# Sprint2

Hong Xin

# Starting...

1. What is the goal?
    a. Build static code analyzers that identifies 7 types of bugs for Android apps

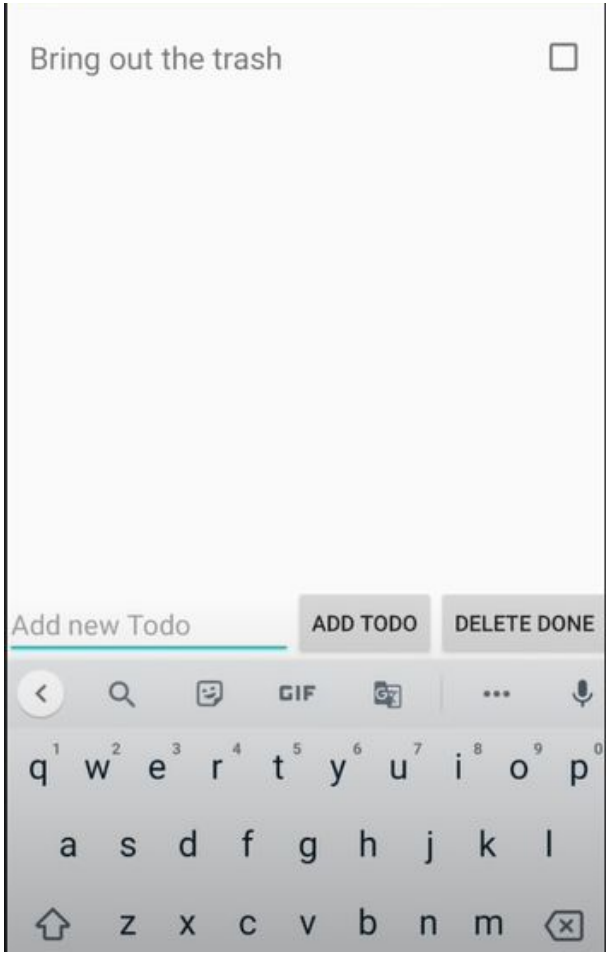
# Starting...

1. What is the goal?
    a. Build static code analyzers that identifies 7 types of bugs for Android apps
        i. Runtime errors.
        ii. Logic errors.
        iii. Compilation errors.
        iv. Syntax errors.
        v. Interface errors.
        vi. Resource errors.
        vii. Arithmetic errors.

# Starting...

1. What is the goal?
    a. Build static code analyzers that identifies 7 types of bugs for Android apps
        i. Runtime errors.
        ii. Logic errors.
        iii. Compilation errors.
        iv. Syntax errors.
        v. Interface errors.
        vi. Resource errors.
        vii. Arithmetic errors.
2. Why do we care?
    a. automatically vs manually
    b. Customize analyzer based on different types of Android apps

1. The ToDoList app is done
2. I am in the process of training model



# Definition of Architecture

1. Phase 1: Build an Android App and use ML classifiers to measure the accuracy of Detekt library

# Definition of Architecture

1. Phase 1: Build an Android App and use ML classifiers to measure the accuracy of Detekt library
2. Phase 2: Use feature vectors to train different types of classifiers and benchmarking their performance to Detekt's performance. Then, select the best classifier

# Definition of Architecture

1. Phase 1: Build an Android App and use ML classifiers to measure the accuracy of Detekt library
2. Phase 2: Use feature vectors to train different types of classifiers and benchmarking their performance to Detekt's performance. Then, select the best classifier
3. Phase 3: Find source codes of a complicated app or build a complicated app and further train those classifiers based on 7 types of most common errors

# Definition of Architecture

1. Phase 1: Build an Android App and use ML classifiers to measure the accuracy of Detekt library
2. Phase 2: Use feature vectors to train different types of classifiers and benchmarking their performance to Detekt's performance. Then, select the best classifier
3. Phase 3: Find source codes of a complicated app or build a complicated app and further train those classifiers based on 7 types of most common errors
4. Phase 4: A static code analyzers that identifies 7 types of programing errors for Android apps

# Technology Selection

1. Development Environment: Android Studio
2. Coding Language: Kotlin
3. Library (Benchmark): Detekt

# Technology Selection

Classifiers:

1. Logistic Regression
2. Naïve Bayes
3. K-Nearest Neighbours
4. Random Forest

# Questions?