Inspect
Vision Document

1. Introduction

The business problem our software solution will solve is the duplication of paper work created by on-site inspections. Often traditional paper forms are taken on-site and filled out, then at a later date and time, digitised and photos added. Our solution hopes to bridge the gap between initial on-site inspection and the digitisation process that occurs later and in the process saving resources, time, and money.

2. Positioning

2.1 Problem Statement

The problem of filling out traditional forms only to digitise them affects employees filling out the form and those tasked with digitising them. The impact of which is hours of labour digitising paperwork, spent resources used in printing, and later the secure destruction of these documents. A successful solution would be to remove the traditional paperwork by doing it in a digital fashion on-site while doing the inspection.

2.2 Product Position Statement

For any business that performs inspections who currently uses a traditional low tech approach. Inspect is an inspection tool that would streamline the inspection process considerably. Unlike traditional low tech inspection processes that rely on paper as an intermediary format between the inspection and digitisation process, our product would effectively cut out the middleman (paper) and digitise the paper work immediately with the added ability to take photos to be added and manipulated with ease to the inspection end result. This would allow for additional information to be easily seen; where defects or other points of interest are accentuated, especially in cases where they may be difficult to discern.

3. Stakeholder Descriptions

3.1 Stakeholder Summary

Name	Description	Responsibilities
Jasmine Booth	Producer	● Understanding scope of assigned tasks to be completed ● Effectively manage their time/schedule ● Performing tasks within the timeline and quality expectations ● Following the planned assignments ● Assumes personal responsibility for achieving assigned tasks ● Tracking and logging their work contributions in the iteration plan ● Communicating issues, changes, risks, and quality concerns to the team ● Communicate and collaborate with other team members ● Assist others in the completion of their tasks to support the group goals
Michael Coleman		
Conrad Fleming		
Elias Zanbaka		
Jasmine Booth	(All) User	● Should be able to report any errors that can be effectively logged, tracked and dealt with promptly. ● Should have access to updates which will improve overall functionality and deal with existing bugs. ● Should be provided with in-app assistance such as tutorials and other supplementary documentation that can minimize the learning curve with regards to use of the app.
Michael Coleman		
Conrad Fleming		
Elias Zanbaka		

		● Be involved in the testing phases of the app design and encouraged to contribute ideas, suggestions, or user stories to enhance overall usability, functionality and features.
Jasmine Booth	(Office) User	● Be able to create, edit, save, discard, preview, export and share (with the client) a custom made template with fields tailored to the needs of the inspection. ● Be able to Preview form template
Michael Coleman		
Conrad Fleming		
Elias Zanbaka		
Jasmine Booth	(Inspector) User	● Fill in text fields. ● Be able to enhance the quality and accuracy of information through the use of photo attachments, either taken on the spot or imported from an existing library/album. The user should be able to crop and annotate them to better communicate the inspection faults and courses of action to the client. ● Export completed form as pdf
Michael Coleman		
Conrad Fleming		
Elias Zanbaka		

3.2 User Environment

The working environment is a mobile one, always requiring the inspector to be on-site. Usually one person, however in the case of a team it is usually one person's job to do the documentation. A task may be a simple rental inspection with one document or a more complicated construction compliance inspection spanning multiple documents and standards. As the standard procedure is to digitise the documents we will be streamlining existing processes and not need to integrate into any existing applications. Inspect will have a template editor so that as standards change or a unique situation presents itself the existing templates can be easily updated or new templates can be made.

4. Product Overview

4.1 Needs and Features

Need	Priority 1-4	Features
Need a way to modify templates	1	Add elements to template
	1	Add module to template
	1	Remove element from template
	1	Remove module from template
	1	Preview a template
	1	Save/save as modified template
Need a way to manage templates, pdfs, and save states	2	Load a template for inspection (so that user can fill data during inspection)
	2	Load template into template editor (for modification)
	2	Create a new blank template (for modification)
	4	Export template (to share)
	2	Delete object (template, pdf, saved template states)
Need a way to take photos	3	Integrate camera
Need a way to alter photos	4	Simple photo editor
Need a way for users to fill data fields during the “inspection” (Input)	3	Input text into fields
	3	Attach photos to image field
	3	Clear field/s

Need a ubiquitous format to export completed inspection forms to	1	Export to pdf as it is a common file type that retains formatting over different systems and programs.
Need a ubiquitous way to distribute final output	4	Open email client and auto attach object (pdf/template).

5. Other Key Product Requirements

Category	**Requirement**
Usability	It is important the system is easy to use with good design practices implemented.
Reliability	The system will be designed to work offline.
	The system needs to be stable and reliable to prevent loss of work.
Performance	The application needs to be responsive to user commands.
	During heavy processing scenarios, loading screens and prompts will notify user of applications status.
Security	The application will not store user data.
	The application will not require logins. Third party applications may require logins (eg. email client)
	The application will not transmit data. Any data sharing will be handled by third parties.
Audit	Errors and current status will be logged, but not user data.
Compatibility	Mobile phones running android operating system.
	Touch input and rear facing camera required.

Maintainability	Modular architecture with components that are cohesive and loosely coupled to aid in code readability, debugging and make updating features simpler.

6. Change log

Elaboration Iteration 1 (28/04/2019)	○ Implemented Conversion Manager ○ Implemented Log Manager ○ Implemented PDF export ○ Commenced work on implementing the custom template system ○ Commenced work on the custom template editing system
Elaboration Iteration 2 (12/05/2019)	○ Finalised implementation of the template ○ Progressed work on the custom template editing system ○ Commenced work on file management system
Elaboration Iteration 3 (26/05/2019)	○ Finalised implementation of the file management system ○ Implemented photo manager & camera integration to the application ○ Commenced work on the template manager ○ Commenced work on the file share management system
Elaboration Iteration 4 (09/06/2019)	○ Corrected issue with photo management system ○ Commenced work on implementing file management with file share and template editor ○ Progressed work on the template manager system ○ Progressed work on the file share management system
Elaboration Iteration 5 (23/06/2019)	○ Finalised implementation of the template editor ○ Finalised implementation of the file management to include saving/loading a save state ○ Finalised implementation of the file share manager ○ Finalised implementation of the photo manager with rest of application ○ Implemented inspector.

7. Reflection

Integrating the Storage Access Framework (SAF) proved to be quite troublesome while simultaneously learning the Android structure. The ability to navigate through the storage was required for the application. Fortunately we had the ability to utilize this system provided by the Android OS. The SAF also integrated with the share and camera functions, this allowed saving, loading, and sharing files with the application due to its ability to grab the URI of a selected file.

There were some problems initially in ensuring the application was successfully saving and reading log files via the Log Manager. The method itself had to be revised from previous iterations as the buffered writer employed was not creating the log files in the directory. Even though we tried to pinpoint the issue through research and online tutorials and came up short in identifying the issue with the buffered writer, an alternate method in outputFileMethod was employed, which managed to successfully read, save and create the file.

With File Share Manager, there was also some difficulty concerning the registration of certain aspects of the class within the Android Manifest, but this was able to overcome this through a combination of extensive research online, assistance from other team members and correctly updating the File Provider permissions within it. Other areas where issues arose included a few of the methods that needed to call variables, which in turn needed to be set up by creating a filepaths .xml with TextView. Trying to run these components proved bothersome as the emulator either kept shutting down or failed to load certain aspects of the application that appeared to run fine for other team members; however, this was soon rectified by creating a new emulator and running the application on it.

There were initial issues regarding the File Provider, mainly in returning the correct values for the URI’s in creating a file (and directory) and sharing it. This also lead to other issues such as being unable to take a photo and save it to the photo gallery. We ended up working on alternate implementations, ensuring null values weren’t being returned for the URI’s, which was part of the initial problem. Through a process of trial and error and extensive research, we were able to correct these errors so that files could be shared and photos saved to gallery.