

## BÀI TẬP 2

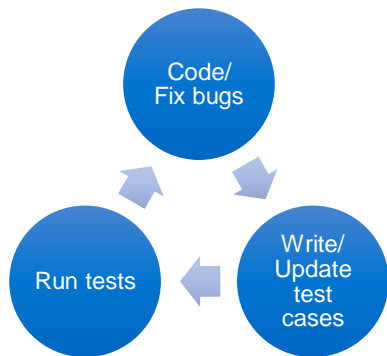
### 1 QUI ĐỊNH NỘI BÀI

- Hạn chót nộp bài: **29/5, 23:55**
- Tổng điểm: 100 (+10 điểm bonus)

### 2 YÊU CẦU

Đây là bài tập làm theo nhóm đã đăng ký cho seminar. Mỗi nhóm sẽ thực hiện các công việc kiểm thử tự động cho một ứng dụng Web bằng việc sử dụng Katalon Studio (<https://www.katalon.com>).

Bài tập này giả lập quá trình thực hiện kiểm thử tự động như trong thực tế các dự án phần mềm. Việc kiểm thử được thực hiện song song với quá trình phát triển mới, chỉnh sửa, và sửa lỗi trong chức năng của ứng dụng phần mềm cần kiểm thử (AUT). Ứng dụng AUT trong bài tập này là Mattermost Web App (<https://github.com/mattermost/mattermost-webapp>). Các công việc kiểm thử được thực hiện liên tục và lặp lại như hình bên dưới:



#### **Yêu cầu 1:** Cài đặt ứng dụng và cấu hình

Cài đặt ứng dụng Web cần kiểm thử Mattermost (<https://github.com/mattermost/mattermost-webapp>). Cài ứng dụng Web này trên local hoặc trên một domain/host nào đó dùng chung cho cả nhóm. Nếu cài trên máy local thì tất cả các thành viên cùng dùng chung một URL (ví dụ <http://localhost:8065>).

- Tạo một tài khoản *git* để chia sẻ mã nguồn/script và Katalon test project giữa các thành viên trong nhóm.
- Cài đặt Katalon Studio (<https://www.katalon.com>).
- Tạo test project trên Katalon Studio và cấu hình Katalon để kết nối với *git*. Tất cả tập tin trong test project này được lưu trữ trong tài khoản *git* của nhóm. (Các bạn cần phải biết sử dụng *git*)

**Yêu cầu 2:** Viết **50 test case**. Các test case cần thỏa mãn các điều kiện sau:

- Kiểm thử các chức năng thông qua giao diện Web của Mattermost.
- Có ít nhất **20** data-driven test case.
  - **5** test case sử dụng Data Type là CSV và **15** test case còn lại sử dụng Internal Data.
  - Mỗi data file có ít nhất **5** dòng dữ liệu.
- Mỗi test case cần kiểm tra kết quả của ít nhất một bước là đúng hay sai. Ví dụ, đối với test case kiểm thử chức năng đăng nhập thì cần kiểm tra đăng nhập đúng hay sai.
- Có ít nhất 5 test case có sử dụng vòng lặp (for hoặc while).
- Có ít nhất 5 test case có sử dụng lệnh *if* (*if statement*).
- Có ít nhất 5 test case có sử dụng lệnh *case* (*case statement*).
- Có ít nhất 5 test case được gọi từ 5 test cases khác.
- Có ít nhất **5 API test case** để kiểm thử các API của Mattermost.

**Yêu cầu 3:** Chạy và chỉnh sửa giao chức năng hoặc giao diện Mattermost (ví dụ, thay đổi id, HTML type, label trên màn hình). Giả lập việc thực hiện kiểm thử và phát triển ứng dụng như trong thực tế. Chạy và chỉnh sửa **15 lần** các test cases được tạo ở Yêu cầu 2.

- Tạo một test suite có **20 test case** từ Yêu cầu 2. Đặt tên test suite này là “Run all test cases”. Chú ý thứ tự các test case.
- Iteration 1: Chạy test suite “Run all test cases”.
  - Đảm bảo tất cả các test case đều chạy thành công.
  - Vào thư mục “Reports/Run all test cases/”, tìm test report tương ứng với lần chạy này và thêm “Iteration1” vào trước tên report này (prefix). Ví dụ, tên report được tạo ra tự động là 20190506\_102048 thì đổi tên report này thành Iteration1\_20190506\_102048.
- Iteration 2:
  - Thêm **10 test case** vào test suit “Run all test cases”.

- Chỉnh sửa giao diện Web của Mattermost sao cho có ít nhất 5 test case không chạy được hay failed. Những test case này đã chạy thành công ở Iteration 1. (Test case có thể bị lỗi nếu màn hình có thêm các control mới, các control bị đổi tên, đổi label, v.v...)
- Chạy lại test suite “Run all test cases”.
- Đổi tên test report tương ứng với lần chạy này bằng cách thêm “Iteration2” vào trước tên test report được tạo tự động (như cách ở trên Iteration 1).
- Iteration 3:
  - Thêm **10 test case** vào “Run all test cases” (test suite này có 40 test case)
  - Xem test report từ Iteration 2, chỉnh sửa giao diện Web của Mattermost để sửa lỗi phát hiện từ kiểm thử tự động trong Iteration 2.
  - Thêm một số label và control vào các màn hình giống như thêm chức năng mới nhưng không cần thiết phải chỉnh sửa database.
  - Có thể có một số test cases bị lỗi. Các test cases bị lỗi này có thể khác với test cases lỗi ở Iteration 2.
  - Chạy lại test suite “Run all test cases”.
  - Đổi tên test report tương ứng với lần chạy này bằng cách thêm “Iteration3” vào tên test report được tạo tự động (như cách ở trên Iteration 2).
- Iteration 4 đến 14:
  - Test suite “Run all test cases” có tất cả **50 test case** (thêm 10 test case còn lại vào test suite từ Iteration 4).
  - Thực hiện tương tự như Iteration 3. Nhưng cần lưu ý:
    - Không cố định các test case có lỗi (không cho tất cả các test case lỗi liên tục trong các Iteration).
    - Có những test case không có lỗi nào trong tất cả các Iteration.
    - Một số Iteration không có lỗi nào cả.
- Iteration 15:
  - Thực hiện như Iteration 4 – 14, nhưng tất cả các test case đều thành công (passed).

### 3 HÌNH THỨC NỘP BÀI

1. Một tài liệu PDF tóm tắt kết bài tập này, đặt tên là <Hw1\_tom tat\_Mã nhóm.PDF>. Nội dung bao gồm:
  - a. Thông tin các thành viên trong nhóm.

- b. Thông tin tài khoản **git** để truy cập vào test project của nhóm.
  - c. URL của ứng dụng Mattermost (local hoặc một domain nào đó).
  - d. Tạo một bảng liệt kê danh sách các test case được tạo với các cột là loại test case (data-driven, UI hay API, test case có vòng lặp, if, case, gọi test case khác).
  - e. Mức độ hoàn thành các Yêu cầu: số test case được tạo, số iteration/số lần chạy.
  - f. Các thông tin ghi chú khác.
2. Nén thư mục test project trong Katalon Studio thành tập tin ZIP. Đặt tên file ZIP này trùng với tên test project.
  3. Tạo một file ZIP bao gồm file PDF (mục #1) và ZIP (mục #2). Đặt tên file này theo <Hw2\_Mã nhóm.zip>. Nộp file này trên Moodle. Nếu file này lớn hơn 10MB thì trong file tóm tắt (<Hw2\_tom tat\_Mã nhóm.PDF>) ghi link Google Drive đến file ZIP này, và nộp file tóm tắt trên Moodle.

## 4 TIÊU CHÍ CHẤM ĐIỂM

- Tất cả các Yêu cầu trên được **hoàn thành đúng theo quy định**: 100 điểm + 10 điểm bonus.
- Hoàn thành 50 test case và 15 lần chạy/Iteration nhưng các loại test case không như quy định: 80 - 90 điểm.
- Số test case trên 30 và số lần chạy trên 10: từ 50 điểm trở lên.
- Số test case dưới 25: 0 điểm.
- Số lần chạy/Iteration dưới 8: 0 điểm.
- Không có chỉnh sửa ứng dụng Mattermost trong các lần chạy: 0 điểm.

## Phụ lục

# Hướng dẫn cài đặt Mattermost

Đây là tài liệu hướng dẫn cài đặt Mattermost ở local nhằm mục đích chạy, kiểm thử ứng dụng Mattermost.

Tài liệu hướng dẫn trên môi trường MacOS nếu cài đặt trên môi trường khác có thể thay đổi nhưng không nhiều (nếu cần thì tham khảo trên <https://developers.mattermost.com/contribute/server/developer-setup/>)

## I. Chọn bản build

Xem danh sách bản build:

- Server: <https://github.com/mattermost/mattermost-server/releases>
- Webapp: <https://github.com/mattermost/mattermost-webapp/releases>

Nên chọn bản build của Server và Webapp cùng phiên bản. Trong tài liệu này sẽ sử dụng bản build: **v5.17.1**. Link tải sẽ là: [Server-v5.17.1](#) và [Webapp-v5.17.1](#)

Tạo thư mục mattermost ở Desktop, sau đó tải 2 bản build về và giải nén ta được 2 thư mục: **mattermost-server-5.17.1** và **mattermost-webapp-5.17.1**

## II. Cài đặt Server

**B1:** Cài đặt Docker: <https://docs.docker.com/docker-for-mac/>

**B2:** Tải và cài đặt homebrew: <https://brew.sh/>

**B3:** Cài đặt Go: **brew install go**, cài đặt biến môi trường:

```
export GOPATH=$HOME/go

export PATH=$PATH:$GOPATH/bin

ulimit -n 8096
```

**B4:** Kiểm tra phiên bản go: **go version**, do Mattermost yêu cầu phiên bản tối thiểu là 1.13+ nên cần cập nhật nếu đang chạy ở phiên bản cũ hơn.

**B5:** Mở terminal và di chuyển đến thư mục **mattermost-server-5.17.1**

**B6:** Chạy server:

```
make run-server
```

**B7:** Kiểm tra server đã được chạy chưa, mở trình duyệt và truy cập đường dẫn:

<http://localhost:8065/api/v4/system/ping>. Nếu có kết quả trả về giống bên dưới thì server đã chạy thành công.

```
{"AndroidLatestVersion":"","AndroidMinVersion":"","DesktopLatestVersion":"","DesktopMinVersion":"","IosLatestVersion":"","IosMinVersion":"","status":"OK"}
```

Giao diện web sẽ được host ở tại <http://localhost:8065/>. Do chưa build webapp nên hiện tại hệ thống sẽ trả về **404 page not found**.

### III. Cài đặt WebApp

**B1:** Cài đặt Nodejs: Node v10.15.3+ and npm v6.4.1+

```
brew install node@10 libpng
```

**B2:** Mở terminal và di chuyển đến thư mục **mattermost-webapp-5.17.1**

**B3:** Link thư mục build của webapp đến server

```
ln -nfs /Users/[user]/Desktop/mattermost/mattermost-webapp-5.17.1/dist  
/Users/[user]/Desktop/mattermost/mattermost-server-5.17.1/client
```

Chú ý đổi **[user]** tương ứng

**B3:** Cài đặt thư viện và test môi trường

```
make test
```

**B4:** Build webapp:

```
make build
```

#### IV. Tạo dữ liệu test

**B1:** Mở terminal và di chuyển đến thư mục **mattermost-server-5.17.1**

**B2:** Chạy lệnh tạo data test

```
make test-data
```

**B3:** Kiểm tra webapp đã hoạt động chưa:

- Truy cập đường dẫn <http://localhost:8065/>
- Nếu thấy trang đăng nhập tức đã chạy webapp thành công
- Kiểm tra đã tạo dữ liệu test thành công chưa bằng cách đăng nhập với tài khoản: username: **user-1**, password: **SampleUs@r-1** (Thông tin tài khoản được hiển thị sau khi chạy lệnh tạo dữ liệu test ở bước 2)
- Nếu đăng nhập thành công, tức dữ liệu test đã được khởi tạo.