NAME: KASHIF ALI

ROLL NO: 2K18/CSME/14

SUBJECT: SOFTWARE ENGINEERING

TEACHER: SIR GULSHER LAGARI

GITHUB LINK: https://github.com/kashif310/2K18-CSME-14.git

Summary 01:

TITLE :

UI Screens Identification and Extraction from Mobile Programming Screencasts

Author Mohammad Alahmadi, Abdulkarim Khormi, Sonia Haiduc

## Track: ICPC 2020 Research

## Date :Mon 13 Jul 2020 15:12 - 15:24 at ICPC - Session 1: Tests Chair(s): Dario Di Nucci

Introduction and motivation:

Smartphones are the most widely used devices nowadays and more than 2.5 million users used worldwide. There are most two popular app stores which are the apple app store and Google play store. There are millions of peoples using screencasting features. So it can be very useful a developers to find programming questions online in videos that are hosted by YouTube about programming. This is due to the hard part of YouTube, which hosted very limited information.

Motivation:

In this paper. We take a step forward to look at this problem by localizing, extracting, and presenting most respective UI screens to the developer. It can be helpful to compare quickly by developers to see an overview of a video about apps developed screencasts programming about relevant to their information needs. We keep a deep eye on the UI of an app since it captures and provides the features and it in actions. Ours focused to explore UI overview and it's based on deep convolution neural network(CNN) which has include features of images extractor and detectors of object UI screens to locate within the frames of a programming screencast. The UI screen is extracted and filtered only the unique UI screen is kept. And this approach integrated into the tool which is also available online for use.

Importance:

## This research is important for developing a new mobile app and focused on its features many developers find their QAs about mobile programming of screencasting materials on youtube, so the UI screencasting is better to understand deep mobile programming, video mining, mobile developers

Methodology:

## We have a 10% fold cross-validation performed, we divide data into two sets of folds and trained, validate, and test the sets. While we have training 80% remaining set of the dataset. The testing and validation of each set 10% during training and ovoid for overfitting. Our training process is over 4,000 iterations until the network stopped and improving or validation loss stable. The first move just before for locating and extracting UI screens from Mobile programming screencasts, given that a UI overview of a video. It can allow developers to rapidly understand which the main features of an app give an explanation are in screencasts and decide if the video is relevant to their in order needs. An assessment based on two practical studies showing that the planned approach is not only perfect but also considered useful by developers. A freely available tool to implement our approach.

Result:

## There have been several works that analyzed the changes made to the GUI during the development of mobile apps, with the purpose of documenting them. Some approach address this problem by Automatically detecting the changes to the

GUI and abbreviation them extract a set of GUI images and their metadata files from different versions of an app and compare them using computer vision algorithms. The main goal of our work is different, as we focus on screencasts. So it's big goal to achieve of UI Screen casting benefits to make developing most advance in developing world

Summary 02:

Tittle:

SimplyHover:

Improving Comprehension of else Statements

Author Name:

Ahmad Jbara, Bar Ben Michael, Or Shacham, Omer Tavor

Conference Name:

## ***ICPC 2020 Tool Demonstration***

# ***Date:***

## ***Wed 15 Jul 2020 00:24 - 00:36 at ICPC - Session 9: For Developers Chair(s): Anderson Oliveira***

Introduction:

In large blocks of code, it's essential to know to make any maintenance activity. if it's particularly governed by if-else statements in large code blocks its increase spatial distance between the if-statement and else equivalent or other dependent else-statements. It's hard to the developers to recall the if logical expression that he has just read and understood .so he needs to go back and read it again. In most cases, if-else pair might occur in different pages so developers have to scroll up and down to gossip them both. Furthermore.

It needs to understand else implicit negated expression he might want to go back to the farthest if-statements and understand the cumulative logical expression all he make understanding if else even harder.

Important:

Its importance is to simply hover on the else distant conditions and helps to simplify them. Its great feature to compare of if statements it's else counterpart in large code blocks.

Motivation:

To understand else implicit expression in the large code block. most of cases developers have a large block of code with multilevel logical expression with if else-statements there is hard to know which else-statement of which if-statement part. So simply hover gives the feature just hover on it and its also applying negation to get the simpler condition.

Methdology :

We take an unsplit method of java from JEdit in which we have 3 else-statements in large blocks of code for comprehending else blocks we used simply hover plug-in tool

in Eclipse IDE that brings the if condition next to its else counterpart. simply hover presents to the developer to know hover over else the if conditions in their negated form and even lets him own simplification for that.

Result:

We get it very useful for developers to understand the large code of blocks of comprehension of else-statements. Simply hover brings to else the distant conditions and helps to simplify them.

Summary 03:

Tittle:

Open Problems and Challenges in Ownership Management

Authurs:

JohnAhlgren ,Maria Eugenia Berezin, Kinga Bojarczuk, Elena Dulskyte ,Inna Dvortsova ,Johann George, Natalija Gucevska, Mark Harman, ShanHe, Ralf Lämmel, ErikMeijer, Silvia Sapora, and Justin Spahr -Summers∗FacebookInc.

Conference :

This paper appears in Proceedings of 28th International Conference on Program Comprehension,ICPC2020.

INTRODUCTION :

Managing software asset ownership in any organization is important.In this paper, when we refer to 'asset' we include entities as diverse as source-code files, tables in the data warehouse, and software configurations. When we refer to the 'owner' of an asset, we mean this term in a broad sense: a set of people who take responsibility for the asset. Standard

processes, e.g., based on escalation, are typically in place to rule out unowned assets, as they would clearly be a cause for concern. A more nuanced question is the one of 'ownership health', i.e., whether each asset is attributed to the 'most suitable' owner.Who is the most suitable owner of a given asset changes over time, e.g., due to reorganization and individual function changes. Attributing assets to owners and measuring ownership health encompasses a combination of static and dynamic properties of the software assets themselves, the workflows for developing and managing the assets, and the structures of the organization that possesses the assets.

Methodology :

Vocabulary of Ownership Management The term resource alludes to such an element that is a piece of a framework or is controlled by an organization of intrigue.Accept an extraordinary piece of a framework: its resource for proprietor attribution planning or just attribution, which maps advantages for proprietor competitors, which are in this way alluded to as proprietors.The dark bolts on the left express that the resource for proprietor attribution planning is mostly encoded in the benefits themselves, for example, by explanation inside documents or a metastore for tables, in

which case extraction can be applied to resources or conceivably to logs that record the proprietors 'in real life'.

The quantity of oncall turns is under 10k. Number of proprietor possibility for a given resource a. It is frequently conceivable, subject to heuristics dependent on key highlights, to limit to a short rundown of proprietor applicants that are at all conceivable for the benefit an and that should be positioned in this manner.

The day by day stir for source-code documents in one of the greater mono repos of Facebook is around 100k. Every day proprietor beat for resource type t. Such stir is important as heuristics/ML should computerize these changes.

For a significant resource type for planned pipelines in the information distribution center with about 100k resources, the totaled day by day proprietor beat in the course of the most recent 4 months is about 10k while there were a few days with a few several influenced resources - this is a result of organized endeavors on possession the executives at Facebook, in light of Ownesty or something else.

3.1 Heterogeneity of Owned Assets Ownership suggestion - particularly when concentrating on code resources - is identified with code origin attribution , as applicable, for instance, for identifying malevolent or copied code, subject to stylometry techniques and the all-encompassing presumption of unmistakable composing style to be seen as a unique mark.Program cutting , idea task , and search-based

advancement , just as numerous different examination strategies, have all been utilized to explore auxiliary parts of a product advantage for help software engineers' perception of the benefit.Existing reliance investigations should be additionally summed up to apply better to heterogeneous resources and the issue of ascribing advantages for proprietors.

Provenance or ancestry might be focused on conditions for information resources while data stream might be focused on program resources, yet mixes or speculations of such techniques are expected to cover the advantage types that happen together by and by Workflow and Organizational Aspects Attribution of resources for proprietors likewise needs to consider communications of proprietor competitors with the benefits.

Obviously, possession proposal requires a speculation of the investigation of connections to represent the various sorts of advantages and proprietor competitors and differing types of communication. To be more concrete, hierarchical structure may bolster better comprehension of possession in that, for instance, the wellbeing of a specific attribution of a resource for a proprietor may rely upon past, later, and up and coming changes to groups or individual jobs.

The accompanying space explicit requirements challenge the arrangement of interpretable and reasonable models for possession suggestion; devoted exploration is required in this

way: The attribution connection among resources and proprietor applicants might be naturally uncertain.

Result :

This paper describes the overall idea of possession the board and the particular parts of utilizing proprietorship proposal for ascribing advantages for proprietors and estimating the wellbeing of any such attribution for huge and complex activities and frameworks. The suggestion of reasonable proprietors and the evaluation of possession wellbeing depends on information removed from resources, work processes and authoritative structures.

They have acquainted the Facebook Ownesty framework with delineate the pragmatic modern significance of the going with proprietorship research plan.

We additionally set out open issues and challenges and their connections to existing examination exercises and networks.