CS898AE Surya Devathi
N659A624

Paper Critique – 8

1.
This paper is about the importance of finding KR errors, the motivation of it is to begin with few examples of these errors. Then, KR data handling in Android discuss restarts and platform support and present a strong example of the incorrect KR data handling and the loss in the Data Slider open source app.

2.
The KR errors are formed when there is an absence in the KR data saves or restores which are usually critical. Developers handle the KR data, still the handling can be unpredictable which further leads to the errors. These can be spotted using data-flow and control-flow analysis.
There are four types of errors:

(1) Type – 1: A KR field has just one spare activity that is put in techniques that are not ensured to be called by the AF, subsequently the KR field worth may be lost.

(2) Type - 2: A KR field has at any rate one spare activity, yet there exists a way from a data change instruction to a movement or application exit AND the way doesn't contain any spare activity for the information; henceforth the KR field worth may be lost.

(3) Type - 3: A KR field has no spare and reestablish tasks. In the event that there exists an explanation that changes the field, the changed worth will be lost, however a KR blunder doesn't really happen.

(4) Type - 4: A KR field doesn't have a spare activity, yet has a reestablish activity, example, when the designer neglected to summon a spare technique. On the off chance that the KR field is reestablished and, at that point transformed, we induce that it ought to have been saved.

3.
It is normal to expect that engineers use onSaveInstanceState(), onStop(), onDestroy(), or onPause(), to spare state. Sadly, depending on these callbacks may be risky.
In the first place, onSaveInstanceState() can't to be called, as referenced priorly, nor is onStop(). Second,onStop() and onDestroy() are 'killable': per the Android documentation - after that strategy restores, the procedure facilitating the movement might be killed by the framework whenever without a different line of its code being executed. Since onStop() is killable, onDestroy() may never be summoned. Henceforth the only “sure wager" for engineers sparing the KR information is in onPause().