Transition Phase Status Assessment

1. Assessment against Objectives of the Transition Phase

1.1 What were we trying to achieve?

The goal of the transition phase was to bring the project to a production ready state and be ready for long term support.

1.2 What have we achieved?

We are happy to report that we achieved all our aims within the phase:

- Implementation Model
- User Acceptance Tests
- Resolution of all issues found
- Programmer Documentation
- Video Demonstration

1.3 What still needs to be done?

- Nothing

2. Deliverables

2.1 Implementation Model

Feature complete with no known bugs.

2.2 User Acceptance Tests

To make this as user friendly as possible we deployed the beta apk using an opt in link and Google Play services so that testers would not have to enable developer options to allow applications from unsigned sources.
Feedback was given using google forms so there were no delays receiving submission of feedback.
We had two rounds of external user acceptance test with bug fixes applied in between. The first round to expose issues and the second to ensure fixes resolved the issues.
Potential testers were supplied a care package with links to opt in, user manual, and feedback forms.

2.3 Issue Resolution

All issues discovered were resolved. As feedback came in it was processed and issues created in Github issue tracking.
Most notable of the issues were ones that were device specific and difficult to replicate without having that specific device on hand. Samsung devices proved difficult as they have no local emulation options only online emulators (4 per device) which were never not in use.
Documentation to support feedback resolution can be found in the PMAS links.

2.3 Programmer Documentation

Documentation has been prepared so that a sufficiently skilled development team can maintain the code base into the near future.

2.4 Demonstration

A short demo video of the applications features and use cases.
https://youtu.be/QIJes-Us2M4

3. General Issues

Entering transition phase we did not have any issues however finding testers proved quite difficult despite our best efforts as it turns out that testing software requires more effort than most people will voluntarily sign up for.
Some testers did not complete the tasks or the form at the end, instead only having a look around the app and giving informal feedback such as "it looks good".
Replicating bugs proved difficult with the limited information from testers and this was further compounded by not having a larger range of devices to test on.

4. Risks

We entered the transition phase with as minimal risk as possible though a couple potential risks still existed.

4.1 File Chooser Library

The library for choosing files was implemented rather late in the construction phase and potentially could have had unknown issues.

4.2 Device Specific Issues

During every phase of development we regularly encountered device specific issues due to androids open source nature and manufacturers implementing their own changes fracturing the device landscape, so it was no surprise that entering testing that there would be a risk that we would not be able to broadly support all android devices due to not having the device on hand.

5 Summary – Overall Project Progress

The project is completed as per the main functional requirements of the project:

- Create Form Templates
- Modify Form Templates
- Loading and Saving Form Templates
- User fill forms (during inspection)
- Saving User Form
- Export to pdf
- Distribute via email (in fact you can share with any app that is registered to do so)

6 Project Reflection

Despite being chosen because all team members knew Java, Android development presented a number of challenges which forced the team to really come together and solve, pooling everyone's resources and efforts when troublesome work items threatened to hinder progress and cascade into subsequent iterations. This highlights how knowledge of the platform and how its internal classes operate present more of a learning curve than knowing the language required to use them. An example of this would be selecting a file using FileProvider. The android content provider can't open at a predefined folder. This is not immediately obvious. This issue required the whole team to resolve after many hours of research and trailing external libraries to replace our original implementation.

This knowledge gap was a common theme that followed us through every phase, during the Elaboration phase, it was the UI life cycle and phone security, during the Construction phase the FileProvider, and even in transition phase ignorance of specific devices where the manufacturers bastardization of the android operating system became apparent.

Team management presented its own unique issues. We initially had Trello but found it was duplicating work that was already in the weekly iteration plans and project plan and opted to drop it. During construction we tried an issue tracker/ticketing system however the team found it just as easy to list issues in the iteration plan as this was necessary for the assessment anyway and the tracker was basically again duplicating work. Finally, during the Transition phase, we tried Github's issue tracking for sorting feedback but with so few issues it is hard to say if it would have been worth the extra effort. Basically since the team had to track work on the iteration plan, when things got tough and no matter how good the intentions, other issue

tracking/planning software that would commonly be used in production would invariably get dropped and the iteration plan and chat would be used. Having a small team, coupled with maintaining a high level of communication and coordination, allowed us to succeed in this respect, whereas it is doubtful that a larger team or less communicative team would have as easily gotten by using just an iteration plan and team meetings as a sole team management solution.