**Summary of Application**

The report begins by explaining the current concerns with smartphone applications, and due to the introduction of new privacy laws such as the EU’s GDPR, developers need to be more proactive in protecting their user’s privacy. The report explains that prior research has identified widespread misbehaviour, that developers “usually treat privacy as a secondary concern”, and that developers frequently have a weak understanding of data privacy protection.

The researchers have identified a lack of support for creating privacy-friendly applications. The goal is to produce tools that aid developers to better manage the requirements for privacy; especially for those that lack the funds to have a dedicated privacy team.

The report goes on to explain their aims to achieve this goal. They set out 4 clear privacy principles. These principles are User-Facing Notice, Consent and Control for Privacy; Data Collection; Data Transmission; Data Retention. These aspects were inferred from guidelines created by governments, non-profit groups, and the industry.

A study was conducted to understand how current Android developers manage privacy. These developers were recruited, interviewed on their experience, privacy training background, and perceptions on privacy. A further interview focused on recent applications those developers had created. The results from the interviews suggested poor understanding of privacy, poor understanding of their application, a lack of knowledge of alternatives, privacy was a secondary concern, and a lack of motivation. This reinforced the conclusions of the previous studies conducted.

With these conclusions, the decision was made to create an Android Studio IDE plug-in. A fictional use case was created, concerning a run tracking app. Examples given in this use case included using java metadata to describe how and why location data is used; a quickfix to auto generate an annotation skeleton; hovering fields giving an in-depth explanation; specific error identifiers where metadata is missed; forcing developers to describe what data may leave the phone at relevant points.

The report explains each choice, and why that would help satisfy the goals. A proof-of-concept was created consisting of the plug-in and a support library. The plug-in focuses on the code, detecting changes, informing the developers of problems. The support library is a small group of APIs and third-party libraries that use personal data that works with Coconut, the name given for the implementation.

An evaluation was performed, using a series of lab studies with 18 participants. These participants were split into two groups of roughly equal experience: a control and an experimental group. The participants performed a warm-up task, and a main task. Only the experimental group used the plug-in for the main task. An exit survey was conducted, part of which asked the participants to write a privacy policy. Finally, the participants were interviewed on their privacy considerations.

The experiments were evaluated by external judges who graded the feedback given by the participants.

The results of the experiment suggested that the Coconut tool did help the developers write better privacy-preserving code, help developers better understand the apps behaviour, and helped write better privacy policies. Coconut was also noted to be perceived as “useful and usable”. The report does mention some limitations of the experiment – such as participants focussing on getting the app to work first and then considering privacy.

The report concludes stating the tool successfully fulfils the goal, and then a very long appendix.