2. Tester verifies that the system has enabled the file based encryption on the device. https://source.android.com/security/encryption/file-based 1. Tester verifies implementation of the feature, run available encryption tests 2. Tester verifies the system does not display any PII in plain text.

Test Case 13: Favorite Pages

Requirement under test: 3.1.14 Favorite Pages

Successfully save the note to Favorite Page

Description:	This test verifies that a user is able to store notes that can not be deleted directly from the application in the one week period and saves the marked note in the *Favorites* page/screen.
Test Type:	User Acceptance
Pre-conditions:	• The user must have previously saved notes. • The notes screen is open.
Test Approach:	1. Tester selects the note screen 2. Tester verifies that the system allows the user to select a specific note from the available list. 3. Tester marks the note as favorite. 4. Tester verifies the system logs the note to the favorite page.

Test Case 14: Notifications

Requirement under test: 3.1.15 Notifications

PM Requirement: 23

Successful Notifications

Description:	This test verifies the application is able to remind the user of the application and its recording abilities, by using various notification methods for the specified device.
Test Type:	User Acceptance
Pre-conditions:	• The user has not used the application. • User has not used that start trigger phrase. • No system functionality of the application for the past 12 hours.

Test Approach:	1. Tester makes sure the application is stopped and is not being used or updated information for the past 12 hours. 2. Tester verifies that the system has sent a notification and is displayed in forms of different banners to the user's phone. 3. Tester verifies the system notification displayed in the banner is informational and suitable for the application. 4. Tester verifies the system allows for accepting or dismissing the notification on the user's device.

Test Case 15: Training Videos

Requirement under test: 3.1.16 Training Videos

PM Requirement: 7

Successfully view Training Videos

Description:	This test verifies that a user is able to use the Training Videos feature available in the *Setting* page. The *Training Videos* section demonstrates how to use the application.
Test Type:	User Acceptance
Pre-conditions:	• The application is currently open and the user is under the *Setting* page/screen.
Test Approach:	1. Tester verifies clicking on the *Training Videos* section of the setting page is functional. 2. Tester verifies that the system retrieves the training videos for the application. 3. Tester verifies the videos are organized in order of start to stop or going through the application. 4. Tester verifies the system plays the selected video from the available list. 5. Tester verifies the system allows switching between previous and next videos available. 6. Tester verifies the user is able to exit the video and go to the videos listing page or exit to the home screen of the application.

Testing Criteria

Test Initiation

Testing will be executed periodically throughout development of the Harkify project. Test execution will include new test cases when the following criteria are met:

- The test case has been fully written.
- Testers, Developers, and Business Analysts agree the test case is accurate according to the system requirements.
- The developer for the requirement under test reports the feature is complete.
- If the test is automated, the automated test is complete.

Test Completeness

The following conditions are required to be met to consider testing complete:

- All test cases have been developed, covering the requirements of the system.
- All test cases have been executed, and results recorded.
- All discovered defects have been prioritized.
- Blocking defects have been fixed and re-tested.

Test Deliverables

The following test documents will be delivered as part of Harkify project development:

Test Plan

This test plan document provides the overview of testing that will be completed during Harkify development. The test plan will include:

- The scope of testing, including what is in and and out of scope.
- The test approach, describing what types of testing will be executed.
- Detailed test cases providing step by step procedures for all documented requirements.

Test Report

The test report document will provide an overview of the results of testing the Harkify application under development. The report will include:

- Results of test executions, including which test cases passed and which test cases failed.
- A summary of defects discovered during testing, and the current status of each defect at the time the report is produced.

Resource & Environment Needs

Testing Tools

The “test” library provided by the Dart language will be used to perform unit testing.

The “flutter_test” library provided by the Flutter framework will be used to perform widget testing.

An Android emulator provided by Android Studio will be used to perform user acceptance testing.

Test Environment

The test environment has the following minimum system requirements, based on Android Studio:

- 64-bit Microsoft® Windows® 8/10 OR MacOS® 10.14 (Mojave) or higher OR Any 64-bit Linux distribution that supports Gnome, KDE, or Unity DE; GNU C Library (glibc) 2.31 or later.
- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor OR ARM-based chips, or 2nd generation Intel Core or newer with support for Hypervisor.Framework
- 8 GB RAM or more
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution

Appendix A: PM Requirements

This table outlines the requirements provided by the overall PM.

Requirement	Mandatory/Optional	Criterion
1.) The application shall not save any voice recording.	Mandatory	Does the application save any voice recording to storage?
2.) The application shall allow the user to edit any speech converted to text.	Mandatory	Does the application allow the user to edit any speech converted to text?
3.) The application shall provide the option to convert the user’s speech to text.	Mandatory	Does the application provide the option to convert the user's speech to text?
4.) The application shall provide an user interface that incorporates the following device features: Bold Text, Display Zoom, the ability to Increase Text Size, and High Contrast Colors. (Optional)	Optional	

5.) The application should read the text and play the synthesized speech into the user's earbuds.	Mandatory	Does the application read the text accurately (what percentage) and play the speech?
6.) The application shall keep the end user persona in mind for UI/UX considerations and offer a flexible environment for the user to customize.	Mandatory	When evaluated by the customer and focus groups, do they validate that the end user was kept in mind for UI/UX considerations? Can the user customize the UI environment?
7.) The application shall provide training videos within the app to guide the user regarding its various features and functionalities.	Mandatory	Does the application provide training videos within the app? Do the video sufficiently guide the user regarding various features and functionalities?
8.) The application shall recognize distinct phrases and sentences that he or she uses while speaking to him or herself or with others.	Mandatory	Does the application recognize distince phrases and sentences the user utilizes during speech?
9.) The application shall learn the phrases that the user provides when talking to someone while trying to save important spoken text and also the phrases that the user wants to use speaking to him or herself trying to retrieve the noted information.	Mandatory	Does the application learn the phrases the user desires?
10.) The application shall provide a facility to the user. This facility will note several phrases that the user may use in natural conversation to save as well as to retrieve saved information to play as speech.	Mandatory	Does the application provide the facility for the user? Does the facility note phrases that the user utilizes in natural conversation?
11.) The application shall ignore everything except what the user speaks.	Mandatory	Does the application sufficiently ignore background noise and other noises?

12.) The application shall provide the means for the user to train the app on its voice.	Mandatory	Does the application provide the ability for the user to train the app on the device?
13.) The application shall bypass asking everyone permission to record.	Mandatory	Does the application bypass asking everyone permission to record?
14.) The application shall provide the ability to save the speech to conversion texts to local device storage.	Mandatory	Does the application provide the ability to save the speech to conversion texts to local device storage?
15.) (Optional) – The Mobile Operating System's Virtual Assistant shall interface directly with the Application by Application Name	Optional	
16.) The application shall provide the following means to activate recording: Tap on the app and immediate voice recognition.	Mandatory	Does the application provide a means to activate recording?
17.) The application shall provide the ability to search through the saved speech to text notes via text field and/or voice command.	Mandatory	Professor states: Yes. The search could be keyword based. It could also be conversation or time based. For instance, the user may be interested in knowing all recorded notes during the present or last conversation.
18.) The application shall retrieve all results related to the search command.	Mandatory	Professor states: Well, kind of. If a user has been using the app for a couple of years, he or she may not be interested in results from a year ago. Find some way to make it relevant and provide an appropriate user experience keeping the persona in mind. I suggest conferring people who may be prime users. The team may want to visit a nearby senior home. Does the application retrieve all results related to the search command?
19.) The application shall retain speech to text recognition notes for 1 week in duration.	Mandatory	Does the application work with the phone to provide a variable in the setting section to accomplish this?

20.) The application shall provide a trigger to end the voice recording.	Mandatory	Does the application provide the trigger to end the voice recording?
21.) The Application must work with IOS and Android	Mandatory	Does the application work on both IOS and Android platforms?
22.) The Application must encrypt the data at rest.	Mandatory	Does the application verifiably encrypt data at rest?
23.) The application shall provide a notification banner as a daily event presented to the user (That the application is available and should get used)	Mandatory	Does the application provide a proper notification banner?

Appendix B: Test Execution Template

Test Case ID		Test Title/Description				
Pre-Requisite		Post Requisite				
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1						
2						
3						