

# Testing Policy

Binary Ninjaz

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Purpose</b>	<b>3</b>
<b>3</b>	<b>Testing Tools</b>	<b>3</b>
3.1	XCode . . . . .	3
3.2	Android Studio . . . . .	3
3.3	NPM . . . . .	3
<b>4</b>	<b>Testing Philosohpy</b>	<b>4</b>
4.1	Benefits . . . . .	4
<b>5</b>	<b>Test Evaluation</b>	<b>4</b>

# **1 Introduction**

This document details the policies that BinaryNinjaz will be using to guide development and ensure a quality product. We describe why we intend to use specific testing methods and how their effects will be impactful in our process.

## **2 Purpose**

Automated testing is vital and ensuring that the number of bugs in a program is as limited as possible. While having automated testing does not guarantee there will be no bugs in a program it can greatly reduce any bugs from appearing when you might not expect them to. While we intend to have a comprehensive test suite, we will not accept that wholly passing test cases mean that a program is without flaw.

## **3 Testing Tools**

### **3.1 Android Studio**

The JUnit framework will be used for our automated testing purposes. To run tests one will need Android Studio. To execute the tests one can right click on the tests directory and select 'Run tests.'

### **3.2 NPM**

The Ava framework will be used for automated testing purposes. To run tests one will need to run "npm test" from the testing directory.

### **3.3 XCode**

The XCTest framework will be used for our automated testing purposes. To run tests one will need XCode. To execute the tests one will need to use XCode's build command.

## 4 Testing Philosophy

Implementation of both simple and holistic tests will be of concern. However, holistic testing will be of more interest. Testing workflows are of vital importance. The project does not have many simple tasks that could provide us with the safety guarantees we would like. We instead seek to find safety in implementing tests that check the whole interaction of these few unit cases.

### 4.1 Benefits

Automated test provide developers with the ability to have safe of mind that they have not made any changes that break the program in any way or introduce bugs in any part of the program. Forcing developers to write tests forces them to check every part of code they write making tests a form of documentation of a projects requirements.

## 5 Test Evaluation

Strict zero test case failure will be followed. Any branch with a failing test case will never be merged into master. Hence we will ensure that every branch runs tests before a pull request.