Sprint1

Hong Xin

Starting...

1. What is the goal?
    a. Build static codes analyzers that finds bugs for different types of Android apps

Starting...

1. What is the goal?
    a. Build static codes analyzers that finds bugs for different types of Android apps
2. Why do we care?
    a. automatically vs manually
    b. Customize analyzer based on different types of Android apps

Literature Review

1. Mokhov, Serguei A., et al. “The Use of NLP Techniques in Static Code Analysis to Detect Weaknesses and Vulnerabilities.” *Advances in Artificial Intelligence*, 2014, pp. 326–332., https://doi.org/10.1007/978-3-319-06483-3_33.
2. Koc, Ugur, et al. “Learning a Classifier for False Positive Error Reports Emitted by Static Code Analysis Tools.” *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2017, https://doi.org/10.1145/3088525.3088675.
3. Yuksel, Ulas, and Hasan Istanbul. “Automated Classification of Static Code Analysis Alerts: A Case StudyUlas.” *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/6676950?casa_token=R0ISduUXE6QAAAAA%3AIHh0SaNNLt7DM0gJ-CIE8Ub1BHM4XkBuxdOV3V49N45coxwnAbv5GH1N-92RoLKi8jenCYU.
4. Sayfullina, Luiza. “The Impact of Classifier Configuration and Classifier Combination on Bug Localization.” *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/6520844?casa_token=k3I1u19_QbEAAAAA%3AojLcRxP78VNdID_1fF0Q-vIEcWlPTwHkolSbFA-EAyPkSAwZG8jWuQjc0uaJ7reBafExKQs.
5. Alikhashashneh, Enas, et al. “Using Software Engineering Metrics to Evaluate the Quality of Static Code Analysis Tools.” *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/8367641?casa_token=ryK5GA8rT2MAAAAA%3AuTbDRYOc5HvhMPVbnmlUp-T1WzZoUXJHdIHJ2Nx-Q3c6XRizo9EhAkrjvxQGf2c7Mmvcb0k.

User stories and MVP

User Stories

1. Anyone who want to exam codes from an Android app

MVP

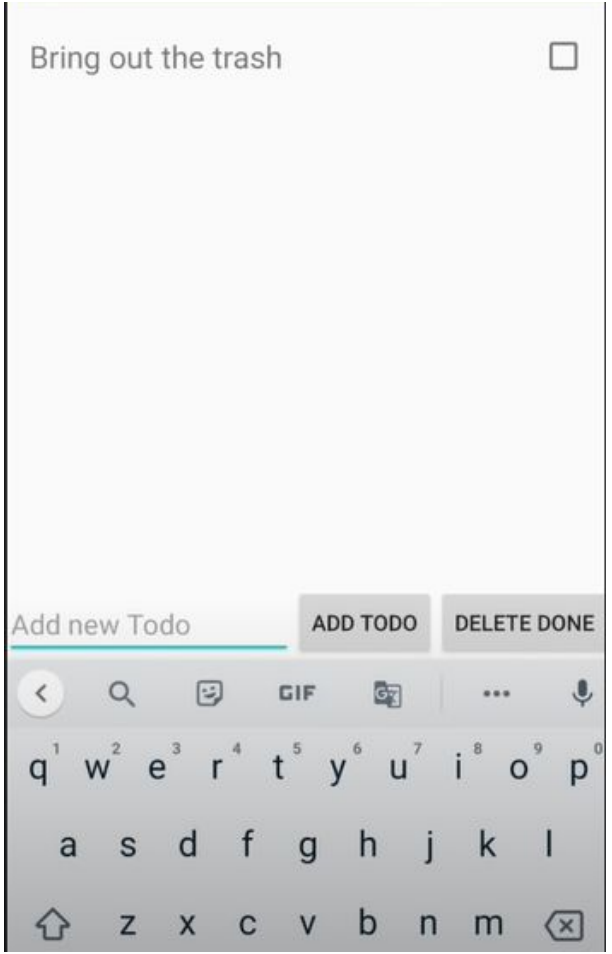
1. Classifier reads through codes (embeddings or a text file)
2. Classifier marks the errors or bugs in the codes

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio

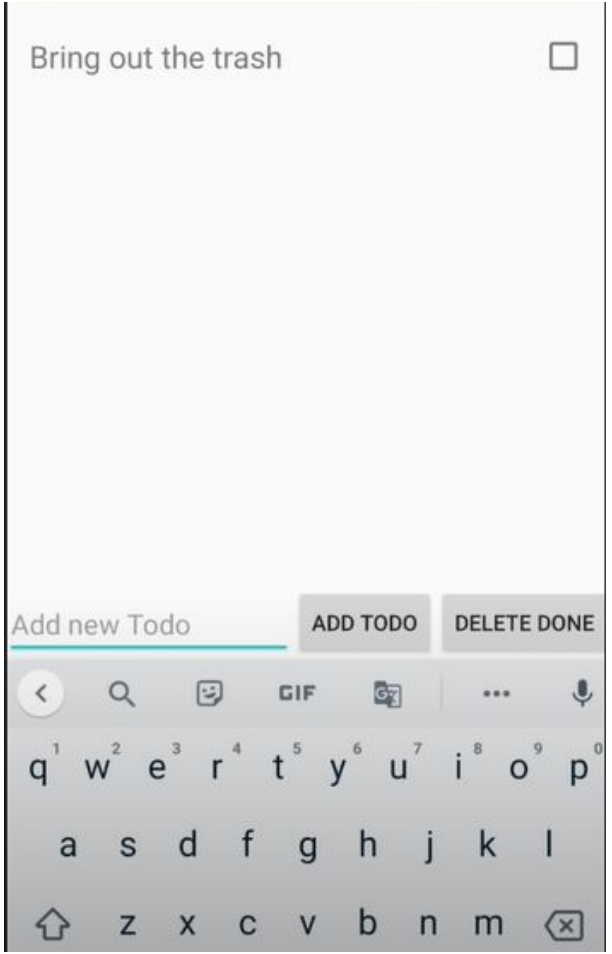
Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
    b. To-Do App



Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
    b. To-Do App
        i. Add new ToDo
        ii. Cross out ToDo
        iii. Delete ToDo



Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
        i. Kotlin: a programming language that interoperates with Java
        ii. Android Studio: development environment

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
        i. A static analysis tool
        ii. analyze and flag the code that breaks rules
        iii. Complexity reports based on...
            1. complexity of the program
            2. amount of code smells

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
3. Phase1 : Build ML classifiers that measure the accuracy of Detekt

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
3. Phase1 : Build ML classifiers that measure the accuracy of Detekt
4. Extract feature vectors from Android apps with and without bugs

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
3. Phase1 : Build ML classifiers that measure the accuracy of Detekt
4. Extract feature vectors from Android apps with and without bugs
    a. Possibly creating an error generator
5. Phrase 2: Use vectors to train classifiers that distinguishes error from Android apps

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
3. Phase1 : Build ML classifiers that measure the accuracy of Detekt
4. Extract feature vectors from Android apps with and without bugs
    a. Possibly creating an error generator
5. Phrase 2: Use vectors to train classifiers that distinguishes error from Android apps
    a. Evaluate the performance and compare to Detekt
    b. Pick the classifier with the highest accuracy

Logistics

1. Build a simple Android app from scratch
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers on the source codes
    a. Detekt
3. Phase1 : Build ML classifiers that measure the accuracy of Detekt
4. Extract feature vectors from Android apps with and without bugs
    a. Possibly creating an error generator
5. Phrase 2: Use vectors to train classifiers that distinguishes error from Android apps
    a. Evaluate the performance and compare to Detekt
    b. Pick the classifier with the highest accuracy
6. Build a different type of Android app, then repeat the above process

Questions?