Name: ALI ABDUL BASIT ALI MEMON

ROLL NO: 2K18/CSME/4

SUBJECT: SOFTWARE ENGINEERING

TEACHER: GULSHER LAGARI

GITHUB LINK: https://github.com/basit376/2K18-CSME-4-ALI-ABDUL-BASIT.git

Title: ***Refactoring Android-specific Energy Smells: A Plugin for Android Studio***

Authours***:*** ***Emanuele Iannone, Fabiano Pecorelli, Dario Di Nucci, Fabio Palomba, Andrea De Lucia***

Publish: ***Mon 13 Jul 2020 17:06 - 17:18 at ICPC - Session 2: Quality Chair(s): Gemma Catolino***

Introduction*:* ***He work on the Energy smells are of the presence of poor implementation choices that badly affects energy consumption in (Android Studio) mobile applications. firstly he compared to famous Detection tools aDoctor and Paprika that are already available he work on aDoctore and extend aDoctor architecture as automation refactoring and publish on android studio.***

Research : ***He Work on 5 Major energy smells issues 1: Durable Wakelock 2: INFFICIENT DATA STRUCTURE 3: LEACKING THREAD 4: MEMBER IGNORING METHOD 5: INTERNAL SETTER***

Methodology: ***Adoctor tool first analyze the mobile applications then it proposes a refactoring solution for each detections and then the user agree it applies Refactoring automatically:***

Summary: *This tool first extract abstract syntax trees from the source code then relying Eclipse JdT Code library it analyzes andry factors the code directory on extracted syntax trees finally the changes are propagaded to source code Adoctor tool can find and fix different energy smells durable 1: Durable Wekelock 2: INFFICIENT DATA STRUCTURE 3: LEACKING THREAD 4: MEMBER IGNORING METHOD 5: INTERNAL SETTER 6:EARLY RESOURCE BINDING*

MOTIVATION: *WE CAN CREATE A AUTOMATIONS TOOLS AND PUBLISH FOR PEOPLE THEY USE IT AND SAVE HIS/HER TIME IN FINDING ERRORS IN THEIR CODES:*

Results: *WE GET IT VERY USE FOR DEVELOPER TO USE Adoctor tool to refactoring automatically these 6 major energy smell components in java mobile applications codes:*

Name: ALI ABDUL BASIT ALI MEMON

ROLL NO: 2K18/CSME/4

SUBJECT: SOFTWARE ENGINEERING

TEACHER: GULSHER LAGARI

GITHUB LINK: https://github.com/basit376/2K18-CSME-4-ALI-ABDUL-BASIT.git

Title*: RTj: a Java framework for detecting and refactoring rotten greentest cases*

Authours: *Matias Martinez, Anne Etien, Stéphane Ducasse, Christopher Fuhrman*

Publish*: Wed 8 Jul 2020 16:25 - 16:28 at Silla - A12-Testing Chair(s): Sasa Misailovic*

Introduction*: he work on Rotten Green tests are passing tests which have at least one assertion not executed and rotten test give developers a false confidence the he present RTj a frame work tool*

Methodology*: A rotten green test is a tests contains assertions that are never executed and rotten test giving developers a false confidence because beyond passing an assertion that should validate some property is in fact not executed for helping java developers to detect green test and create a framework called RTj RTj needs to run on java virtual machine 8 Moreover Maven must be installed on your machines and accesible from command line in analysis of any project RTj can analyze any kind of Java project, not necessary Maven project. For that, it will be necessary to pass (via command line arguments) to RTj additional information about the project under analysis. As by default, RTj calls project-info-maven-plugin, to disable this feature, pass the argument autoconfigure false. Then, use the following arguments to pass the information related to the project to analyze. -location "absolute location of the project to analyze"*

-dependencies "folder with the dependencies of the application to analyze"

-javacompliancelevel "compliance level of source code

-srcjavafolder "source code folder"

-srctestfolder "test source code folder"

-binjavafolder "class folder"

-bintestfolder "test class folder"

MOTIVATION : work for detecting and refactoring smelly test.
: We can see that Rotten Green tests can give false confidence in testing their projects and RTj give automatic solution to developer for thier projects and we provide developer a unified frame work for detecting and refactoring smelly test.

Result: *The Main Result of this paper RTj a frame work that analyzes statically and dynamically test cases with the goal of detecting smelly test including rotten test cases rotten test give devlopers to false confidence and RTj give helping to developers automatically detect*

those smelly test and then be refactoring to improve the quality of an application

Name: ALI ABDUL BASIT ALI MEMON

ROLL NO: 2K18/CSME/4

SUBJECT: SOFTWARE ENGINEERING

TEACHER: GULSHER LAGARI

GITHUB LINK: https://github.com/basit376/2K18-CSME-4-ALI-ABDUL-BASIT.git

Title*: Doodle2App: Native App Code by Freehand UI Sketching*

Authours: *Soumik Christoph Csallner Mohian,*

Publish: *Tue 14 Jul 2020 16:25 - 16:35 at MobileSoft - Software Development and Evolution Chair(s): Mariano*

Introduction*: In Modern time here most challenging work for developer to make a user reliable in this paper he work on freehand UI Sketching and make Doodle2App 712 sketch samples achieve higher accuracy then the state of the art tool Teleport in this UI Sketching App we work Freehand prototype UI Designers today have to Manually Recreate Prototype in high fidelity tools such as Photoshop or in a ide Which is laborious and costly for developer*

Methodology: *Doodle2 App based on Integrating Freehand UI Sketch is hard to maintain well known benefits The Doodle2App tool is web based and provides a canvas for sketching with mouse or tochscreen user can create one sketch of UI at one time at a time The Doodle2 App Currently support 21 UI elements types If user makes his project with Freehand UI sketch At Any time the designer can export the current UI state to a compilable for android app*

MOTIVATION : *In this paper the motivation for developer to make his project desing with freehand and draw desing with mouse and convert it to android code to make an app for user reliable*

Result: *This paper is All about Freehand UI sketching Prototype Contains 21 UI sketch (eg. Avtar , back , cancel checkbox dropdown , Forward, Left arrow , Menu , Play, Plus Search, Setting, Share , Sliders , Squiggle, switch, star envelope, camera,*

square) and each UI prototype contains 712 quick draw scetch and the user can draw with mouse and convert into Android code and make Android app with Android studio