

# Testing Automation

of the

Mobile App Control Centre – Malaria  
(MACC-MAL)

Google Summer of Code 2015

Daisy Nkweteyim

# Team Members



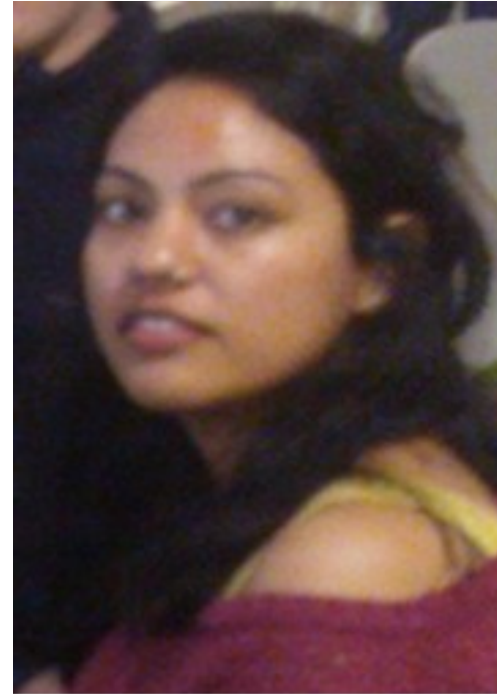
Daisy Nkweteyim

Department of Computer Engineering

University of Buea

Buea, Cameroon(UTC+1)

(Student)



Mariam Kiran, PhD

Department of Computer Science

University of Sheffield

Sheffield, UK (BST)

(Mentor)

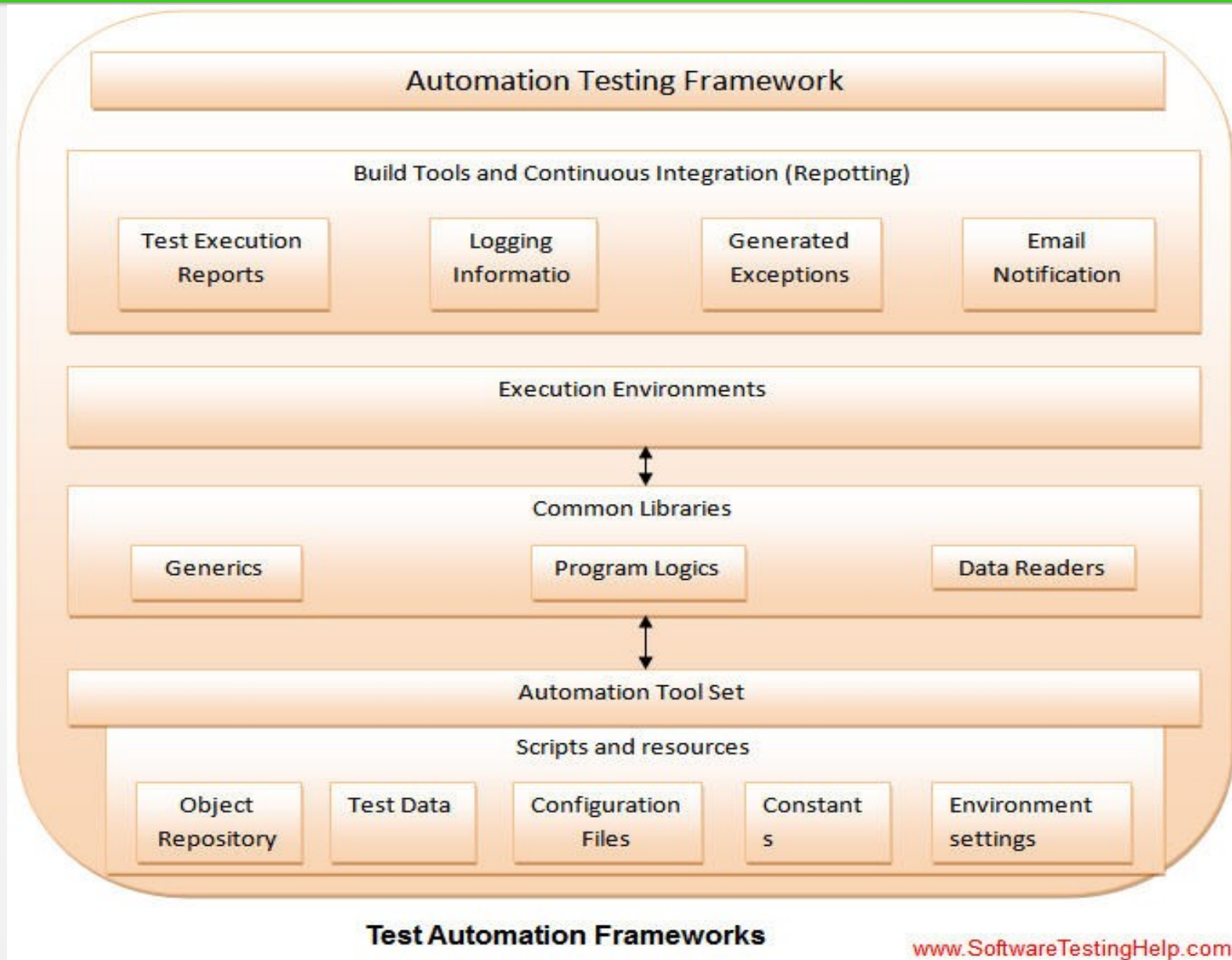
# Introduction

- Four mobile apps (2 Android, 2 IOS), one web app (MACC)
- MACC is the web version of the mobile apps (malaria and peacetrack)
- I tested the malaria part of the Mobile App Control Centre (MACC)

# Project Overview and Deliverables

- Build an automated test framework for regression testing
  - Used the data driven test framework approach
- Test the MACC in conjunction with its database
- Documentation of the system

# Automation Test Framework



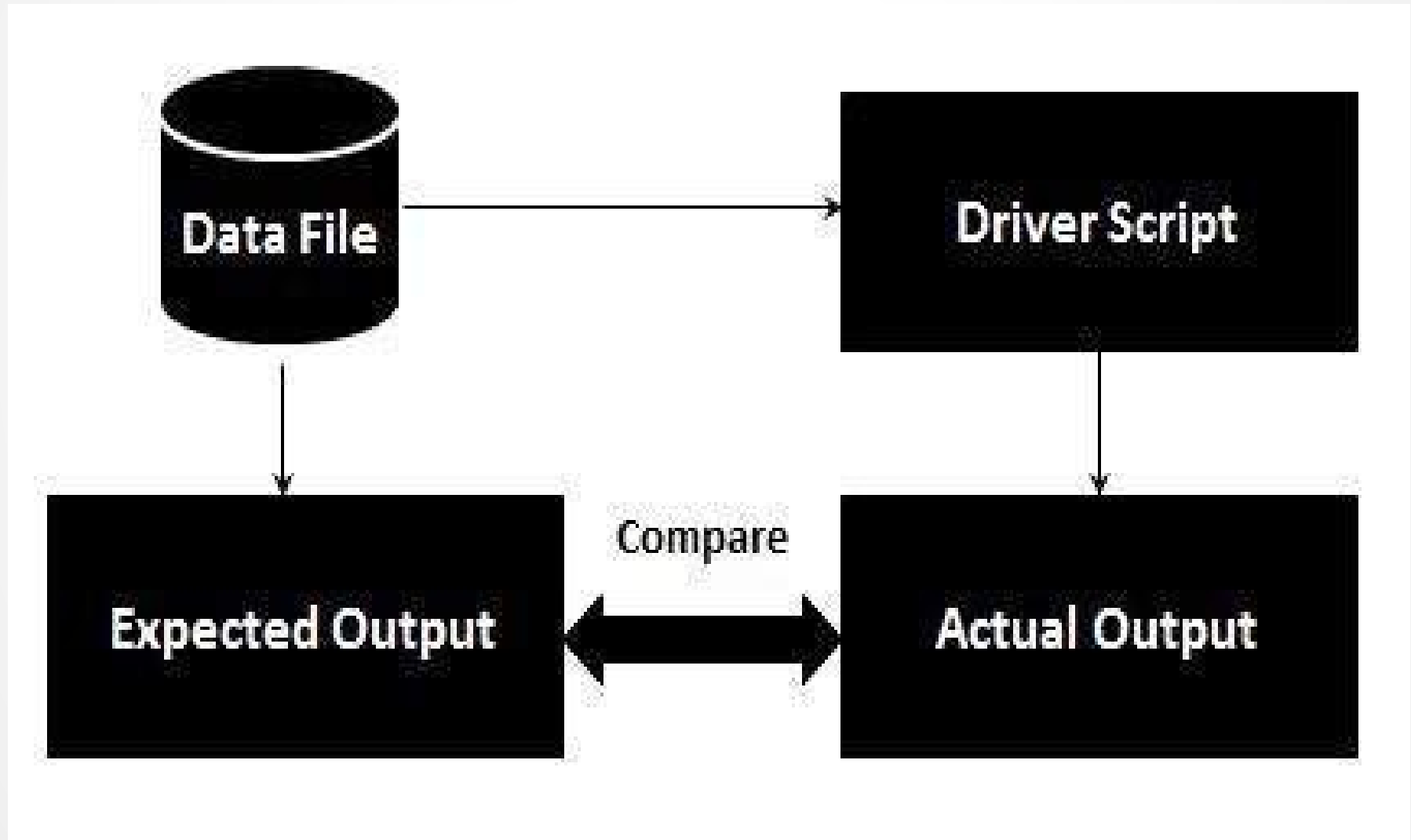
# Project Overview and Deliverables

- Build an automated test framework for regression testing
  - Used the data driven test framework approach
- Test the MACC in conjunction with its database
- Documentation of the system

# Data Driven Test Framework

- Test data kept separate from test scripts
- Test data kept in excel sheets (XSSF format)
- Apache POI reads data from excel sheet
- TestNG @DataProvider provides data from excel to the test method

# Data Driven Test Framework Cont.





# Data Driven Test Framework Cont.

	A	B
1	<u>Username</u>	<u>Password</u>
2	admin	<u>mypassword</u>
3	<u>daisyab</u>	<u>mypassword</u>
4	daisy	password
5	hello	hi
6		

# Data Driven Test Framework

The screenshot shows an IDE window titled "Results of running class Login". The top toolbar includes icons for Problems, Javadoc, Declaration, Search, Console, History, and Synchronize. A red progress bar at the top indicates "Tests: 1/1 Methods: 4 (67272 ms)". Below this is a search bar and a summary: "Passed: 1 Failed: 3 Skipped: 0". The test results are organized into a tree view with tabs for "All Tests", "Failed Tests", and "Summary". The tree shows a "Default suite (1/3/0/0) (13.391 s)" which contains a "Default test (13.391 s)" and a "home\_page\_features.Login" test. The "home\_page\_features.Login" test is expanded, showing four sub-tests: "login (13.391 s)". The "login" test is further expanded, showing four data-driven test cases: "admin", "mypassword" (6.9 s), "daisyab", "mypassword" (2.263 s), "daisy", "password" (1.906 s), and "hello", "hi" (2.322 s). Each test case has a small icon indicating its status (green checkmark for passed, red X for failed).

Problems Javadoc Declaration Search Console Results of running class Login History Synchronize

Tests: 1/1 Methods: 4 (67272 ms)

Search:

Passed: 1 Failed: 3 Skipped: 0

All Tests Failed Tests Summary

Default suite (1/3/0/0) (13.391 s) Failure Exception

Default test (13.391 s)

home\_page\_features.Login

login (13.391 s)

- "admin", "mypassword" (6.9 s)
- "daisyab", "mypassword" (2.263 s)
- "daisy", "password" (1.906 s)
- "hello", "hi" (2.322 s)

# Project Overview and Deliverables

- Build an automated test framework for regression testing
  - Used the data driven test framework approach
- Test the MACC in conjunction with its database
- Documentation of the system

# Database Testing

- Selenium used to automate tests for UI only
- Java Database Connectivity API (JDBC)
- Perform CRUD operations

# Project Overview and Deliverables

- Build an automated test framework for regression testing
  - Used the data driven test framework approach
- Test the MACC in conjunction with its database
- Documentation of the system

# Timeline

<https://docs.google.com/document/d/1Tfq7yocOFXniEpFi-4mSatC6a8PymhC8ywG6mmXsY-E/edit?usp=sharing>

# Difficulties

- Setting up environment
- Figuring out how to access information from excel sheets
- Getting postgresql driver to connect to database

# Successes

- Learnt new technologies (how to use selenium for automated testing) and coding styles
- Learnt more about version control and how to keep track of the project
- Covered a good portion of my timeline (currently finishing what is left)
- Had a good experience developing the project
- Would like to continue after the Google Summer of Code ends



# Future Plans (Next Steps)

- Cover the Peacetrack part of the testing
- Refine database tests
- Configure a continuous integration tool.

Thank You

Q&A