

# Menu Tingz


## Testing Report

Group: K.O.D Inc  
Members: Ottor Mills  
Kenneth Anglin  
Nicoy Smith  
David Thomas

An abstract graphic in the bottom right corner of the slide. It consists of several overlapping, semi-transparent geometric shapes, primarily cubes and rectangular prisms, in various shades of green and teal. Some shapes have thin white outlines, giving them a wireframe-like appearance. The overall effect is a modern, architectural feel.

# Software Testing

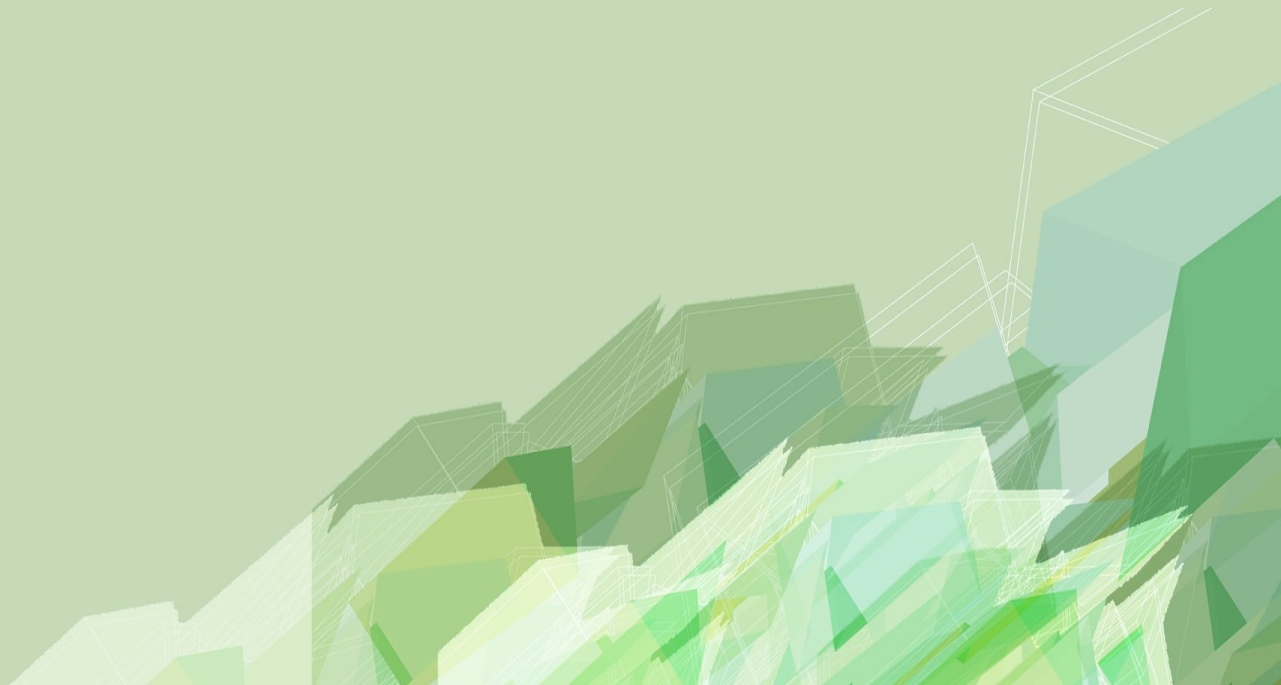
Software testing is a crucial aspect of the software development life cycle. It is said that no software is error free. Software testing reveals any vulnerabilities in an application's source code. Hackers make some of the best software testers.



# Tools Used

1. Calabash Testing Framework

2. JavaScript

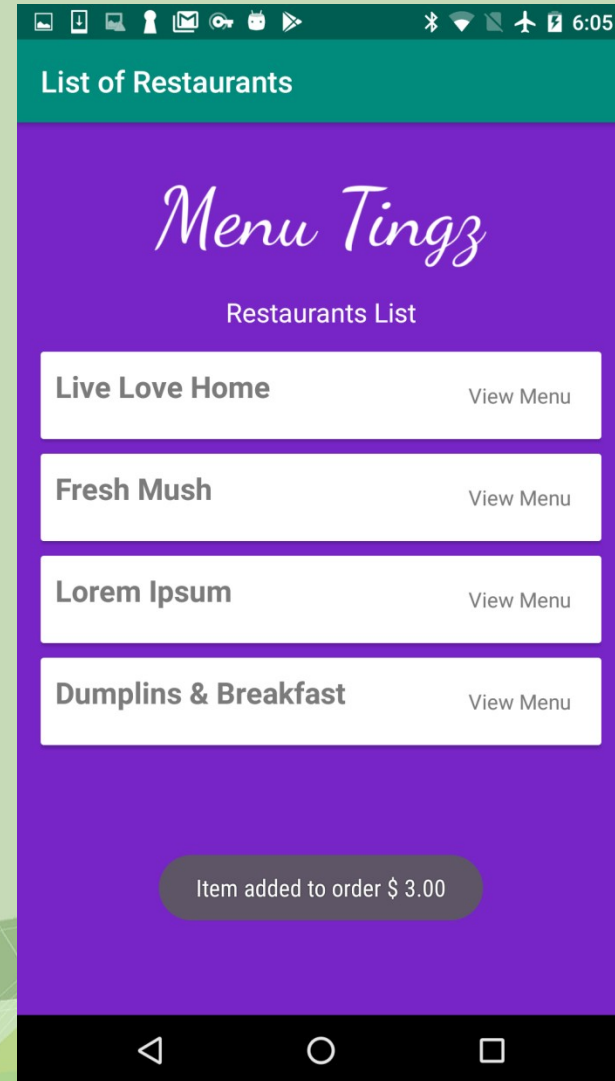
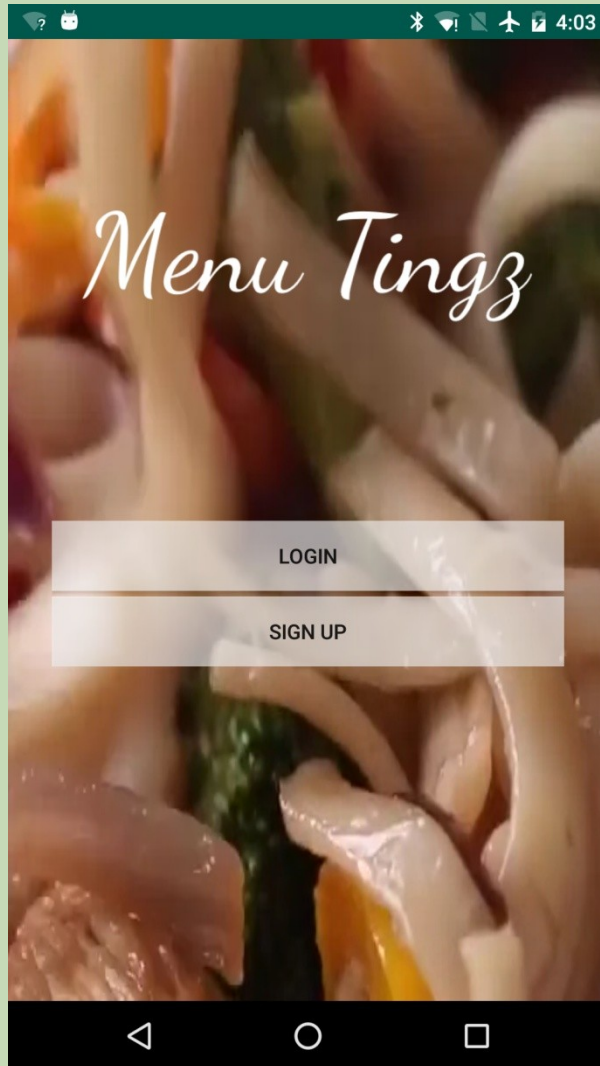


# What is Menu Tingz

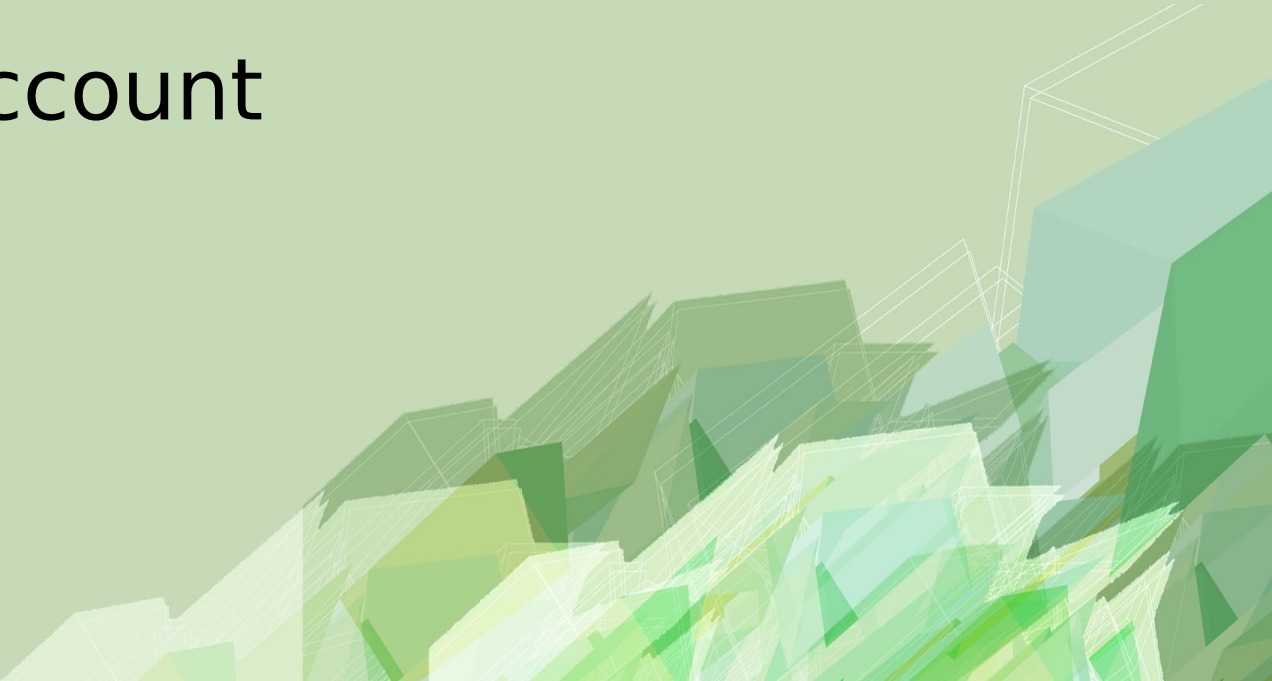
Menu Tingz is an android application made by BajanTech which allows users view and order dishes from various restaurants listed within the application.



# User Interface



# Use Cases

- Adding dishes to tray
  - Removing dishes from tray
  - Placing orders
  - Creating Account
  - Logging In
- 
- The bottom right corner of the slide features an abstract geometric design. It consists of various overlapping polygons and wireframe structures in shades of green, teal, and light blue, creating a modern, architectural feel.

# Use Cases (Contd.)

The Menu Tingz app can successfully carry out each use case. After logging in the app has well defined event flows. (Not much testing required)

Small Domain

An abstract graphic in the bottom right corner of the slide. It consists of several overlapping, semi-transparent geometric shapes, primarily triangles and polygons, in various shades of green and teal. The shapes are arranged in a way that creates a sense of depth and complexity, resembling a stylized mountain range or a cluster of buildings. The overall effect is a modern, geometric aesthetic.



# Use Cases (Contd.)

On the other hand, the login and sign up forms is vulnerable to malformed inputs and as such automation is necessary.

Large Domain

An abstract graphic in the bottom right corner consisting of overlapping, semi-transparent geometric shapes in various shades of green and teal. The shapes include rectangles, triangles, and polygons, some with thin white outlines, creating a layered, 3D effect.



# Methodology

Test: Black Box Vulnerability Testing

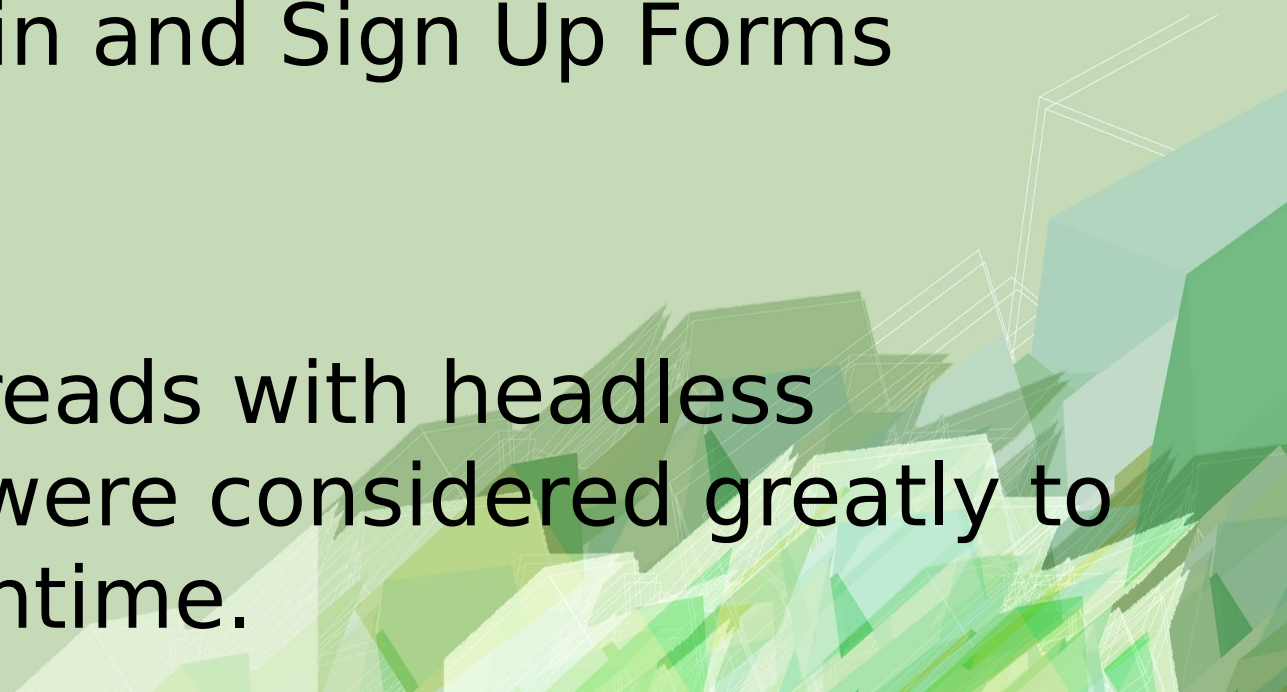
Method: Bruteforcing

Testing Domain: Very Large

Order of Growth: Very Large


Target: Login and Sign Up Forms

Multiple threads with headless emulators were considered greatly to improve runtime.



# Vulnerability Testing (Contd.)

This test consisted of a feature file in combination with a JavaScript program. The feature file acted as a template and the JavaScript program made changes to this file to carry out each test case. Once changes were made to the file it was executed using a subprocess.



# Sample Output

```
When I see "My Orders" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "REMOVE" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
# NOW TESTING ORDER PLACEMENT
When I see "MAKE ORDER" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "MAKE ORDER" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
When I see "Are you sure you want to place this order" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "YES" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
```

```
Scenario: checking out with an empty cart # features/my_first.feature:97
# It is observed that no error is thrown when checking out with an empty cart
When I see "Live Love Home" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "Live Love Home" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
Then I should see "Live Love Home" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:9
When I see "Pasta with Chicken" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "Pasta with Chicken" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
When I see "My Orders" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "REMOVE" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
Then I touch the "MAKE ORDER" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
When I see "Are you sure you want to place this order" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I touch the "YES" text # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:25
```

6 scenarios (6 passed)  
70 steps (70 passed)  
1m26.434s

NOW TESTING CHARACTER SET: !"#\$%&'()\*+,-./

Number of Combinations: 58650

Testing:  
Length: 1

Feature: Form Vulnerability Test

Scenario: As a new user I should be able to create a new account # ./features/form\_field\_test.feature:4

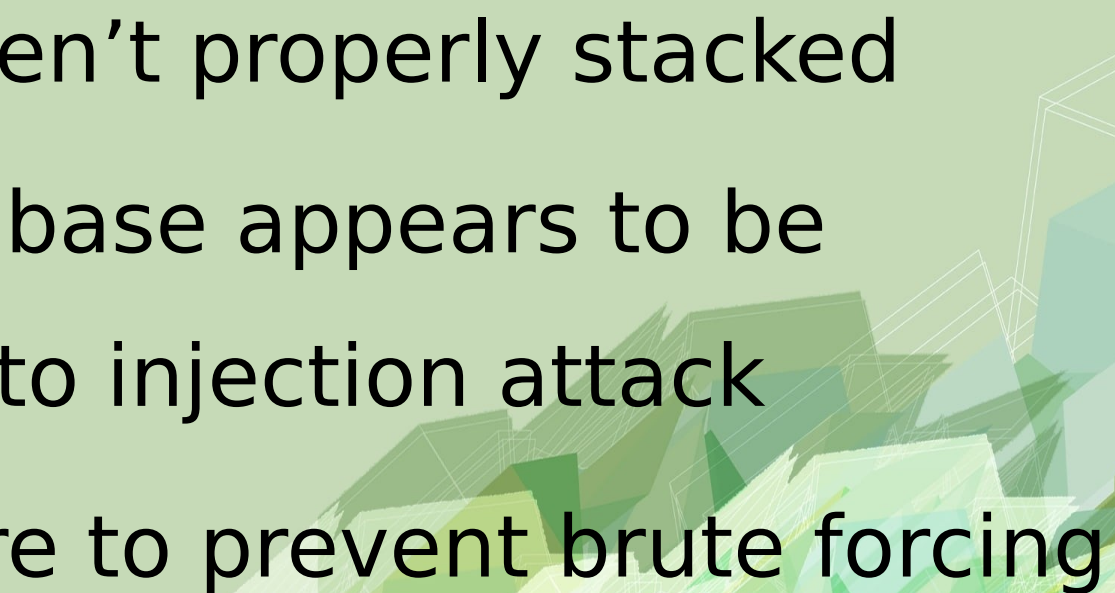
```
When I see "Sign Up" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
Then I press "Sign Up" # calabash-android-0.9.8/lib/calabash-android/steps/press_button_steps.rb:17
When I see "Username" # calabash-android-0.9.8/lib/calabash-android/steps/assert_steps.rb:5
```

# Demonstration

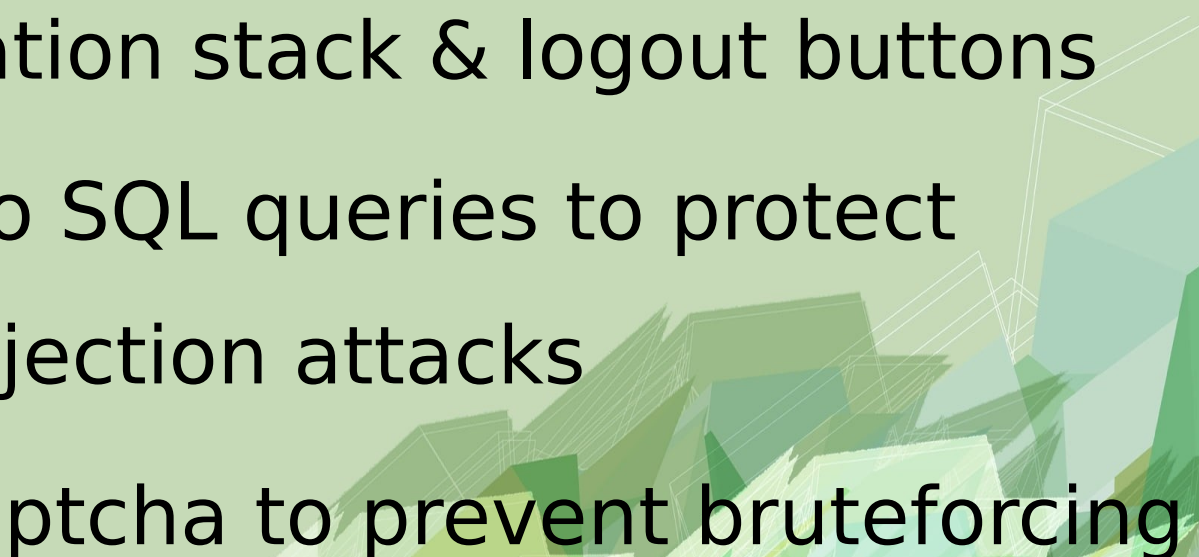
*(Attachment file not found)*



# Noteworthy Discoveries

- App allows for weak usernames and passwords
  - No max length of the form fields
  - Screens aren't properly stacked
  - Sqlite database appears to be vulnerable to injection attack
  - No measure to prevent brute forcing
- 

# Application Improvements

- Setting a minimum length for passwords and usernames
  - Setting a max length of the form fields
  - Fix the navigation stack & logout buttons
  - Add slashes to SQL queries to protect against SQL injection attacks
  - Implement captcha to prevent bruteforcing
- 



# Alternative App Testing Tools

1. Espresso
2. Flutter Testing Package
3. WebdriverIO + Appium





# The End

## Any Questions?

