Sprint1

Hong Xin

Product

Goal: Build a classifier that distinguishes errors from Android apps

Product

Goal: Build a classifier that distinguishes errors from Android apps

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.

Product

Goal: Build a classifier that distinguishes errors from Android apps

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.

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.

Product

Goal: Build a classifier that distinguishes errors from Android apps

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.

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.

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.

Product

Goal: Build a classifier that distinguishes errors from Android apps

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.

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.

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.

Marcilio, Diego, et al. “SpongeBugs: Automatically Generating Fix Suggestions in Response to Static Code Analysis Warnings.” *Journal of Systems and Software*, Elsevier, 1 June 2020, https://www.sciencedirect.com/science/article/pii/S016412122030128X?casa_token=QHXh0DkbDzcAAAAA%3AAkAjF6S_LjPcI6WDvbbuUMYRJuo0kuIRLp8QRFjGg2JqsBc3-oI7BIMJMT0zmnNGQKS2TuSa.

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
   a. Kotlin and Android Studio

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
   a. Kotlin and Android Studio
2. Implement existing static Android app analyzers
   a. Detekt

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers
    a. Detekt
3. Extract feature vectors from good and bad Android Apps

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
   a. Kotlin and Android Studio
2. Implement existing static Android app analyzers
   a. Detekt
3. Extract feature vectors from good and bad Android Apps
4. Extra log files from good and bad Android Apps
   a. Drain3

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers
    a. Detekt
3. Extract feature vectors from good and bad Android Apps
4. Extra log files from good and bad Android Apps
    a. Drain3
5. Train a machine learning classifier

Product

Goal: Build a classifier that distinguishes errors from Android apps

1. Build an Android app
    a. Kotlin and Android Studio
2. Implement existing static Android app analyzers
    a. Detekt
3. Extract feature vectors from good and bad Android Apps
4. Extra log files from good and bad Android Apps
    a. Drain3
5. Train a machine learning classifier
6. Evaluate the performance

User stories and MVP

User Stories

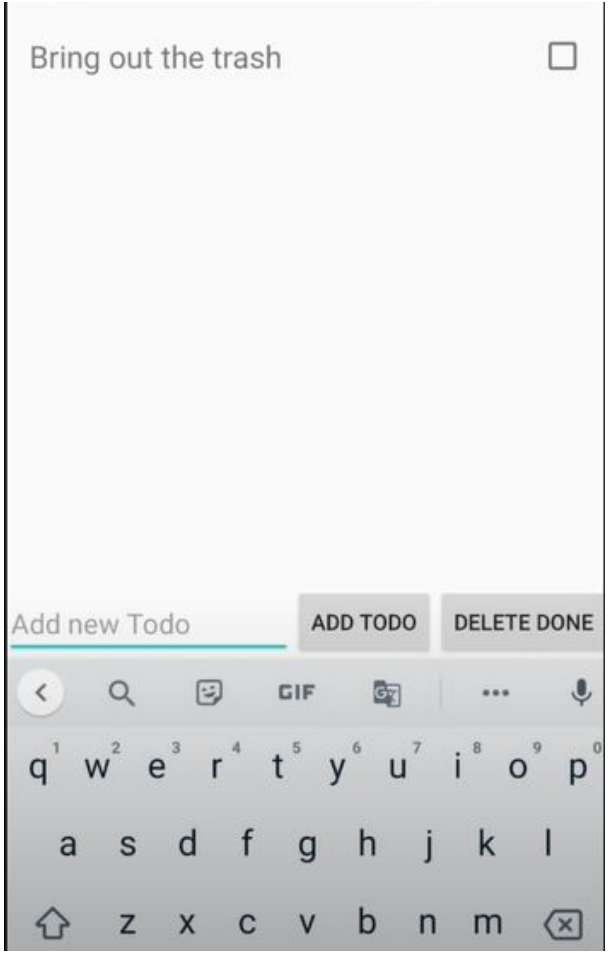
1. Engineers and developers who want to exam codes in an Android app project

MVP

1. Classifier reads through codes in a text file
2. Classifier marks the errors and/or vulnerabilities in the codes
3. Classifier outputs a log file

Android App

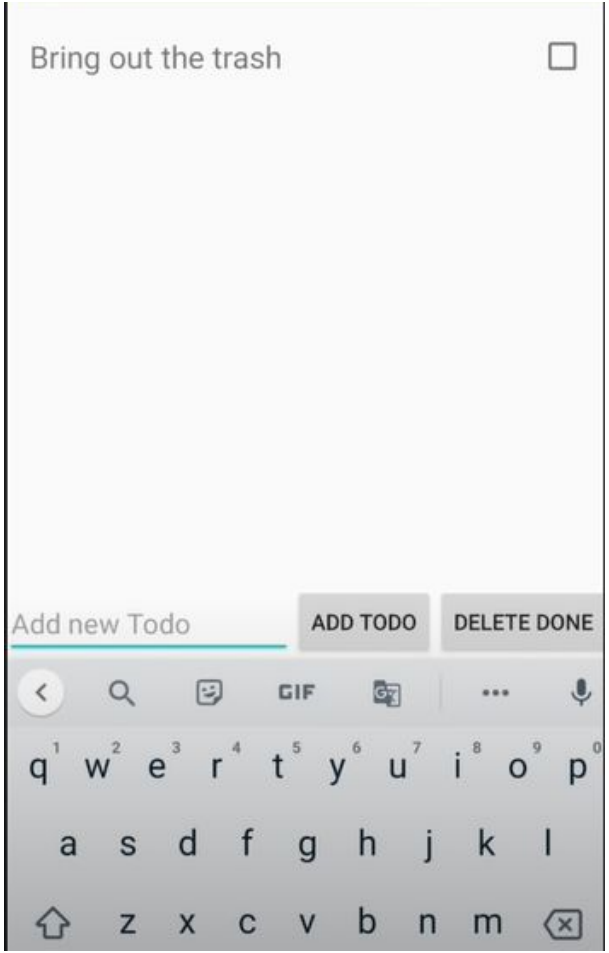
1. To-Do list



Bring out the trash □

Android App

1. To-Do list
2. User can ...
   a. Add new ToDo
   b. Cross out ToDo
   c. Delete ToDo



Bring out the trash □

Add new Todo

Tools

Kotlin

1. A cross-platform, statically typed, general-purpose programming language with type inference
2. Kotlin interoperates fully with Java

Tools

Android Studio

1. The official integrated development environment for Google's Android operating system

Tools

Detekt

1. A static analysis tool for the Kotlin programming language
   a. analyze Kotlin code with multiple rule sets
   b. flag the code that breaks any of its rules

Tools

Detekt

1. A static analysis tool for the Kotlin programming language
   a. analyze Kotlin code with multiple rule sets
   b. flag the code that breaks any of its rules
2. Complexity reports based on...
   a. lines of code
   b. complexity of the programand
   c. amount of code smells

Tools

Drain3

1. Open source log processing tool
2. Creates a balanced tree-like structure
   a. each leaf is based on the length of a log

Questions?