



CogniOpen Software Application

Test Report

Project Name: CogniOpen Software Application
Document: Test Report (TR)
Version: 1.0
Date: November 7, 2023
Prepared By: Team A

Team Lead / Project Manager:
Vincent Galeano
Dream Team Technology Solutions LLC (DTTS)
vgaleano1@student.umgc.edu

Client:
Dr. Mir Assadullah
University of Maryland Global Campus (UMGC)
mir.assadullah@faculty.umgc.edu

Leads Sign-off Sheet

| Key Reviewer | Version | Date | Signature |
|--|---------|-------------|------------------------------|
| Vincent Galeano Team Lead / Project Manger | 1.0 | 07/Nov/2023 | <i>Vincent Galeano</i> |
| Kavon Johnson Lead Technical Writer | 1.0 | 07/Nov/2023 | <i>Kavon Johnson</i> |
| Zach Bowman Lead Business Analyst | 1.0 | 07/Nov/2023 | <i>Zach Bowman</i> |
| David Bright Architect / Lead Software Developer | 1.0 | 07/Nov/2023 | <i>David Bright</i> |
| Juan Torres-Chardon Lead UI/UX Designer | 1.0 | 07/Nov/2023 | <i>Juan Torres - Chardon</i> |
| Laura Hamann Lead Test Engineer | 1.0 | 07/Nov/2023 | <i>Laura Hamann</i> |

Revision History

| Date | Version | Description | Author |
|-------------|---------|---------------------|--------|
| 22/Oct/2023 | 0.1 | Document created | DTTS |
| 07/Nov/2023 | 1.0 | Milestone 4 version | DTTS |
| | | | |

Table of Contents

| | |
|---|-----------|
| <i>Leads Sign-off Sheet</i> | 2 |
| <i>Revision History</i> | 3 |
| <i>Table of Contents</i> | 4 |
| <i>Figures & Tables</i> | 6 |
| 1 Executive Summary | 7 |
| 2 Introduction | 8 |
| 2.1 Purpose | 8 |
| 2.2 Project Document Suite | 8 |
| 2.3 References | 9 |
| 2.4 Acronyms, Definitions and Abbreviations | 9 |
| 3 Scope | 10 |
| 3.1 In-Scope | 10 |
| 3.2 Out-of-Scope | 10 |
| 4 Testing | 11 |
| 4.1 Testing Tools & Environment | 11 |
| 4.1.1 Tools..... | 11 |
| 4.1.2 Environment..... | 11 |
| 4.1.3 Test Data Sources | 12 |
| 4.2 Testing Approach..... | 13 |
| 4.2.1 Unit/Widget Testing..... | 13 |
| 4.2.2 Integration Testing | 14 |
| 4.2.3 Regression Testing | 14 |
| 4.2.4 System Testing | 14 |
| 4.2.5 User Acceptance Testing | 14 |
| 4.3 Testing Automation..... | 14 |
| 4.3.1 GitHub Actions | 15 |
| 4.3.2 Viewing Results | 15 |
| 4.4 Testing Results..... | 17 |
| 4.4.1 Defect Classification | 17 |
| 4.4.2 Pass/Fail Criteria | 19 |
| 5 Requirements Traceability Matrix | 20 |
| 6 Test Summary | 22 |
| 6.1 Test Metrics | 26 |

| | | |
|----------|------------------------------------|-----------|
| 7 | Test Assessment | 27 |
| 8 | Test Results..... | 28 |
| 8.1 | Defects Found During Testing | 31 |
| 9 | Suggested Actions | 33 |

Figures & Tables

| | |
|---|----|
| Table 1: Project Documents | 8 |
| Table 2: Terms, Abbreviations, & Acronyms | 9 |
| Table 3: Software Tools | 11 |
| Table 4: Hardware Specifications..... | 12 |
| Table 5: Requirements Traceability Matrix..... | 21 |
| Table 6: Test Summary Table | 26 |
| Table 7: Test Metrics Table | 26 |
| Table 8: Test Results Table..... | 31 |
| Table 9: Defects found during testing | 32 |
| | |
| Figure 1: Testing Lifecycle..... | 13 |
| Figure 2: Test Results..... | 15 |
| Figure 3: Test Failed | 16 |
| Figure 4: Detailed Test Results | 17 |

1 Executive Summary

This document provides a detailed report on the tests defined within Section 4 of the Test Plan. The tests created support the development efforts of Dream Team Technology Solutions (DTTS) photographic and videographic object detection. CogniOpen's conversational, training, account management, and guided tour features are tested in Team B's companion test report.

The team concocted a broad suite of experiments to analyze the quality of their product. There are fifty-four unit tests. These lightweight routines are configured to execute automatically throughout product development. They exercise specific lines of code, the smallest discernable building block of a program. The unit tests directly invoke procedures implemented in CogniOpen backend code. No modifications of the core program code are permitted if a single unit test fails. All fifty-four passed.

The Flutter Graphical User Interface (GUI) is implemented by derivations of its base Widget class. DTTS wrote five Widget tests. They have both one-to-one and onto mapping to three application screens. The home, gallery, and video pages are individually invoked programmatically. This allowed the team to iteratively fix bugs and encourage imaginative enhancements. All three tests passed.

Integration tests were created, however due to the rapid development environment it became troublesome to keep updating these tests. Therefore, the automated integration tests were removed, and the integration testing became part of the manual testing done by the developers and testers of DTTS. This is a script that imitates user actions in the application, start to finish.

The team created 33 manual tests. These are typical scenarios of application usage. The tests contain actual data gathered by the team in the course of using CogniOpen to complete tasks. The tests isolate particular system functions within the app as well as tested non-functional application requirements. Performed at the highest level, they serve the dual purpose of debugging and instilling confidence in the organizational ecosystem. The team infers the customers will feel the same—they are User Acceptance Tests (UATs). 33 of the 33 tests passed.

Generally speaking, the application is able to identify video-recorded objects with reasonable accuracy.

2 Introduction

CogniOpen is an innovative mobile application used by individuals with disabilities related to memory, struggling to find misplaced items and recall conversations. CogniOpen is implemented with Flutter, a revolutionary extension to the Dart programming language. This technology has been specially designed for mobile platforms. Since CogniOpen is a mobile app, automated testing relies on the robust capabilities of Flutter Unit/Widget/Integration Test packages. Manual testing is conducted courtesy of the Flutter Runtime Application Programming Interface (API).

2.1 Purpose

This Test Report is a wide conspectus of experiments that DTTS conducted on CogniOpen. Automated tests to verify developer logic are presented in great detail. Manual tests validate the usefulness of the product and its adherence to the application's stated intention. Their results are offered, and the overall effectiveness of the app is assessed.

This document guides executive leadership to evaluate if CogniOpen is ready to be released. The development team and Quality Engineers (QEs) collaborate to fix any discovered glitches and/or unexpected limitations. This motivates and inspires subsequent software releases and innovations.

2.2 Project Document Suite

This Test Report is part of a suite of project documents that collectively provide comprehensive project documentation. The suite includes:

| Document | Version | Date |
|---|---------|-------------|
| Project Plan (PP) | 4.0 | 07/Nov/2023 |
| Software Requirements Specification (SRS) | 4.0 | 07/Nov/2023 |
| Technical Design Document (TDD) | 3.0 | 07/Nov/2023 |
| Test Plan (TP) | 3.0 | 07/Nov/2023 |
| Programmer Guide (PG) | 2.0 | 07/Nov/2023 |
| Deployment and Operations Guide (Runbook) | 2.0 | 07/Nov/2023 |
| User Guide (UG) | 1.0 | 07/Nov/2023 |
| Test Report (TR) | 1.0 | 07/Nov/2023 |

Table 1: Project Documents

2.3 References

Fowler, M. (2018, January 16). *Integration Test*.

<https://martinfowler.com/bliki/IntegrationTest.html>

Fowler, M. (2014, May 5). *Unit Test*. <https://martinfowler.com/bliki/UnitTest.html>

Fowler, M. (2004, August 26). *Window driver*.

<https://martinfowler.com/eaaDev/WindowDriver.html>

Gillis, A. (n.d.). *User acceptance testing (UAT)*.

<https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT>

Hamilton, T. (2023, October 21). Test summary reports tutorial: Learn with example & template. <https://www.guru99.com/how-test-reports-predict-the-success-of-your-testing-project.html>

Vocke, H. (2018, February 26). *The practical test pyramid*.

<https://martinfowler.com/articles/practical-test-pyramid.html>

2.4 Acronyms, Definitions and Abbreviations

This section is a glossary of terms, abbreviations, and acronyms used in the Test Report. It reduces or eliminates the ambiguity of multiuse letter combinations between project stakeholders.

| Term | Definition |
|------|--------------------------|
| AI | Artificial Intelligence |
| Ex. | Example |
| GUI | Graphical User Interface |
| QE | Quality Engineering |
| UI | User Interface |
| UAT | User Acceptance Test |

Table 2: Terms, Abbreviations, & Acronyms

3 Scope

This section delineates which aspects of CogniOpen are covered in the test reports.

3.1 In-Scope

Computer vision is the most significant in-scope item of this test report.

1. The application must load without errors when started on a smartphone.
2. Photographic images must be added to the database when they are selected or captured.
3. Videos of live events must be recorded and saved to local files.
4. Images containing objects of particular significance to the user can be specified as such.
5. Objects within photos and videos are identified, presented, and clearly displayed.
6. The application unloads gracefully when exited.

3.2 Out-of-Scope

Team B has created a similar test report for the other product features. CogniOpen has rich conversational functionality. This includes recording, transcribing, parsing, and recalling dialogues with other persons, as well as Artificial Intelligence (AI) agents. There are also ancillary, yet critical, tasks such as user account creation, biometric login validation, onboarding, and a tour of the CogniOpen User Interface (UI). Team B has developed and tested those components, as well.

In addition, there will be no testing defined for external web or data services. These services are assumed to have their own quality standards and to be tested by the organizations who provide the services. These services include:

- OpenAI Chat Generative Pre-trained Transformer (ChatGPT)
- Amazon Web Services (AWS) Rekognition
- AWS Transcribe
- SQLite
- AWS S3

4 Testing

This section defines the overall testing approach including tools, environments, pass/fail criteria, and defect severity level.

4.1 Testing Tools & Environment

This section defines the tools, environments and data which supported the testing.

4.1.1 Tools

The following tools were utilized to complete the software testing.

| Name | Use |
|--|---|
| Android Studio Giraffe 2022.3.1 Patch 1 with Android SDK v34.0.0 | Integrated Development Environment (IDE) used to test CogniOpen on Android |
| Dart Testing Library (test 1.24.6) | Unit testing of the business layer of CogniOpen |
| Flutter SDK Testing Package (flutter_test) | Unit testing of the presentation layer of CogniOpen |
| Flutter SDK Integration Testing Package (integration_test) | Integration testing of CogniOpen |
| Dart Mockito Library (Mockito 5.4.2) | Used to mockup data required for unit and integration testing including data from live web services and or databases. |
| Git | Versioning control system used to track developed test cases. |
| GitHub CI/CD | Used for continuous integration testing. Will trigger unit and integrations tests using event hooks with GitHub. |
| Microsoft Excel | Used to track testing details such as who, when, what and results. |

Table 3: Software Tools

4.1.2 Environment

The tests can be developed and executed in a Windows, macOS or Linux environment. The following table lists the corresponding system requirements needed to successfully use the tools identified in Section 4.1: Testing Tools.

| Operating System | Requirements |
|------------------|--|
| Windows | <ul style="list-style-type: none">OS: Windows 10/11 64-bitCPU: Intel Core i5-8400 3.0 GHz or betterMemory: 16 GB RAM |

| Operating System | Requirements |
|------------------|---|
| | <ul style="list-style-type: none">▪ Free storage: 30 GB (SSD is strongly recommended)▪ Screen resolution: 1920 x 1080 |
| macOS | <ul style="list-style-type: none">▪ OS: macOS 10.15 (Catalina)▪ CPU: Intel Core i5-8400 3.0 GHz or better▪ Memory: 8 GB RAM▪ Free storage: 30 GB (SSD is strongly recommended)▪ Screen resolution: 1920 x 1080 |
| Linux | <ul style="list-style-type: none">▪ OS: Any 64-bit Linux distribution that supports Gnome, KDE, or Unity DE▪ CPU: Intel Core i5-8400 or better▪ Memory: 8 GB▪ Free storage: 20 GB SSD▪ Free resolution: 1920 x 1080▪ GNU C Library (glibc) 2.19 or later |

Table 4: Hardware Specifications

4.1.3 Test Data Sources

The data used for automated unit and integration testing will come from one of the following sources:

- Pre-recorded images, videos and audio files
- Mocked up database and web services using Mockito

4.2 Testing Approach

The diagram below depicts the overall testing approach which was implemented. The types of testing performed as part of this project were unit/widget testing, regression testing, integration testing, system testing and user acceptance testing.

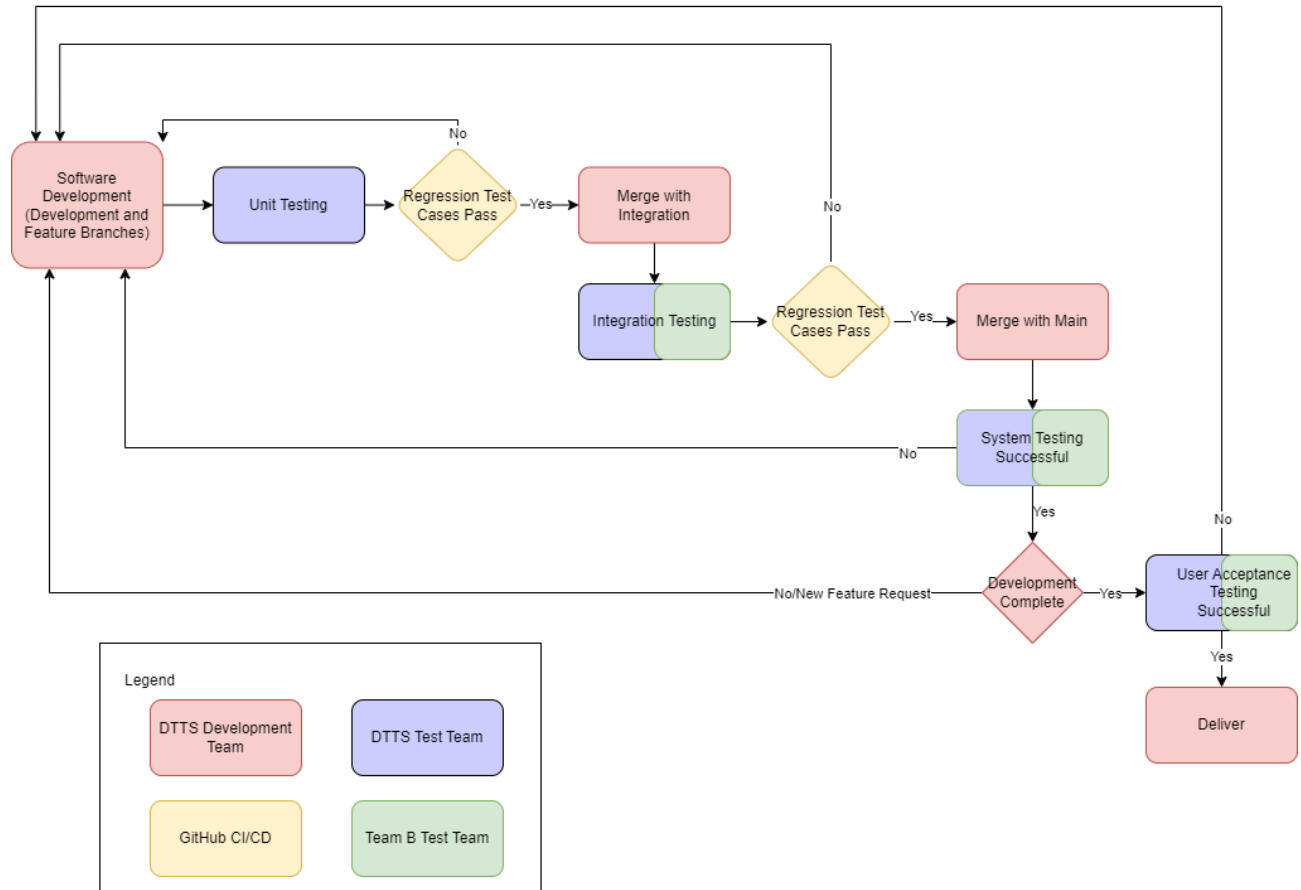


Figure 1: Testing Lifecycle

4.2.1 Unit/Widget Testing

Unit/Widget testing is part of the automated testing process. The individual test cases are documented in the Test Plan, Section 4.1: Unit Tests. This type of testing will be used to verify the correctness of small units of code, functions and widgets within CogniOpen. Unit tests were written and run by the testing team after development on a particular piece of code was completed. Due to rapid development, the unit/widget testing was performed on a unit-testing branch which was based off of the development branch. Upon successful completion of the unit tests, the code is then deemed ready to be integrated into the integration branch.

4.2.2 Integration Testing

Integration testing is part of the automated testing process. The individual test cases are documented in the Test Plan, Section 4.2: Functional Tests. This type of testing was used to verify new components of CogniOpen as they are developed. The integration tests were written and run by the testing team on the integration-testing branch which was based off the integration branch. New integration tests were created based off of the feedback from the software development team indicating that new components of CogniOpen application were complete. Upon successful completion of the integration tests, the new component was deemed ready to be integrated into the main branch.

4.2.3 Regression Testing

Regression testing is part of the automated testing process and will be performed as part of the continuous integration (CI) process. This kind of testing ensures existing functionality has not been altered or broken due to new functionality. Automated regression testing was triggered by a pull request and merged with both the development and main branches.

4.2.4 System Testing

System Testing is a manual process performed by the testing team for DTTS as well as the testing team for Team B. This type of testing was performed on the main branch once major pieces of the application were completed. This testing verified that the system was behaving as expected prior to entering into User Acceptance Testing (UAT) with the client.

4.2.5 User Acceptance Testing

User Acceptance Testing is a manual testing process performed by the testing team and client. This testing was performed once all software development was completed, and the product was ready to be delivered. User Acceptance Testing intended purpose is to validate all functional and non-functional requirements as defined in the Software Requirements Specification document. Upon successful completion of these tests, CogniOpen was delivered to the client along with the test results.

4.3 Testing Automation

As stated above, all unit, integration and regression testing will be automated using the GitHub CI pipeline. A pull request and merge with both the development branch and main branch will trigger automatic testing of all test cases. The results are displayed to the user under the pull request and can also be found under the Actions tab of the GitHub project page.

4.3.1 GitHub Actions

The automation testing and results were setup using GitHub's CI/CD pipeline. To view configuration and results, navigate to the Actions tab under the fall2023 GitHub project. The link is here: [Android CI · Workflow runs · umgc/fall2023 \(github.com\)](#)

4.3.2 Viewing Results

The automation testing results can be viewed under the GitHub Actions tab, a link to the page can be found here: [Workflow runs · umgc/fall2023 \(github.com\)](#)

Each new trigger of the automation testing is called a workflow run. Using the image below for clarity. From the Actions page, each workflow will have a summary item which displays whether all tests have passed (the green check indicates all tests passed, the red x indicated tests have failed), the name of the branch triggering the result (Ex. feature-ui-overhaul branch), the type of action triggered that the automation testing (Pull request #35 synchronize), when it was run (2 days ago) and the duration of the run (one minute and 53 seconds).



| | | | |
|--|---------------------|----------------------|-----|
| ✔ Feature/UI overhaul Android CI #150: Pull request #35 synchronize by zcappella | feature/ui-overhaul | 2 days ago 1m 53s | ... |
| ✘ Feature/UI overhaul Android CI #149: Pull request #35 synchronize by BenjaminSutter | feature/ui-overhaul | 2 days ago 1m 56s | ... |

Figure 2: Test Results

CogniOpen Software Application Test Report

To view more details on which tests passed and/or failed the user can click on the title (Ex. Feature/UI overhaul) next to the passing indicator to open up a details page on a particular run. It will display how many tests passed and how many failed, as shown in the below graphic.

Triggered via pull request 2 days ago

Status

Total duration

Artifacts

BenjaminSutter synchronize #35 [feature/ui-overhaul](#)

Failure

1m 56s

—

android-ci-cd.yaml

on: pull_request

build

1m 45s

Annotations

2 errors and 1 warning

build

29 tests passed, 2 failed.

tells you how many tests passed, and how many failed

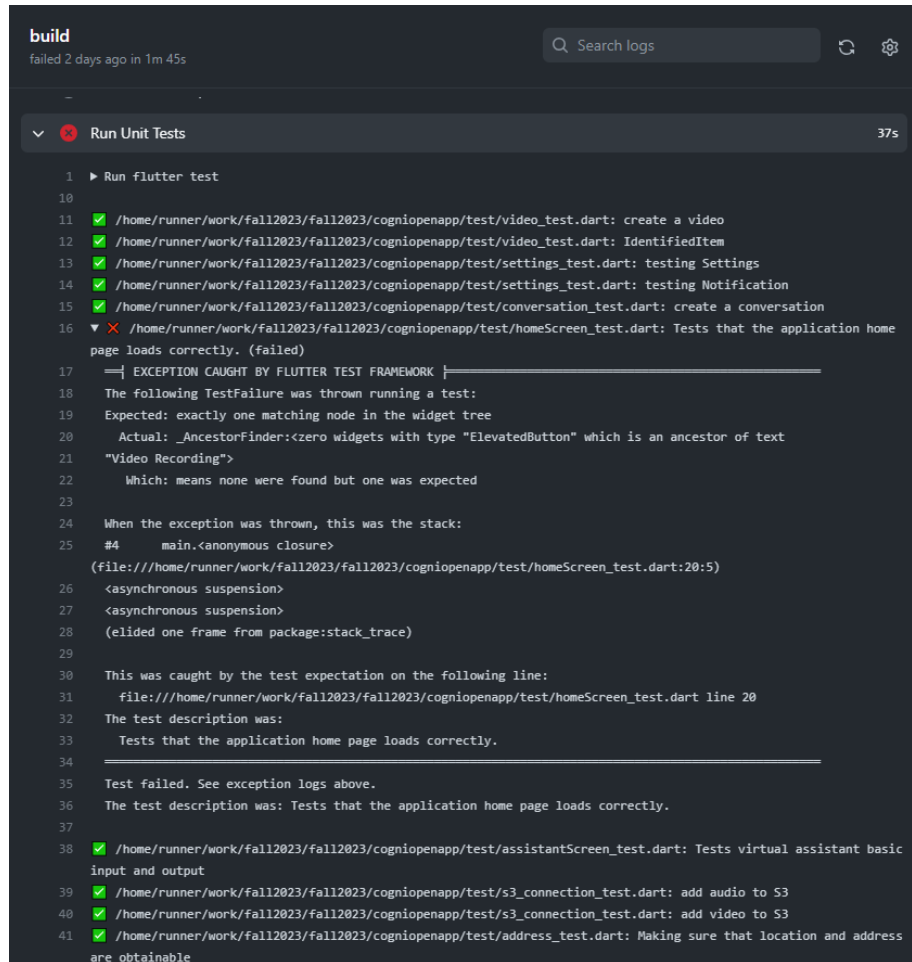
build

Process completed with exit code 1.

Figure 3: Test Failed

CogniOpen Software Application Test Report

For more on the failed and passed tests, click on the build text to open up the details. It will display a line with a green checkbox or red x next to each line which represents one test execution. The white carrot can be clicked on to expand details on failing tests, as seen in the image below.



```
build
failed 2 days ago in 1m 45s

Search logs

Run Unit Tests 37s

1 ▶ Run flutter test
10
11 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/video_test.dart: create a video
12 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/video_test.dart: IdentifiedItem
13 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/settings_test.dart: testing Settings
14 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/settings_test.dart: testing Notification
15 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/conversation_test.dart: create a conversation
16 ✗ /home/runner/work/fall2023/fall2023/cogniopenapp/test/homeScreen_test.dart: Tests that the application home
page loads correctly. (failed)
17 ── EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK ──
18 The following TestFailure was thrown running a test:
19 Expected: exactly one matching node in the widget tree
20 Actual: _AncestorFinder<zero widgets with type "ElevatedButton" which is an ancestor of text
21 "Video Recording">
22 Which: means none were found but one was expected
23
24 When the exception was thrown, this was the stack:
25 #4 main.<anonymous closure>
26 (file:///home/runner/work/fall2023/fall2023/cogniopenapp/test/homeScreen_test.dart:20:5)
27 <asynchronous suspension>
28 <asynchronous suspension>
29 (elided one frame from package:stack_trace)
30
31 This was caught by the test expectation on the following line:
32 file:///home/runner/work/fall2023/fall2023/cogniopenapp/test/homeScreen_test.dart line 20
33 The test description was:
34 Tests that the application home page loads correctly.
35
36 Test failed. See exception logs above.
37 The test description was: Tests that the application home page loads correctly.
38
39 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/assistantScreen_test.dart: Tests virtual assistant basic
input and output
40 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/s3_connection_test.dart: add audio to S3
41 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/s3_connection_test.dart: add video to S3
42 ✓ /home/runner/work/fall2023/fall2023/cogniopenapp/test/address_test.dart: Making sure that location and address
are obtainable
```

Figure 4: Detailed Test Results

4.4 Testing Results

4.4.1 Defect Classification

Priority and severity are not necessarily related. The former considers time. It defines the schedule when an issue ought to be rectified. Some problems, which are seemingly minor, must be addressed forthwith. An example is a spelling error on the home screen. A different error might cause CogniOpen to occasionally crash. However, it might not be realistic to attempt a repair shortly before a deadline.

4.4.1.1 Priority

The following priority levels are available in Microsoft Azure DevOps, and were employed by the team in classifying defects:

Priority: 1

The most elevated designation. It is of the utmost importance that the component or repair occurs immediately. It is unreasonable to expect that CogniOpen is delivered in such a fulsome state.

Priority: 2

The second level means substantial, but not truly tectonic. The issue must be redressed before the end of the CogniOpen Software Development Life Cycle (SDLC). However, other tasks may supersede it temporarily.

Priority: 3

A bug marked as this level ought to be fixed. Whether it occurs is dependent on availability of the development team and outstanding test assignments.

Priority: 4

Software glitches at this level are purely stylistic in nature. They do not affect the product's ability to accomplish its designated computing chores. Rather, possible abatement would clarify ambiguities and streamline CogniOpen.

4.4.1.2 Severity

Severity is a qualitative designation of the effect of a coding mistake on the application as a whole. Microsoft Azure DevOps defines four severity levels, ranging from cataclysmic to trifling. The following severity levels were employed by the team:

Severity: 1 – Critical

It is imperative to repair these problems. These types of errors may yield premature termination of CogniOpen. They could ruin data stores and expose severe security vulnerabilities. There is no viable workaround to be discovered in other buttons or commands.

Severity: 2 – High

This bug is painful for users to endure, but not a showstopper. It represents a noteworthy weakness in CogniOpen. But with guidance and/or technical support, the user may employ an equivalent course of action.

Severity: 3 – Medium

This is a significant blotch. It may cause incorrect output and confusion. The operator could, however, find a comparable method to obtain the desired information.

Severity: 4 – Low

Definitely a weakness. But it is readily ignored or understood for its intended purpose.

4.4.2 Pass/Fail Criteria

For automated testing such as unit/widget, integration and regression testing if test cases pass the test is considered to be passing. If the test cases fail the test is considered to be failing. For individual details on criteria for those tests see the detailed test case pass/fail criteria established in section 4 of the Test Plan.

For manual system and user acceptance testing tests are considered to be passing when no defects were found with priority 1 through 3 and/or severity 1 through 3.

5 Requirements Traceability Matrix

The following table traces the test cases back the requirements documented in the software requirements specifications described in the Project Suite – SRS document. For ease of readability, test cases related to the same SRS have been condensed into a single row entry, with a comma-delineation in the “Test Case No” column.

| SRS No | Description | Test Case No |
|--------|----------------------------------|---|
| SRS-1 | Initialize the Application | U-7-1, U-7-2 |
| SRS-2 | Record a conversation | U-13-1, F-15, F-16 |
| SRS-4 | View conversation in the gallery | U-2-1, U-2-2, U-2-3, U-2-4, U-2-5, U-2-6, U-2-7 |
| SRS-5 | Edit a conversation | F-17 |
| SRS-6 | Set up significant objects | U-8-1, U-8-2, F-04, F-37, F-38, F-39, U-11-1, U-11-2, U-11-3 |
| SRS-7 | Search in location history | F-12, F-30, U-10-1, U-10-2, U-10-3 |
| SRS-8 | Record a Video | U-3-1, U-3-2, U-6-1, U-6-2, U-9-1, U-9-2, U-9-3, U-9-4, U-9-5, U-9-6, U-9-7, U-13-2, F-07 |
| SRS-9 | Pause video recording | F-22, F-23 |
| SRS-11 | Locate an object | F-06, F-11 |
| SRS-13 | Add Media to the Gallery | U-13-3, U-15-1, F-01, F-02, F-05, F-40 |
| SRS-14 | View media in Gallery | U-4-1, U-4-2, U-5-1, U-5-2, U-5-3, U-5-4, U-5-5, U-5-6, U-5-7, F-35, F-36 |
| SRS-16 | View Timeline | U-1-1 |
| N/A | System utilities | U-17-1, U-17-2, U-17-3, U-17-4, U-17-5, U-17-6, U-17-7, U-17-8, U-17-9 |
| N/A | Configure database | U-18-1, U-18-2, U-18-3 |
| N/A | Home Screen Widget Test | W-1 |

| SRS No | Description | Test Case No |
|------------|--|--------------|
| N/A | Gallery Screen Widget Test | W-2 |
| N/A | Video Screen Widget Test | W-3 |
| N/A | Location History Widget Test | W-4 |
| N/A | Significant Object Screen Widget Test | W-5 |
| NFR-Perf-1 | Start-up Time | NFR-01 |
| NFR-Perf-2 | Response Time Navigation Video | NFR-02 |
| NFR-Perf-3 | Response Time Navigation Gallery | NFR-03 |
| NFR-Perf-4 | Response Time Navigation Profile | NFR-04 |
| NFR-Perf-5 | Response Time Navigation Virtual Assistant | NFR-05 |
| NFR-Perf-6 | Response Time Significant Objects | NFR-06 |
| NFR-Rel-1 | Uptime and Availability | NFR-07 |
| NFR-Rel-2 | Backup and Recovery | NFR-08 |
| NFR-Rel-3 | Offline Use Data Access | NFR-09 |
| NFR-Rel-4 | Offline Use Recording | NFR-10 |
| NFR-Usa-1 | Speech Processing | NFR-11 |
| NFR-Usa-2 | Video Processing | NFR-12 |
| NFR-Usa-3 | Status Messages | NFR-13 |
| NFR-Usa-4 | User Guidance | NFR-14 |

Table 5: Requirements Traceability Matrix

6 Test Summary

The test case ID is prefixed with a letter identifying which type of testing it was associated with as defined below:

- U – Unit Testing
- W – Widget Testing
- NF – Non-Functional Testing
- F – Functional Testing

| ID | Application Area | Automated? | Test Case Description | Result |
|--------|---------------------|------------|---|--------|
| U-1-1 | Location Services | Yes | Validating location and address are obtainable | Pass |
| U-8-1 | Significant Objects | Yes | Testing deleting alternate names | Pass |
| U-8-2 | Significant Objects | Yes | Testing adding alternate names | Pass |
| U-7-1 | Settings | Yes | Testing creating user settings | Pass |
| U-7-2 | Settings | Yes | Testing creating user notifications | Pass |
| U-17-1 | Utilities | Yes | Testing formatting storage size – KB | Pass |
| U-17-2 | Utilities | Yes | Testing formatting storage size – Bytes | Pass |
| U-17-3 | Utilities | Yes | Testing formatting storage size – MB | Pass |
| U-17-4 | Utilities | Yes | Testing formatting storage size – GB | Pass |
| U-17-5 | Utilities | Yes | Testing formatting DateTime string | Pass |
| U-17-6 | Utilities | Yes | Testing formatting null DateTime string | Pass |
| U-17-7 | Utilities | Yes | Testing formatting Date string | Pass |
| U-17-8 | Utilities | Yes | Testing formatting null Date string | Pass |
| U-17-9 | Utilities | Yes | Testing calculating the difference between today and DateTime | Pass |
| U-15-1 | File Management | Yes | Testing creating all directories | Pass |
| U-3-1 | Camera | Yes | Testing camera controller initialization | Pass |

| ID | Application Area | Automated? | Test Case Description | Result |
|--------|------------------|------------|---|--------|
| U-3-2 | Camera | Yes | Testing camera initialization | Pass |
| U-6-1 | S3 API | Yes | Testing add audio to S3 | Pass |
| U-6-2 | S3 API | Yes | Testing add video to S3 | Pass |
| U-13-1 | DB | Yes | Testing adding audio to DB | Pass |
| U-13-2 | DB | Yes | Testing adding video to DB | Pass |
| U-13-3 | DB | Yes | Testing adding photo to DB | Pass |
| U-9-1 | DB | Yes | Testing creating video object (all values) | Pass |
| U-9-2 | DB | Yes | Testing creating video object (non-nullable only) | Pass |
| U-9-3 | DB | Yes | Testing creating video object (all values) from JSON | Pass |
| U-9-4 | DB | Yes | Testing creating video object (non-nullable only) from JSON | Pass |
| U-9-5 | DB | Yes | Testing exception thrown when creating video object from invalid JSON | Pass |
| U-9-6 | DB | Yes | Testing creating JSON from video object (all fields) | Pass |
| U-9-7 | DB | Yes | Testing creating JSON from video object (non-nullable fields) | Pass |
| U-4-1 | DB | Yes | Testing MediaFields values set correct | Pass |
| U-4-2 | DB | Yes | Testing MediaFields field names are correct | Pass |
| U-5-1 | DB | Yes | Testing creating photo object (all values) | Pass |
| U-5-2 | DB | Yes | Testing creating photo object (non-nullable only) | Pass |
| U-5-3 | DB | Yes | Testing creating photo object (all values) from JSON | Pass |
| U-5-4 | DB | Yes | Testing creating photo object (non-nullable only) from JSON | Pass |

| ID | Application Area | Automated? | Test Case Description | Result |
|--------|------------------|------------|---|--------|
| U-5-5 | DB | Yes | Testing exception thrown when creating photo object from invalid JSON | Pass |
| U-5-6 | DB | Yes | Testing creating JSON from photo object (all fields) | Pass |
| U-5-7 | DB | Yes | Testing creating JSON from photo object (non-nullable fields) | Pass |
| U-2-1 | DB | Yes | Testing creating audio object (all values) | Pass |
| U-2-2 | DB | Yes | Testing creating audio object (non-nullable only) | Pass |
| U-2-3 | DB | Yes | Testing creating audio object (all values) from JSON | Pass |
| U-2-4 | DB | Yes | Testing creating audio object (non-nullable only) from JSON | Pass |
| U-2-5 | DB | Yes | Testing exception thrown when creating audio object from invalid JSON | Pass |
| U-2-6 | DB | Yes | Testing creating JSON from audio object (all fields) | Pass |
| U-2-7 | DB | Yes | Testing creating JSON from audio object (non-nullable fields) | Pass |
| U-18-1 | DB | Yes | Testing MediaType audio set correctly | Pass |
| U-18-2 | DB | Yes | Testing MediaType photo set correctly | Pass |
| U-18-3 | DB | Yes | Testing MediaType video set correctly | Pass |
| U-10-1 | DB | Yes | Testing LocationDataModel constructor | Pass |
| U-10-2 | DB | Yes | Testing LocationDataModel toMap | Pass |
| U-10-3 | DB | Yes | Testing LocationDataModel fromMap | Pass |

| ID | Application Area | Automated? | Test Case Description | Result |
|--------|---------------------|------------|---|--------|
| U-11-1 | DB | Yes | Testing SignificantObject model constructor | Pass |
| U-11-2 | DB | Yes | Testing SignificantObject toJson | Pass |
| U-11-3 | DB | Yes | Testing SignificantObject fromJson | Pass |
| W-1 | UI | Yes | Testing Home Screen UI Elements | Pass |
| W-2 | UI | Yes | Testing Gallery Screen UI Elements | Pass |
| W-3 | UI | Yes | Testing Video Screen UI Elements | Pass |
| W-4 | UI | Yes | Testing Location History UI Elements | Pass |
| W-5 | UI | Yes | Testing Significant Objects UI Elements | Pass |
| F-01 | Permissions | No | Testing decline camera permissions | Pass |
| F-02 | Permissions | No | Testing accepts camera permissions | Pass |
| F-04 | Significant Objects | No | Testing photo upload | Pass |
| F-05 | Significant Objects | No | Testing photo capture | Pass |
| F-07 | Video Recording | No | Testing video capture | Pass |
| F-11 | Object Search | No | Testing bounding box in a video image | Pass |
| F-12 | Location | No | Testing location history | Pass |
| F-15 | Record Audio | No | Testing first audio capture | Pass |
| F-16 | Record Audio | No | Testing second audio capture | Pass |
| F-17 | Gallery | No | Testing editing media description | Pass |
| F-22 | Video Recording | No | Testing pause video | Pass |
| F-23 | Video Recording | No | Testing un-pause video | Pass |
| F-30 | Object Search | No | Testing locate existing item | Pass |
| F-35 | Gallery | No | Testing view photograph | Pass |
| F-36 | Gallery | No | Testing view video | Pass |

| ID | Application Area | Automated? | Test Case Description | Result |
|--------|------------------|------------|---|--------|
| F-37 | Object Search | No | Testing find one instance | Pass |
| F-38 | Object Search | No | Testing find no instances | Pass |
| F-39 | Object Search | No | Testing find multiple instances | Pass |
| F-40 | Gallery | No | Testing adding a photo to the Gallery | Pass |
| NFR-01 | Performance | No | Testing application startup time | Pass |
| NFR-02 | Performance | No | Testing navigation to Video Screen | Pass |
| NFR-03 | Performance | No | Testing navigation to Gallery Screen | Pass |
| NFR-04 | Performance | No | Testing navigation to Settings Screen | Pass |
| NFR-05 | Performance | No | Testing navigation to Virtual Assistant | Pass |
| NFR-06 | Performance | No | Testing navigation to Significant Object Screen | Pass |
| NFR-07 | Reliability | No | Testing uptime of application | Pass |
| NFR-08 | Reliability | No | Testing backup of media | Pass |
| NFR-09 | Reliability | No | Testing offline photo access | Pass |
| NFR-10 | Reliability | No | Testing backup video access | Pass |
| NFR-11 | Usability | No | Testing speech processing | Pass |
| NFR-12 | Usability | No | Testing video processing | Pass |
| NFR-13 | Usability | No | Testing status messages | Pass |
| NFR-14 | Usability | No | Testing application tour | Pass |

Table 6: Test Summary Table

6.1 Test Metrics

| Test Type | Test Number | Pass | Fail |
|----------------|-------------|------|------|
| Unit | 54 | 54 | 0 |
| Widget | 5 | 5 | 0 |
| Functional | 19 | 19 | 0 |
| Non-Functional | 14 | 14 | 0 |

Table 7: Test Metrics Table

7 Test Assessment

Testing for the CogniOpen application has been successfully completed. Tests for each of the in-scope items listed in Section 3.1 were created and all tests are now passing. There were 9 defects identified throughout the testing process, documented below in Section 8.1. Each defect was assigned an appropriate priority and severity level as defined in Section 4.4.1. Most of the defects found were of low severity and priority, and DTTS Team was able to address and fix 6 of the 9 defects. Defects 7, 8 and 9 do not interfere with functionality, but would provide added value to the application.

The Unit, Widget and Integration tests were all automated through the use of GitHub's CI pipeline. These tests were continuously run on each push and pull request to the main, development and testing branches. These tests helped build the robust regression testing to ensure new features to the CogniOpen application did not break existing functionality. If developers mistakenly broke existing functionality, this allowed them to identify those issues early on in development phase and work with the testing team to fix and adjust tests as needed. All 60 tests passed at the time of delivery to the customer.

As a note to the customer, due to the rapid development environment and change in system design, the automated integration tests were removed from the code base. There was not enough time to keep fixing these tests. Therefore, the integration testing became part of the manual process performed by the software and test teams as new features of the application became available.

The Functional and Non-Functional tests were all manual testing done by the DTTS Testing Team. These tests were written to support the System and User Acceptance Testing to ensure all promised functionality was delivered to the customer. All 33 tests passed at the time of delivery to the customer.

The testing team is confident in delivery of the CogniOpen application to the customer. The team feels that the application has been thoroughly tested creating a reliable and user centric tool to assist individuals with memory related tasks.

8 Test Results

| ID | Automated? | Result | Defects Found | Date Executed | Executed By |
|--------|------------|--------|---------------|---------------|-------------|
| U-1-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-8-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-8-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-7-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-7-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-3 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-4 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-5 | Yes | Pass | Defect-1 | 10/30/23 | GitHub |
| U-17-6 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-7 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-8 | Yes | Pass | None | 10/30/23 | GitHub |
| U-17-9 | Yes | Pass | None | 10/30/23 | GitHub |
| U-15-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-3-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-3-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-6-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-6-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-13-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-13-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-13-3 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-3 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-4 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-5 | Yes | Pass | None | 10/30/23 | GitHub |

CogniOpen Software Application Test Report

| ID | Automated? | Result | Defects Found | Date Executed | Executed By |
|--------|------------|--------|---------------|---------------|-------------|
| U-9-6 | Yes | Pass | None | 10/30/23 | GitHub |
| U-9-7 | Yes | Pass | None | 10/30/23 | GitHub |
| U-4-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-4-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-3 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-4 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-5 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-6 | Yes | Pass | None | 10/30/23 | GitHub |
| U-5-7 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-2 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-3 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-4 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-5 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-6 | Yes | Pass | None | 10/30/23 | GitHub |
| U-2-7 | Yes | Pass | None | 10/30/23 | GitHub |
| U-18-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-18-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-18-1 | Yes | Pass | None | 10/30/23 | GitHub |
| U-10-1 | Yes | Pass | None | 11/6/23 | GitHub |
| U-10-2 | Yes | Pass | None | 11/6/23 | GitHub |
| U-10-3 | Yes | Pass | None | 11/6/23 | GitHub |
| U-11-1 | Yes | Pass | None | 11/6/23 | GitHub |
| U-11-2 | Yes | Pass | None | 11/6/23 | GitHub |
| U-11-3 | Yes | Pass | None | 11/6/23 | GitHub |
| W-1 | Yes | Pass | None | 10/30/23 | GitHub |
| W-2 | Yes | Pass | None | 10/30/23 | GitHub |

CogniOpen Software Application Test Report

| ID | Automated? | Result | Defects Found | Date Executed | Executed By |
|--------|---------------------|--------|---------------|---------------|--------------|
| W-3 | Yes | Pass | None | 10/30/23 | GitHub |
| W-4 | Yes | Pass | None | 11/6/23 | GitHub |
| W-5 | Yes | Pass | None | 11/6/23 | GitHub |
| F-01 | Permissions | Pass | Defect-2 | 11/1/23 | Laura Hamann |
| F-02 | Permissions | Pass | None | 11/1/23 | Laura Hamann |
| F-04 | Significant Objects | Pass | Defect-3 | 11/1/23 | Laura Hamann |
| F-05 | Significant Objects | Pass | Defect-8 | 11/1/23 | Laura Hamann |
| F-07 | Video Recording | Pass | Defect-4 | 11/1/23 | Laura Hamann |
| F-11 | Object Search | Pass | None | 11/5/23 | Vivek Goel |
| F-12 | Location | Pass | Defect-5 | 11/5/23 | Vivek Goel |
| F-15 | Record Audio | Pass | None | 11/5/23 | Vivek Goel |
| F-16 | Record Audio | Pass | None | 11/1/23 | Vivek Goel |
| F-17 | Gallery | Pass | None | 11/1/23 | Vivek Goel |
| F-22 | Video Recording | Pass | None | 11/5/23 | Vivek Goel |
| F-23 | Video Recording | Pass | None | 11/5/23 | Vivek Goel |
| F-30 | Object Search | Pass | Defect-9 | 11/5/23 | Vivek Goel |
| F-35 | Gallery | Pass | None | 11/5/23 | Vivek Goel |
| F-36 | Gallery | Pass | None | 11/5/23 | Vivek Goel |
| F-37 | Object Search | Pass | None | 11/5/23 | Vivek Goel |
| F-38 | Object Search | Pass | Defect-7 | 11/5/23 | Vivek Goel |
| F-39 | Object Search | Pass | None | 11/5/23 | Vivek Goel |
| F-40 | Gallery | Pass | None | 11/6/23 | Laura Hamann |
| NFR-01 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-02 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-03 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-04 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-05 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-06 | Performance | Pass | None | 11/1/23 | Vivek Goel |
| NFR-07 | Reliability | Pass | None | 11/1/23 | Vivek Goel |

| ID | Automated? | Result | Defects Found | Date Executed | Executed By |
|--------|-------------|--------|---------------|---------------|-------------|
| NFR-08 | Reliability | Pass | None | 11/1/23 | Vivek Goel |
| NFR-09 | Reliability | Pass | None | 11/1/23 | Vivek Goel |
| NFR-10 | Reliability | Pass | None | 11/1/23 | Vivek Goel |
| NFR-11 | Usability | Pass | None | 11/1/23 | Vivek Goel |
| NFR-12 | Usability | Pass | None | 11/1/23 | Vivek Goel |
| NFR-13 | Usability | Pass | None | 10/31/23 | Vivek Goel |
| NFR-14 | Usability | Pass | Defect-6 | 10/31/23 | Vivek Goel |

Table 8: Test Results Table

8.1 Defects Found During Testing

The following defects were identified throughout the testing process. The defects were tracked using the Microsoft Azure board using individual bug tickets.

| Defect ID | Description | Priority/Severity Level | Azure Identifier | Status |
|-----------|---|-------------------------|------------------|--------|
| Defect-1 | FormatUtils getDateTimeString not formatting correctly | Level-3/Level-4 | AB#373 | Fixed |
| Defect-2 | On denial, device does not re-prompt user to accept permissions on next use | Level-2/Level-2 | AB#403 | Fixed |
| Defect-3 | Significant object screen does not use application permissions granted when loading the application | Level-3/Level-3 | AB#404 | Fixed |
| Defect-4 | Location toggle in settings does not actually correspond to location service permissions | Level-4/Level-4 | AB#408 | Fixed |
| Defect-5 | Passive video toggle in settings does not actually correspond to video recording settings | Level-4/Level-4 | AB#407 | Fixed |
| Defect-6 | Welcome statement does not provide detail to the user on how to use the Tour Guide | Level-4/Level-4 | AB#226 | Fixed |

CogniOpen Software Application Test Report

| | | | | |
|----------|--|-----------------|--------|-----------|
| Defect-7 | Application does not provide feedback with no object is found in the object search | Level-4/Level-4 | AB#415 | Not fixed |
| Defect-8 | Significant Object tiles sometimes do not automatically populate | Level-4/Level-4 | AB#417 | Not fixed |
| Defect-9 | Photo items in Gallery not sent to Amazon Rekognition | Level-3/Level-3 | AB#419 | Not fixed |

Table 9: Defects found during testing

9 Suggested Actions

The CogniOpen application meets all functional and non-functional requirements of the customer. To increase the amount of testing coverage and ease of integration testing, it is suggested for future users of the application to look into using Appium as a testing alternative to the flutter test framework. The flutter testing framework had various limitations including the ability to interact with native device interfaces as well as the ability to create external API calls.

For future development, the significant objects could be used to make the object search smarter by giving preference to the significant objects. It should also be noted that on various occasions the AWS Rekognition service provided the wrong name of an item. For example, a fire extinguisher was identified as dynamite and a soap bottle as perfume. Alternative solutions should be researched to see if there is any way to increase the performance of object detection.

For an enterprise deployment, the utilization of API keys should be moved from the environment file to a server which would handle the application's requests.

In addition, the usability of the application could be improved through the following enhancements:

- The significant object screen should have a back button without having to click the camera, then the X. The white back button on screen is also hard to see.
- The video screen should say "recording started" rather than "recording resumed" when recording is initially started.
- The object search should allow search by voice, rather than just text.