

1. Management Summary

The purpose of this document is to outline the test requirements and plans for the Android application known as Inspect, created by Binary Giant.

The testers involved will be the Binary Giant team – Michael, Jasmine, Conrad, and Elias. Each member will have different roles for testing which will move around to ensure each member is able to test a various number of scenarios throughout the development to ensure stability among multiple tests by different users.

2. Project Test Objectives

2.1. Security:

2.1.1. Screen Lock (4 - Low Priority):

- Test Objective:
 - System inactivity that locks the screen, then unlocking the screen to resume template editing, saving, or inspection.
- Test Approach:
 - Test the results of what happens to the Inspect applications state when various devices lock when the app is in different states of use and ensure Inspect will be able to resume its previous state.
- Results and Deliverables:
 - When user unlocks device, Inspect resumes in same state it was in.
 - If Inspect detects it has been inactive for X minutes, it will prompt the user if they wish to continue (in case of no screen lock on device).
- Risks/Mitigation Strategies:
 - If the application goes into a “sleep” mode, it may not remember its state on waking if not saved properly and will need to be saved before sleep and reloaded on wake.
 - Multiple devices and multiple Android versions to test how sleep mode interacts with Inspect.

2.1.2. Storage (1 - High Priority):

- Test Objective:
 - Ensure the Inspect application is able to save templates and PDF exports to the Android device’s internal/external storage.
 - Ensure Inspect is able to load templates saved on the Android device for use.
- Test Approach:

- Ensure exported files are correctly saved to a location on the Android device.
 - Verify that templates are imported/exported correctly with no missing or corrupt options.
- Results and Deliverables:
 - Templates are able to be saved and loaded to and from the Android device respectively.
 - PDF exports to the Android device are successful.
- Risks/Mitigation Strategies:
 - Data usage of the Android device requires sufficient storage to be readily available by the device.

2.2. Audit:

2.2.1. Logging (2 - Moderate Priority):

- Test Objective:
 - Ensure logs are presentable and readable.
- Test Approach:
 - Test log file is able to be opened and read.
- Results and Deliverables:
 - Ensure errors presented are able to be used for future patches.

2.3. Performance

2.3.1. Start-up (4 - Low Priority):

- Test Objective:
 - Application launch times.
- Test Approach:
 - Time how long it takes for Inspect to initialise.
- Results and Deliverables:
 - Fast enough so users will not spend time getting disgruntled and upset due to being forced to wait for Inspect to start up
- Risks/Mitigation Strategies:
 - Client can get irritated and cause them to go for another application/method to perform inspections

2.3.2. User Input (1 - High Priority):

- Test Objective:
 - Keyboard input, including loading times
- Test Approach:
 - Testing the speed on input from keyboard to be instantaneous updates to the inspection.
 - Time between user tapping on a keyboard input field and the keyboard appearing on the screen.
- Results and Deliverables:

- Keyboard input should appear instantaneously in the inspection.
- Load time for the keyboard to appear on the screen should be minimal (<1 second)

2.3.3. File export speed (1 - High Priority):

- Test Objective:
 - Speed for inspection export to PDF
- Test Approach:
 - Export various inspections to PDF format to test how long it takes for the export process.
- Results and Deliverables:
 - Minimal speed for exporting to PDF

2.3.4. Camera load times (1 - High Priority):

- Test Objective:
 - Camera open and response speed while user is performing an inspection.
- Test Approach:
 - Test opening the camera and ensure the load times are reasonable.
- Results and Deliverables:
 - No freezes or delays with the camera starting up in Inspect.

2.3.5. Loading Template (1 - High Priority):

- Test Objective:
 - The inspector loads a previously saved template.
- Test Approach:
 - Selecting File Manager > Load Template and then selecting a template to load.
- Results and Deliverables:
 - The SAF successfully loads the saved template upon accessing File Manager > Load Template.

2.3.6. Template load times (1 - High Priority):

- Test Objective:
 - Time to load template to begin an inspection.
- Test Approach:
 - Test opening the camera and ensure the load times are reasonable.
- Results and Deliverables:
 - No freeze's and minimal delays starting up an inspection in Inspect.

2.3.6. Double check export (1 - High Priority):

- Test Objective:
 - Ensure the system is able to export the PDF and verify to the user that information is correct before export.
- Test Approach:

- Test an inspection and export a PDF.
- Results and Deliverables:
 - Ensure the PDF export is correctly structured and only exports when the user confirms.

2.3.7. Rotation Reset (3 - Medium Priority):

- Test Objective:
 - Ensure the application is able to be rotated to landscape view instead of portrait during an inspection.
- Test Approach:
 - The application will be opened by the inspector to start an inspection (by selecting 'Start Inspection') and the device will be rotated to provide a landscape view.
- Results and Deliverables:
 - The screen successfully rotates to accommodate a landscape view and data maintains a correct presentation in its elements.
- Risks/Mitigation Strategies:
 - The application may crash upon switching from a portrait to a landscape view.

2.3.8. Creating Template (1 - High Priority):

- Test Objective:
 - Ensure the application is able to successfully created a template with multiple pages that can later be edited, previewed and then saved.
- Test Approach:
 - The application will be opened via the emulator or installed on an Android device.
 - Load the inspection application from application.
 - Select File Manager > Create New Template to create a new template.
 - Add a heading to the new template.
 - Add editable fields that can be changed.
 - Provide input to these editable fields.
 - Select Add Page to show a new page has been added.
 - Create headings and editable fields with input for this new page.
 - Select Add Space to create a space between the two pages.
 - Select Preview PDF to load the print screen of these two pages.
 - Select back button to return to the template to show the template is still there as the user left it.
 - Select Save to save the file to the phone. This can be confirmed by checking the files.
- Results and Deliverables:

- The application successfully create a template with multiple pages that can be later returned to and edited, saved as a file and previewed.

2.3.9. Completing an Inspection (1 - High Priority):

- Test Objective:
 - Ensure that the application successfully exports the completed template/file and saves it to the proper location.
- Test Approach:
 - Load up the emulator/Android device.
 - The application will then be opened on the device.
 - Once the template/file has been completed select Preview PDF to display the template/file as a PDF.
 - In the dropdown box, go to Select a Printer > Save as PDF option to save the PDF, confirming the name of the output and location to save the PDF to the proper location.
 - Once the PDF is saved to a location, the screen should return back to the inspection form.
- Results and Deliverables:
 - The template is successfully exported as a PDF and saved to the correct location.

2.3.10. Sharing a File (1 - High Priority):

- Test Objective:
 - Ensures the application is able to successfully share a file once the inspection has been completed and the data saved to the device.
- Test Approach:
 - Load up the emulator/Android device.
 - The application will then be opened on the device.
 - Once the inspector has completed the inspection, Share File will be selected which opens the SAF view for the user to select which file to share.
 - The user will select the file they want to share.
 - The user will then select the way in which they want to share the file via the in-built Android share function. They will select the Gmail/Email option to share the file.
 - The user will now have the new email option appear with the selected file attached and ready to be sent to the client.
- Results and Deliverables:
 - An email (file sharing option) with the selected file to be shared attached to it and ready to be sent to the client upon completion of the inspection.

2.3.11. Adding Photo to an Inspection (1 - High Priority):

- Test Objective:

- Ensures the application allows the user to successfully add an image to the inspection form in progress either from camera or gallery.
- Test Approach:
 - Load up the emulator/Android device.
 - The application will be then opened on the device.
 - As the inspection is in progress, the camera input option will be made available to attach photos to the inspection form.
 - Select the camera icon to attach an image via camera or gallery.
 - To attach an image via camera, select Open Camera.
 - Select the camera icon to take a photo, which should display the image taken.
 - Select the tick icon to confirm the taken photo. This should display/load the taken image onto the application screen.
 - Select Accept to show the taken image in the inspection form.
 - To attach an image already taken and available in the gallery, select Open Gallery. The gallery containing any taken photos should load.
 - Select the camera folder, which will display the images taken on the phone.
 - Select the photo in the gallery that is desired for attachment to the inspection form.
 - Select Accept. The selected image should then appear in Inspection.
- Results and Deliverables:
 - Images are able to be taken via camera or selected via the photo gallery to be added to the inspection form while an inspection is in progress.

2.4. Capacity

2.4.1. Data storage (1 - High Priority):

- Test Objective:
 - Ensure all data is appropriately sized and is stored correctly.
- Test Approach:
 - Output file sizes of templates, images, and PDFs.
- Results and Deliverables:
 - Low space used up to ensure user does not overfill their Android device with the Inspect application.

2.5. Availability

2.5.1. Storage (1 - High Priority):

- Test Objective:
 - Test various storage locations to find optimal storage location for user to easily find output files.
- Test Approach:
 - Test Android hierarchy browser in app to find exported items from the Inspect application.
- Results and Deliverables:
 - An easy to locate space on the Android device for the user to manually search through the file system and retrieve exported files, as well as an easy to access method from within the Inspect application.

2.6. Compatibility

2.6.1. Other system input (1 - High Priority):

- Test Objective:
 - Ensure camera and keyboard both work correctly.
- Test Approach:
 - Ensure inputs from Android devices keyboard and camera are inputted into the Inspect application correctly
- Results and Deliverables:
 - Fully functional keyboard and camera integration with the Inspect application

3. Test strategy

3.1. Product risk analysis (see template)

Risk	Classification
Data loss	Storage, Security, Stability
Application crash	Stability, Compatibility
Slow User Interface	Performance

3.2. Test Levels – Test Strategy

Test level	Goal
Unit Testing	Ensure methods within classes give expected output and meet user specifications.
Integration Testing	Ensure that two modules behave and function as expected after integration.
User Acceptance Testing	Ensure that Use Cases can be executed and the system behaves and functions as expected in real world scenarios.

4. Infrastructure

4.1. Test environments

Test level	Test environments
Unit Testing	Android Studio
Integration Testing	Android Studio
User Acceptance Testing	Emulators and Physical Devices

4.2. Test Tools

Test level	Test Tools
Unit Testing	J-Unit
Integration Testing	Mockito and Espresso
User Acceptance Testing	Emulators and Physical Devices.

5. Test management

5.1. Test process management

Test level	Guidelines
Unit Testing	Is done by the author of the code before merge and commit to master.
Integration Testing	Will be done by all team members for all components being integrated during each Elaboration Phase.
User Acceptance Testing	Will be done internally by all team members in the last Elaboration Iteration and for all Construction Phases. Externally this will occur during the Transition Phases.

5.2. Defects Procedure

For the registration and maintenance of defects the following tool will be used: Finotes.

- Defect is identified
 - Check defect list to see if defect has already been identified.
 - If defect is already present in the list, team member will fill in any missing details. No further action is required.
 - If the identified defect is not present, add defect to the defect list and complete all necessary fields.
 - Assign a team member. Ideally it will be the author of the code that the defect appears to be in.
- Investigation by assigned team member:
 - Reproduce issue.
 - Determine source of defect or
 - Determine it is not a defect and mark as rejected.
- Resolution:
 - Repair the issue.
 - Mark as "To be verified".
- Verification by Team member who initially reported the defect:
 - Verifies defect is resolved.
 - If it is resolved mark as closed, otherwise reopen defect.