**Testing Phase**

In our project we have used a Content Management System provided by Total.js. Total.js is a node-js framework hence all of its files are divided into modules. Each module contains the functions that are passed an object containing the present user state. While testing the code we implemented Black-Box testing on each individual module and White-Box testing on the internal functions.

**Unit Testing:**

**Module: controller/default.js**

1. Function: view\_newsfeed

Test Code:

it( ‘returns the newsfeed.html file using render module’ ){

var result\_url = yield.self.url;

var result\_fileName = yield.self.path;

expect(result\_url).to.eql(‘/newsfeed’);

expect(result\_fileName).to.eql(‘/themes/default/views/newsfeed.html’);

};

1. Function: view\_homepage

Test Code:

it( ‘returns the index.html file using render module’ ){

var result\_url = yield.self.url;

var result\_fileName = yield.self.path;

expect(result\_url).to.eql(‘/’);

expect(result\_fileName).to.eql(‘/themes/default/views/index.html’);

};

1. Function: view\_contact

Test Code:

it( ‘returns the contact.html file using render module’ ){

var result\_url = yield.self.url;

var result\_fileName = yield.self.path;

expect(result\_url).to.eql(‘/contact’);

expect(result\_fileName).to.eql(‘/themes/default/views/contact.html’);

};

1. Function: view\_page

Test Code:

it( ‘returns the pages.js file using render module’ ){

var result\_url = yield.self.url;

var result\_fileName = yield.self.path;

expect(result\_url).to.eql(‘/’);

expect(result\_url).to.eql(‘/newsfeed);

expect(result\_url).to.eql(‘/contact’);

expect(result\_fileName).to.eql(‘/models/pages.js);

};

1. Function: file\_read

Test Code:

it( ‘reads a specific file from database’ ){

var req\_url = yield.req.url;

var id = yield.id;

var req\_ext = yield.req.extension;

expect(id).to.beDigit();

expect(req\_url).to.eql(‘/download/\*.\*’);

expect(req\_ext).to.noteql(‘jpg’);

expect(req\_ext).to.noteql(‘gif);

expect(req\_ext).to.noteql(‘png’);

};