Coachbook

ATDD with Ruby and Watir

Sample Project : https://github.com/nishants/watir-example

Challenges !

- *No **design pattern** for writing acceptance tests*
- *No **standard for organizing test data***
- *No **build plans***
- *Lack of **architectural standards***
- *MobileWeb-BigWeb **code reuse***
- *Page objects lack **abstraction**, UI dependency is scattered across the codebase.*
- *Page read data from files (**single responsibility ??**)*
- *Missing **archetypes***

Contents

A. *Project Structure and Code Organization*

B. *Defining Responsibilities (for feature/step/page/data)*

C. *Functional Vs Integration Tests (code reusability)*

D. *Creating builds (sample rake commands)*

E. *@tags*

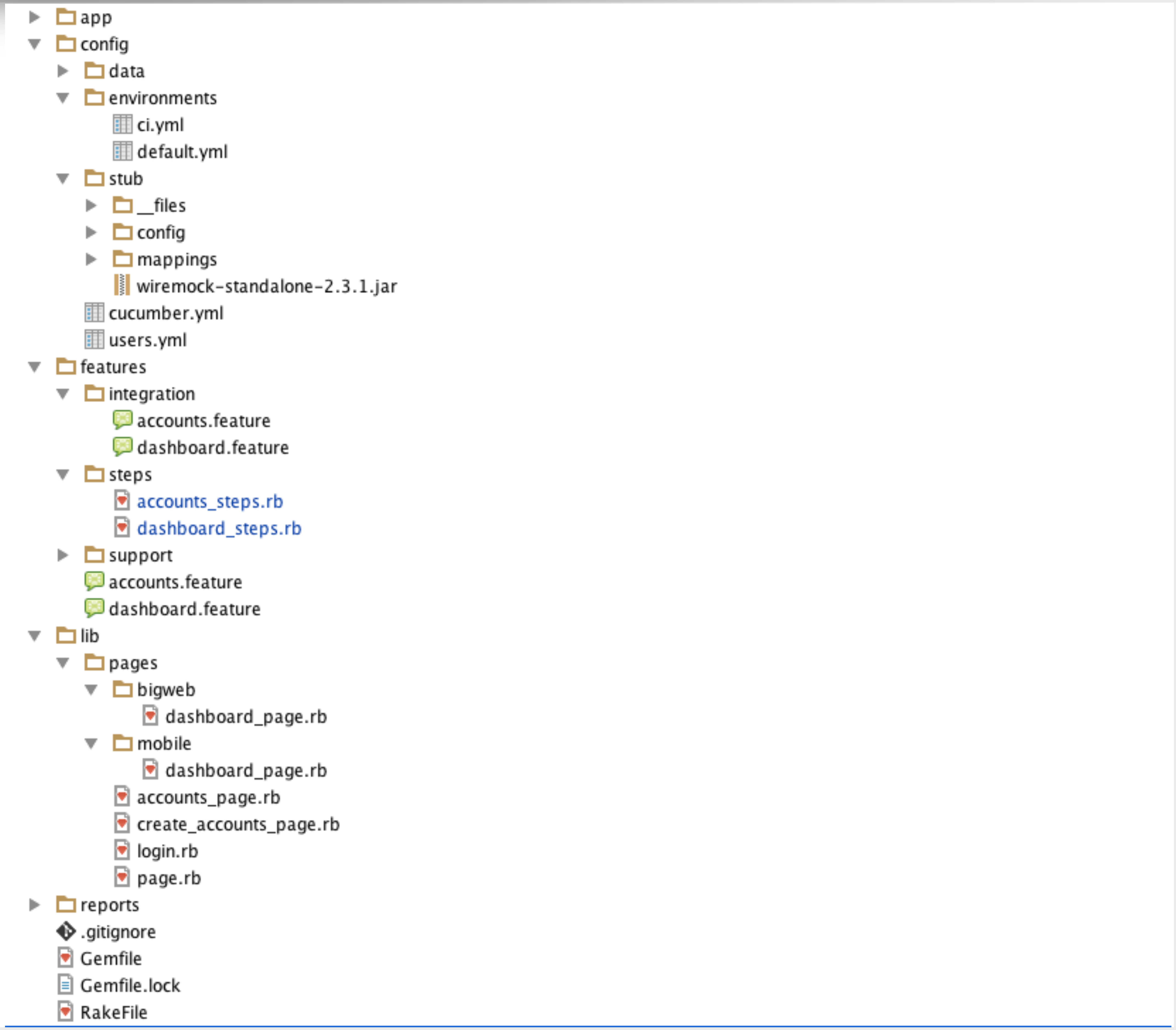
F. *@data*

G. *@users*

H. *A Sample Test (for a user story on wizard and form validation, having separate UI for mobile and desktop)*

Project Structure & Code Organization

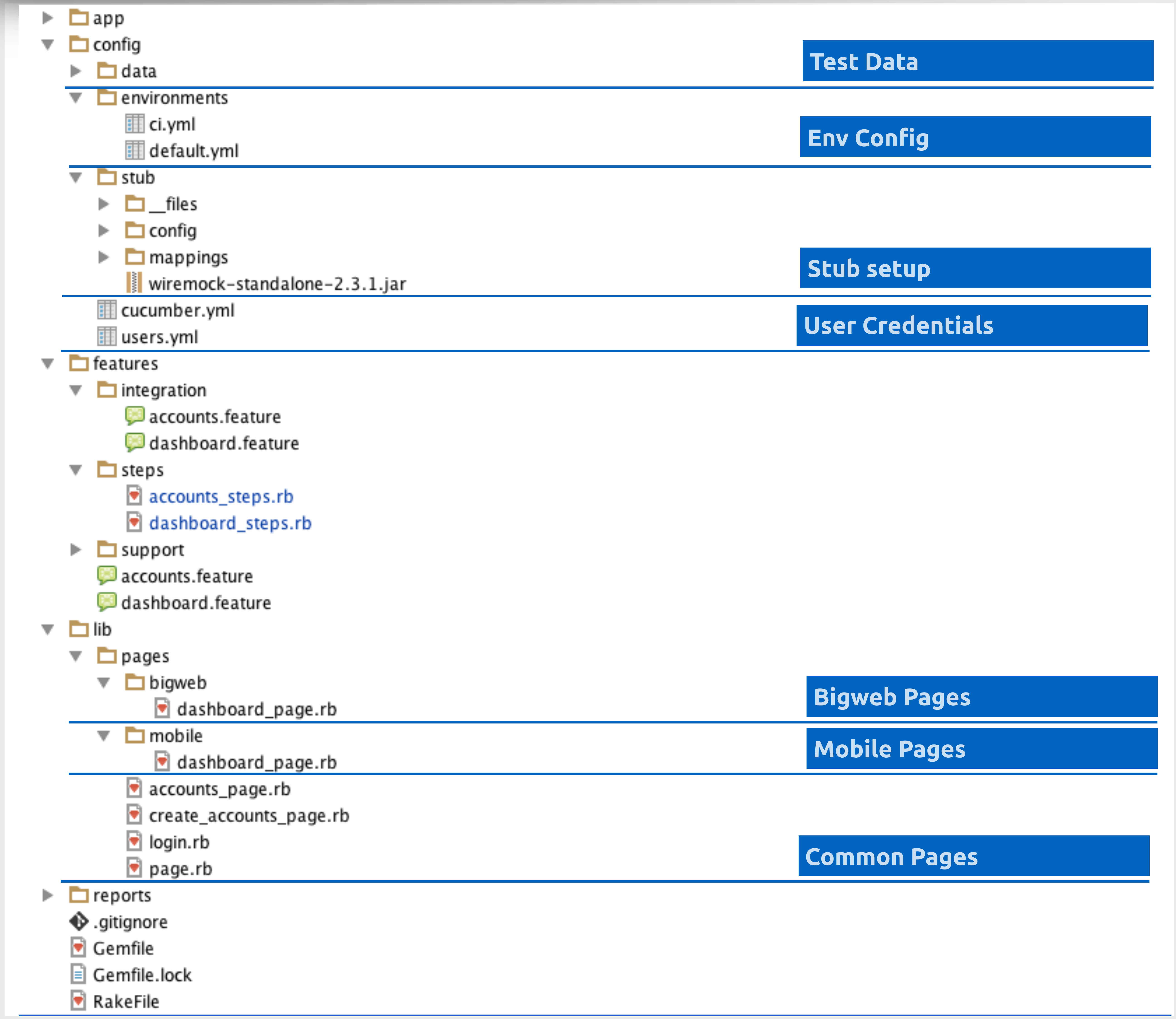
Project Structure & Code Organization



```
▶ app
▼ config
  ▶ data
  ▼ environments
      ci.yml
      default.yml
  ▼ stub
    ▶ __files
    ▶ config
    ▶ mappings
      wiremock-standalone-2.3.1.jar
    cucumber.yml
    users.yml
▼ features
  ▼ integration
      accounts.feature
      dashboard.feature
  ▼ steps
      accounts_steps.rb
      dashboard_steps.rb
  ▶ support
    accounts.feature
    dashboard.feature
▼ lib
  ▼ pages
    ▼ bigweb
        dashboard_page.rb
    ▼ mobile
        dashboard_page.rb
      accounts_page.rb
      create_accounts_page.rb
      login.rb
      page.rb
▶ reports
  .gitignore
  Gemfile
  Gemfile.lock
  RakeFile
```

```
urls :
  default : http://localhost:4567/
  stubbed : http://localhost:4567/

data :
  default : config/data/bigweb
  mobile  : config/data/mobile

page_dir:
  default : ./lib/pages/bigweb
  mobile  : ./lib/pages/mobile

users :
  default : config/users.yml
  stubbed : config/users.yml

stub :
  wiremock :
    jar       : standalone-2.3.1
    path      : config/stub
    port      : 8081
    root_dir  : config/stub
    verbose   : false

clients :
  default:
      browser: chrome
  firefox:
      browser: firefox
  ie:
      browser: ie
  iphone5:
    browser: chrome
    device : Apple iPhone 5
```

Project Structure & Code Organization



```
▶ app
▼ config
    ▶ data
    ▼ environments
          ci.yml
          default.yml
    ▼ stub
        ▶ __files
        ▶ config
        ▶ mappings
          wiremock-standalone-2.3.1.jar
      cucumber.yml
      users.yml
▼ features
    ▼ integration
          accounts.feature
          dashboard.feature
    ▼ steps
          accounts_steps.rb
          dashboard_steps.rb
    ▶ support
      accounts.feature
      dashboard.feature
▼ lib
    ▼ pages
        ▼ bigweb
              dashboard_page.rb
        ▼ mobile
              dashboard_page.rb
          accounts_page.rb
          create_accounts_page.rb
          login.rb
          page.rb
▶ reports
  .gitignore
  Gemfile
  Gemfile.lock
  RakeFile
```

Test Data

Env Config

Stub setup

User Credentials

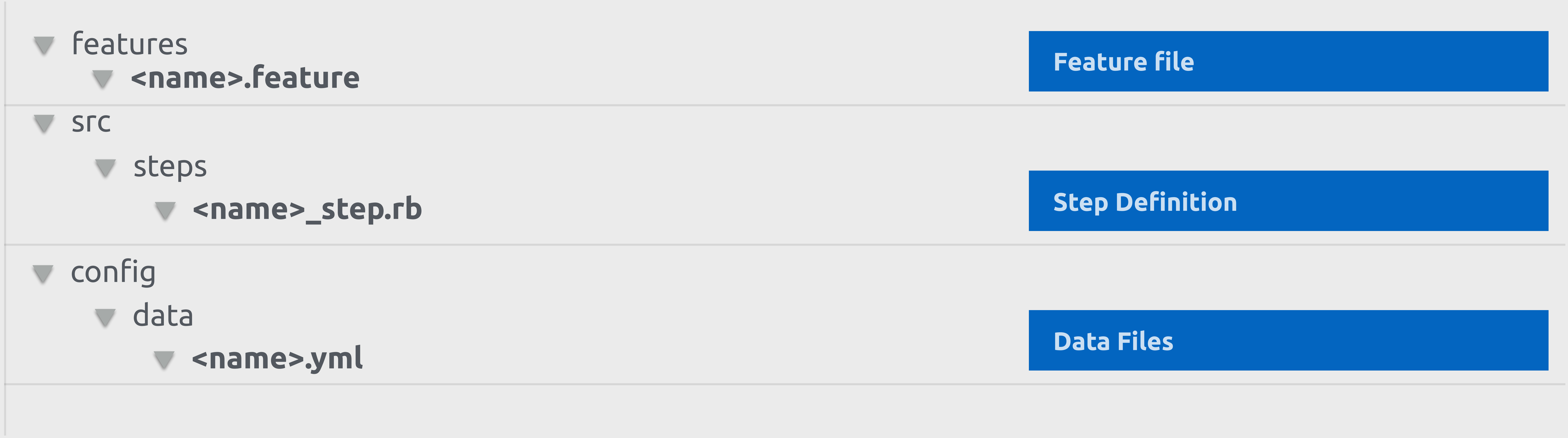
Bigweb Pages

Mobile Pages

Common Pages

```
urls :
  default : http://localhost:4567/
  stubbed : http://localhost:4567/

data :
  default : config/data/bigweb
  mobile  : config/data/mobile

page_dir:
  default : ./lib/pages/bigweb
  mobile  : ./lib/pages/mobile

users :
  default : config/users.yml
  stubbed : config/users.yml

stub :
  wiremock :
    jar       : standalone-2.3.1
    path      : config/stub
    port      : 8081
    root_dir  : config/stub
    verbose   : false

clients :
  default:
      browser: chrome
  firefox:
      browser: firefox
  ie:
      browser: ie
  iphone5:
    browser: chrome
    device : Apple iPhone 5
```

Feature centered **Code** and **Data** Organization



- features
  - **<name>.feature**

Feature file

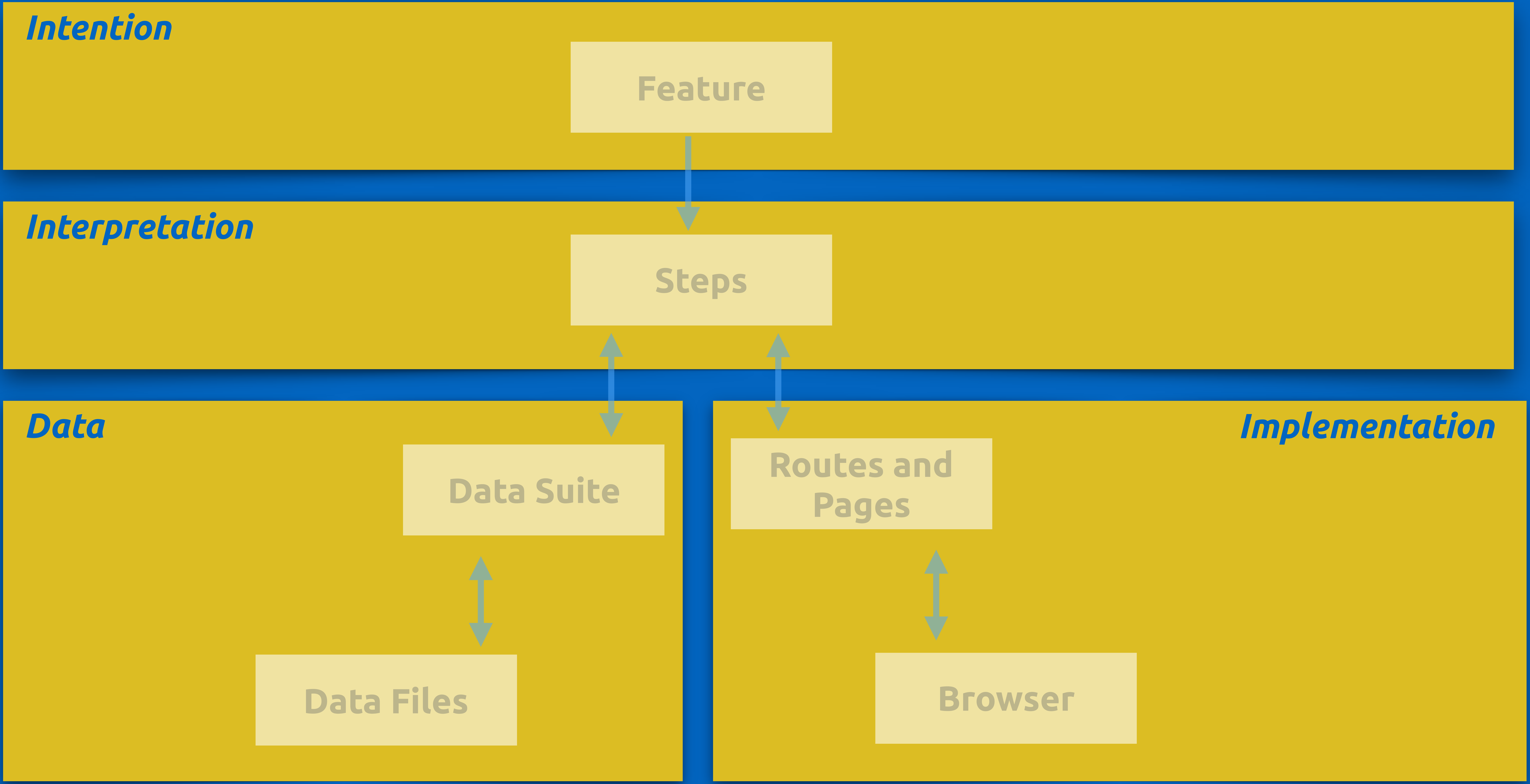
- src
  - steps
    - **<name>_step.rb**

Step Definition

- config
  - data
    - **<name>.yml**

Data Files

Defining Responsibilities


Intention
Feature
Interpretation
Steps
Data
Data Suite
Data Files
Implementation
Routes and Pages
Browser

Feature File **(intent)**	**Step File** **(interpretation)**	**Page and Route** **(implementation)**	**Data** **(setup and expected test data)**
Business Intention	**Interpreting business intention**	**Navigation and View Elements**	**Expectations and Input Data**
Should change if business intent changes	Uses Page to interact with app and Data layer for input and expected values	Should be Maintained against UI changes in application	Provides exact data for functional tests and visual patterns for integration tests
Should be readable and meaningful to business users	Should not need to change for minor ui changes in application	Can have different set of implementations for desktop/ mobile-web/mobile-native.	Can have different data, depending on environment.
Should not provide data for tests, and should be independent of navigation and view elements	Should not talk to browser elements directly	Should not have assertions, Should not talk to data layer, Should have logically meaningful public methods	Organized by features and scenarios

Functional Vs Integration Tests

Functional Vs Integration Tests

features/accounts.feature

```
@accounts
Feature: Accounts

  @admin @create @functional @smoke
  Scenario: Admin creates a guest user
    Given I am "admin"
    Then  I am allowed to create a new account
    And   I am not allowed to create account without username
    And   I am not allowed to create account without role
    And   I am not allowed to create account without email
    Then  I enter all fields and save account
    When  I review user details and confirm
    Then  I can see account listed on accounts page
```

features/integration/accounts.feature

```
@accounts
Feature: Accounts

  @admin @create @integration @smoke
  Scenario: Admin creates a guest user
    Given I am "admin"
    Then  I am allowed to create a new account
    And   I am not allowed to create account without username
    And   I am not allowed to create account without role
    And   I am not allowed to create account without email
    Then  I enter all fields and save account
    When  I review user details and confirm
    And   I can see all users table
```

Functional Vs Integration Tests

features/steps/accounts_steps.rb

```ruby
Then(/^I can see account listed on accounts page$/) do
  expected  = @data.expected_users
  actual    = on(AccountsPage).get_users
  expect(actual).to eq(expected)
end
```

features/steps/accounts_steps.rb

```ruby
And(/^I can see all users table$/) do
  expected  = @data.expected_users
  actual    = on(AccountsPage).get_users
  expect(actual).to look_like(expected)
end
```

Functional Vs Integration Tests

config/data/accounts.yml

```yaml
Admin creates a guest user:
  name_error      : "Can't proceed without username"
  role_error      : "Can't proceed without assigning role"
  email_error     : "Can't proceed without email"
  expected_users  : [
    ["User One", "user-one@mails.com", "10 Minutes"],
    ["User Two", "user-two@mails.com", "9 Hours, 10 Minutes"],
    ["Guest Singh", "guest@one.com", ""]
  ]

  form_data:
    email     : "guest@one.com"
    full_name : "Guest Singh"
    role      : "guest"
    username  : "guest"
```

config/data/integration/accounts.yml

```yaml
Admin creates a guest user:
  name_error      : "Can't proceed without username"
  role_error      : "Can't proceed without assigning role"
  email_error     : "Can't proceed without email"
  expected_users  : [
    ["name", "email", "/Minute|Hour|Day/*"],
  ]

  form_data:
    email     : "guest@one.com"
    full_name : "Guest Singh"
    role      : "guest"
    username  : "guest"
```

Creating Builds

Sample Rake Commands

- ***Functional*** tests, ***Firefox, Dev*** **machine**, for specific feature

```
rake test:bigweb:functional client=firefox tags=@accounts
```

- ***Readonly Integration tests, staging env***

```
rake test:mobile:integration env=staging readonly=true
```

- ***Functional Smoke, for mobile web , on CI machine,***

```
rake test::mobile:functional env=ci tags=@smoke
```

- ***Regression Integration, for big-web , on CI machine,***

```
rake test:bigweb:integration env=ci tags=@regression
```

- ***Integration tests, chrome iphone-5 emulator , CI machine***

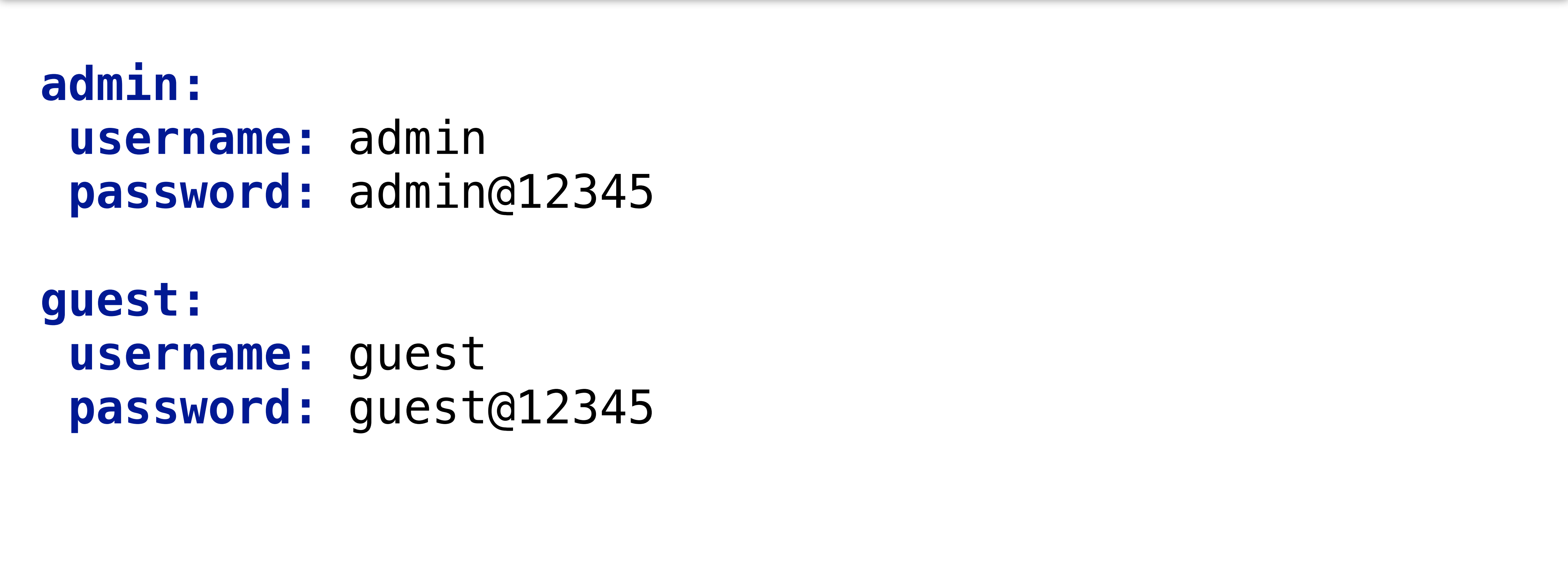
```
rake test:mobile:integration  env=ci  client=iphone5
```

@users

@users

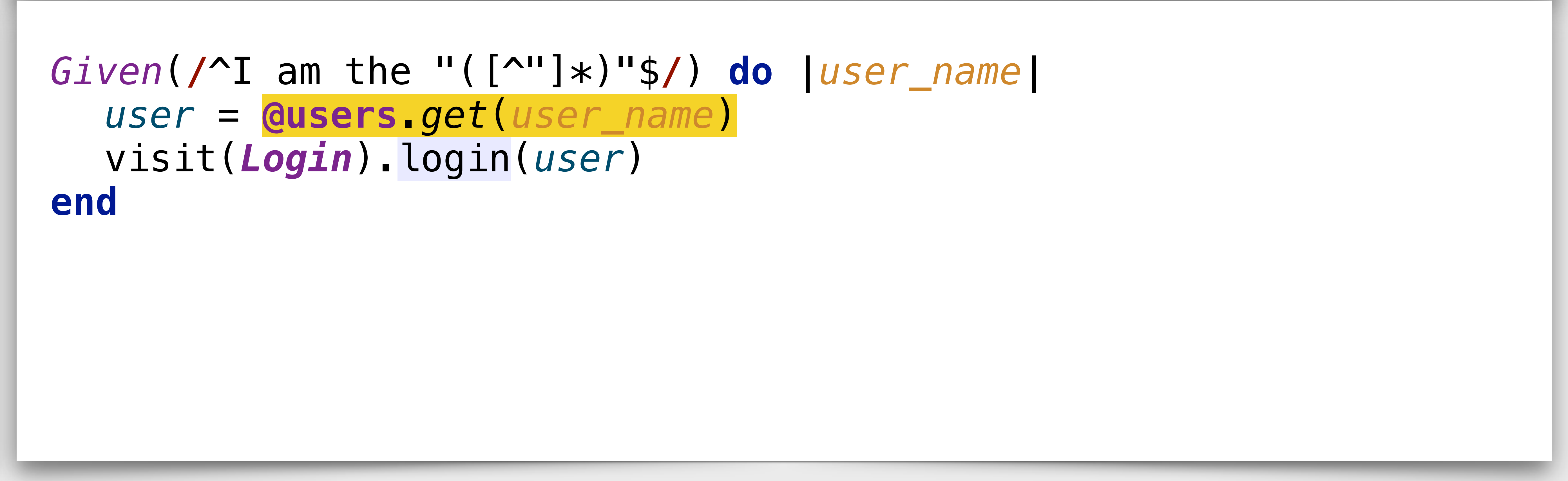
- Add user to config/users.yml
- Use @user.get("user name") to read user in step files.

config/users.yml



```yaml
admin:
 username: admin
 password: admin@12345

guest:
 username: guest
 password: guest@12345
```

step file



```ruby
Given(/^I am the "([^"]*)"$/) do |user_name|
  user = @users.get(user_name)
  visit(Login).login(user)
end
```

@data

@data

To add data for a scenario,

- Go to **config/data/<feature>.yml**

- Create an entry for scenario name,

- Add values to scenario

- Use **@data.param_name to access data in steps**

step file

```
Then(/^I can see my name on dashboard as an admin$/) do
  expected   = @data.welcome_message
  actual     = on(Dashboard).get_user_message
  expect(actual).to eq(expected)
end
```

config/data/<name>.yml

```
Admin creates a guest user:
  welcome_message : "Can't proceed without
```

features/<name.feature>

```
@accounts
Feature: Accounts

  @integration @smoke @admin @create
  Scenario: Admin creates a guest user
    Given I am "admin"
    Then  I am allowed to create a new account
    And   I am not allowed to create account without username
    Then  I enter all fields and save account
    When  I review user details and confirm
    And I can see all users table
```

@tagging

Refer : https://github.com/nishants/watir-example

@tagging

Type (mutually exclusive)	**@funtional,** **@integration,** **@device**	Define types of tests, for using correct runtime config.	
Priority (mutually exclusive)	**@smoke,** **@core,** **@regression**	**smoke > core > regression** **Group tests by priority,** Highest priority ones as **@smoke,** Lower priority, larger in number as **@core**, Lowest priority, largest in number as **@regression**	Tag all tests to **@smoke,** When smoke build takes longer than **10 minutes** to run, Tag less important as **@core,** When **core** build starts taking over **30 minutes,** Tag less important ones as @**regression**
Features	**@admin,** **@mutual_funds,** **@sell_equity**		
Team/Project Specific	**@mobile_only** **@no_mobile** **@readonly,** **@reset_db,** **@quote_service**		

A Sample Test

Refer : https://github.com/nishants/watir-example

User Story : Admin Creates a new User

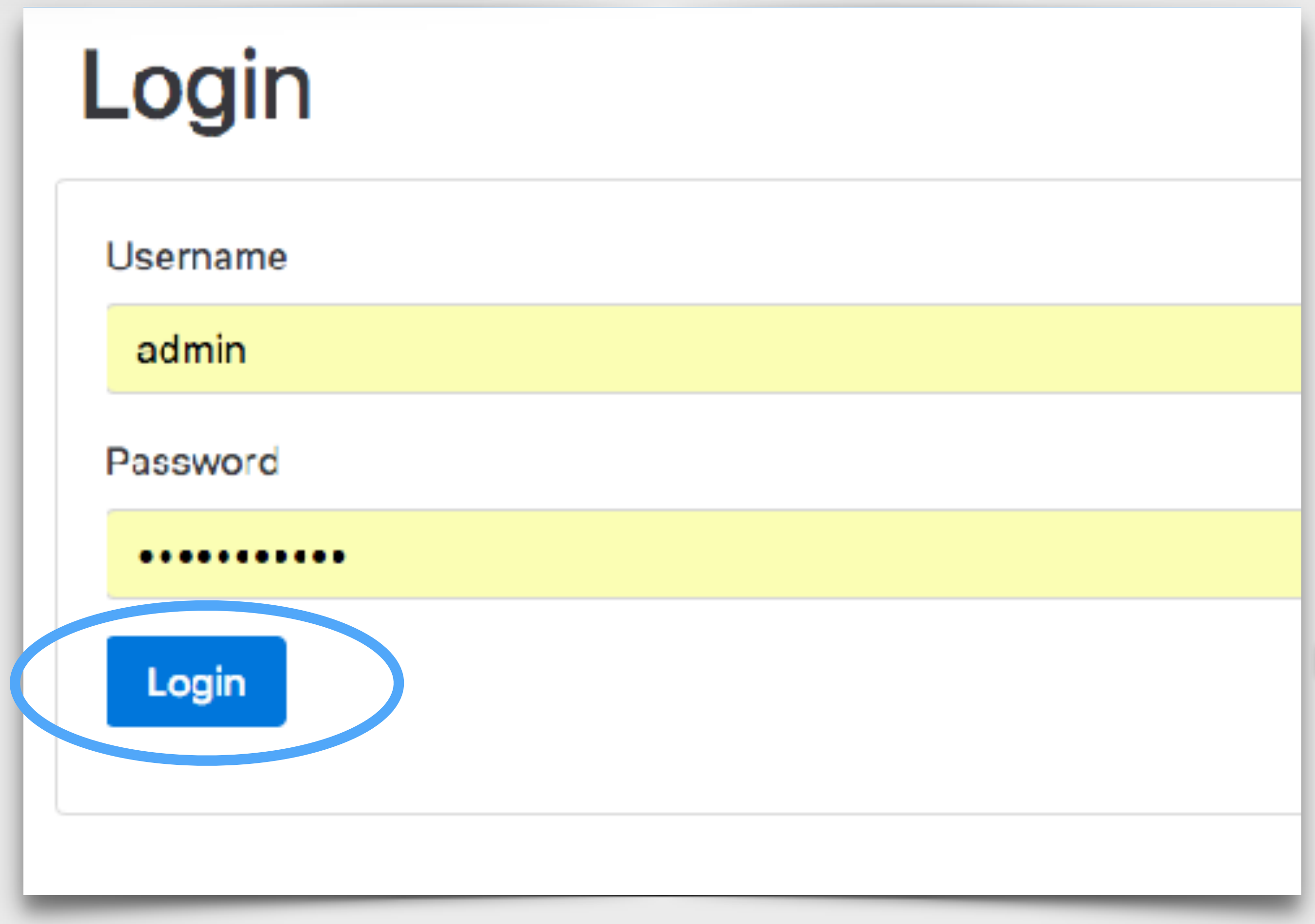
- As an admin I must be able to create a new user
- Admin can select role and username for new user
- Admin can set name for new user
- Admin should be able to review the user details before confirming create account.
- Form must have following validations

  -> missing username : Can't proceed without username

  -> missing email : Can't proceed without email

  -> Missing role : cant proceed without role

User Story : Mockups



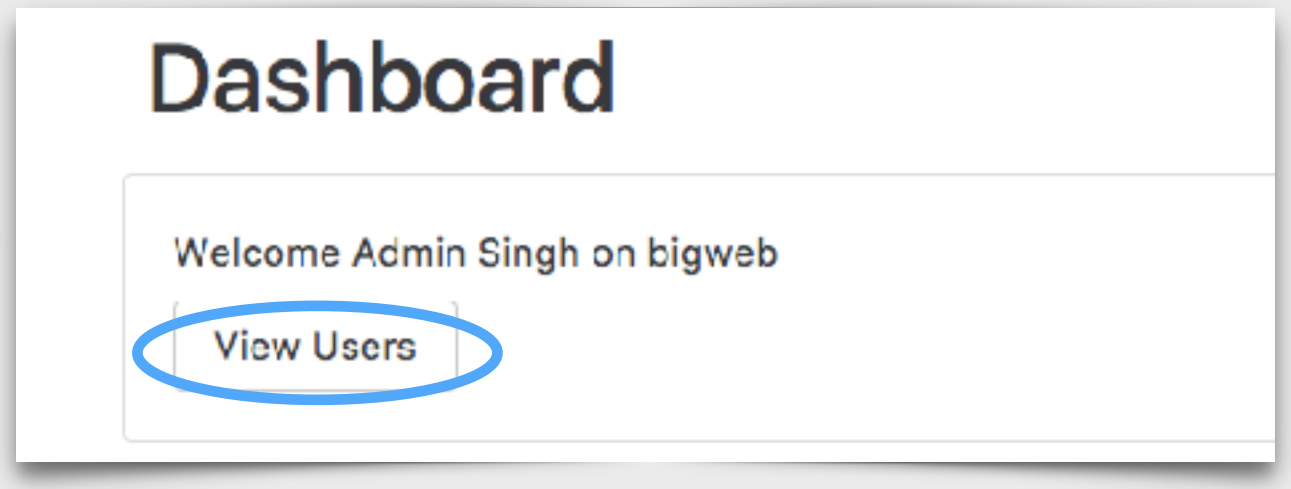
Login

Username

admin

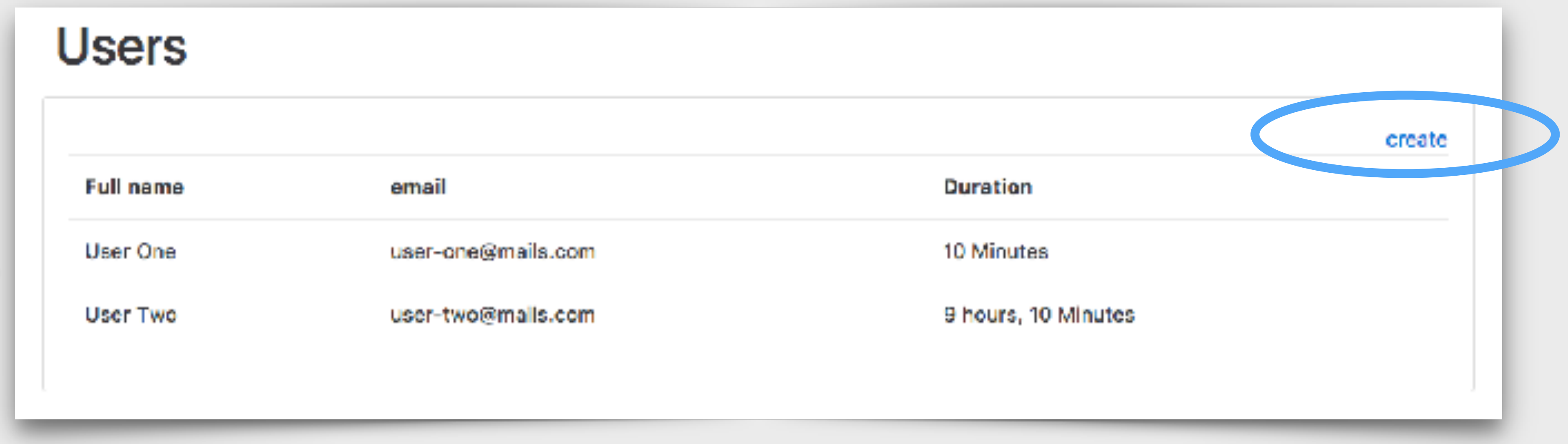
Password

•••••••••••



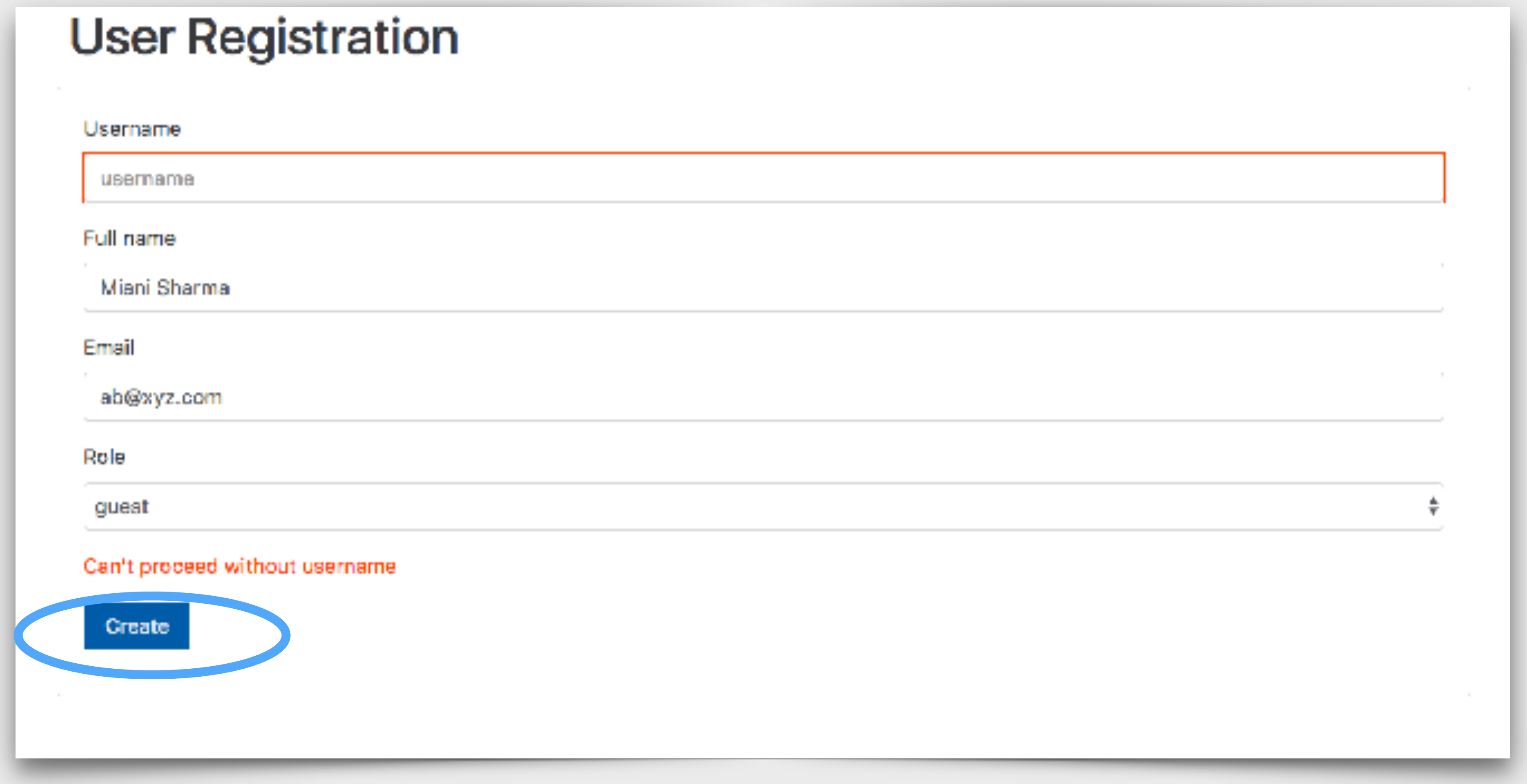
Dashboard

Welcome Admin Singh on bigweb



Users

Full name	email	Duration
User One	user-one@mails.com	10 Minutes
User Two	user-two@mails.com	9 hours, 10 Minutes



User Registration

Username

username

Full name

Miani Sharma

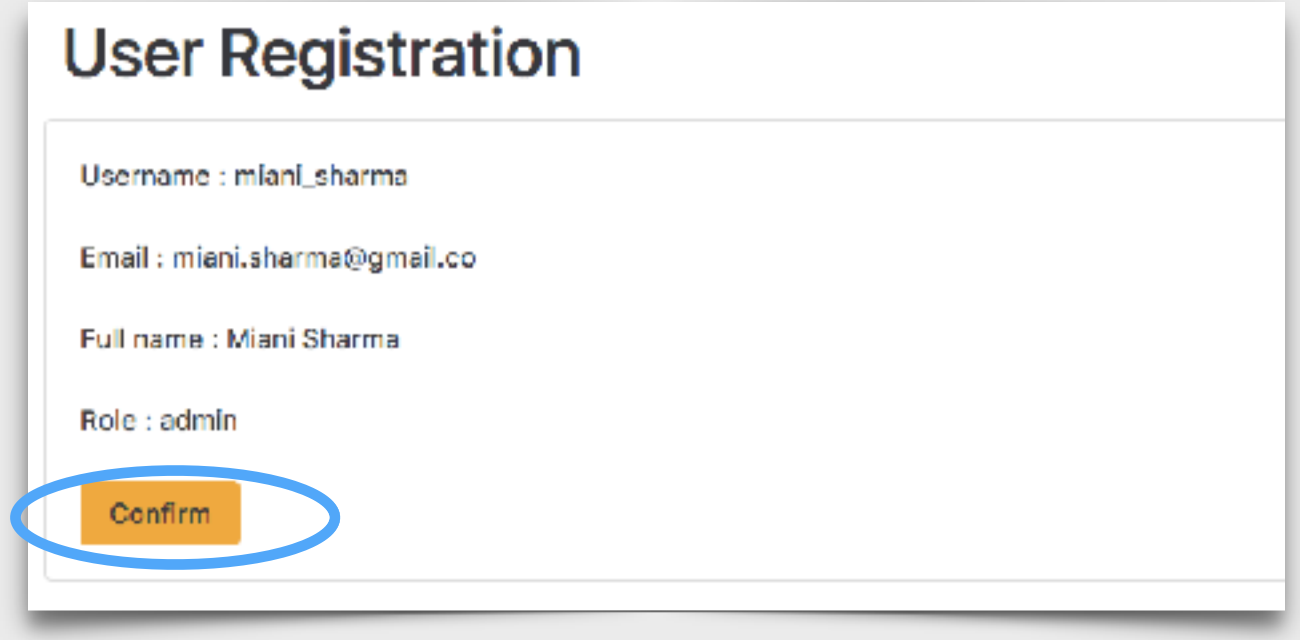
Email

ab@xyz.com

Role

guest

Can't proceed without username



User Registration

Username : miani_sharma

Email : miani.sharma@gmail.co

Full name : Miani Sharma

Role : admin

Flow Analysis : Admin Creates a new User

- **I am allowed to create a new account**
  - - I click the **create** button to view form

- **I am not allowed to create account without username**
  - If I fill all fields and not name
  - and I click create button
  - then I should see error “cant proceed without username” ”

- **I am not allowed to create account without role**
  - If I fill all fields and not name
  - and I click create button
  - then I should see error “cant proceed without role” ”

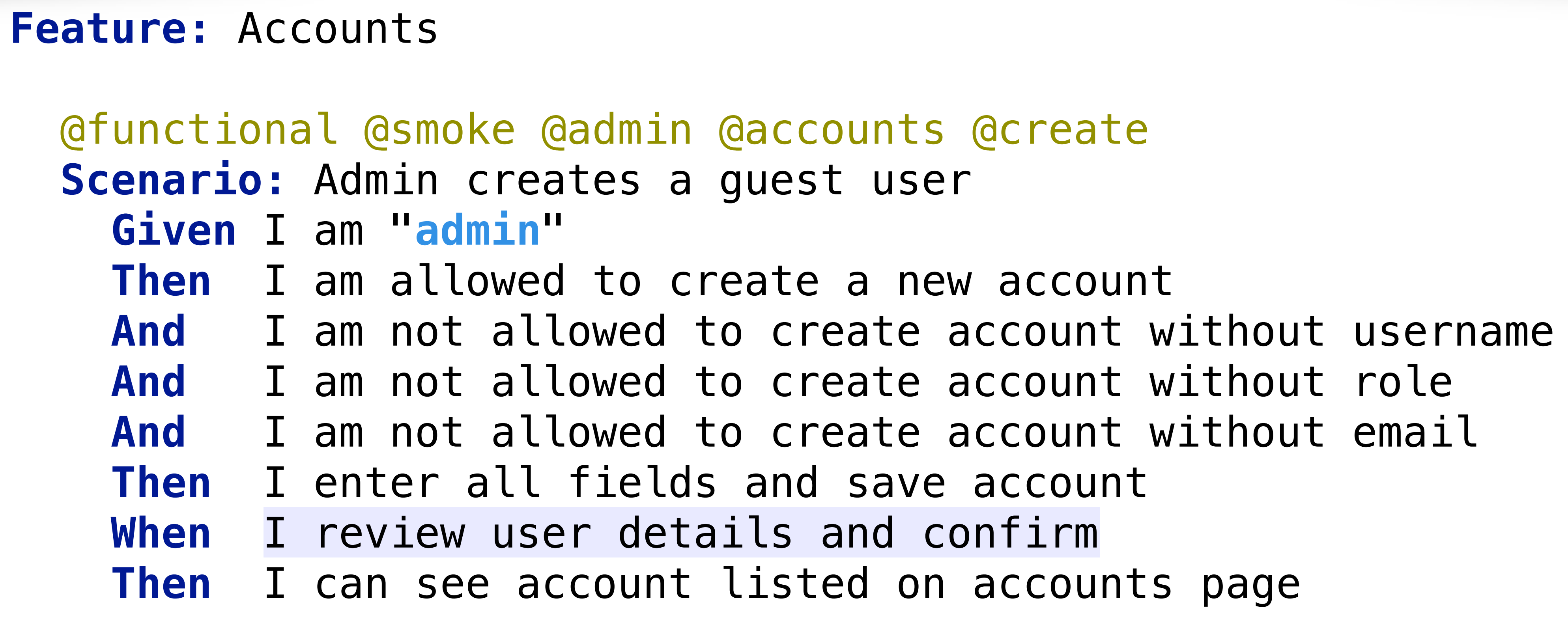
- **I am not allowed to create account without email**
  - If I fill all fields and not name
  - and I click create button
  - then I should see error “cant proceed without email” ”

- **I enter all fields and save account**
    - - fill all fields and click create

- **I review user details and confirm**
- - verify if user details are same as what was entered on the form

- **`I can see account listed on accounts page`**
- - Go to users page, and check if newly created user is present in the table.

features/accounts.rb



```
Feature: Accounts

  @functional @smoke @admin @accounts @create
  Scenario: Admin creates a guest user
    Given I am "admin"
    Then  I am allowed to create a new account
    And   I am not allowed to create account without username
    And   I am not allowed to create account without role
    And   I am not allowed to create account without email
    Then  I enter all fields and save account
    When  I review user details and confirm
    Then  I can see account listed on accounts page
```

Interpretation

I am allowed to create a new account

- I click the **create** button to view form

I am not allowed to create account without username

- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without username”″

I am not allowed to create account without role

- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without role”″

I am not allowed to create account without email

- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without email”″

I enter all fields and save account

- fill all fields and click create

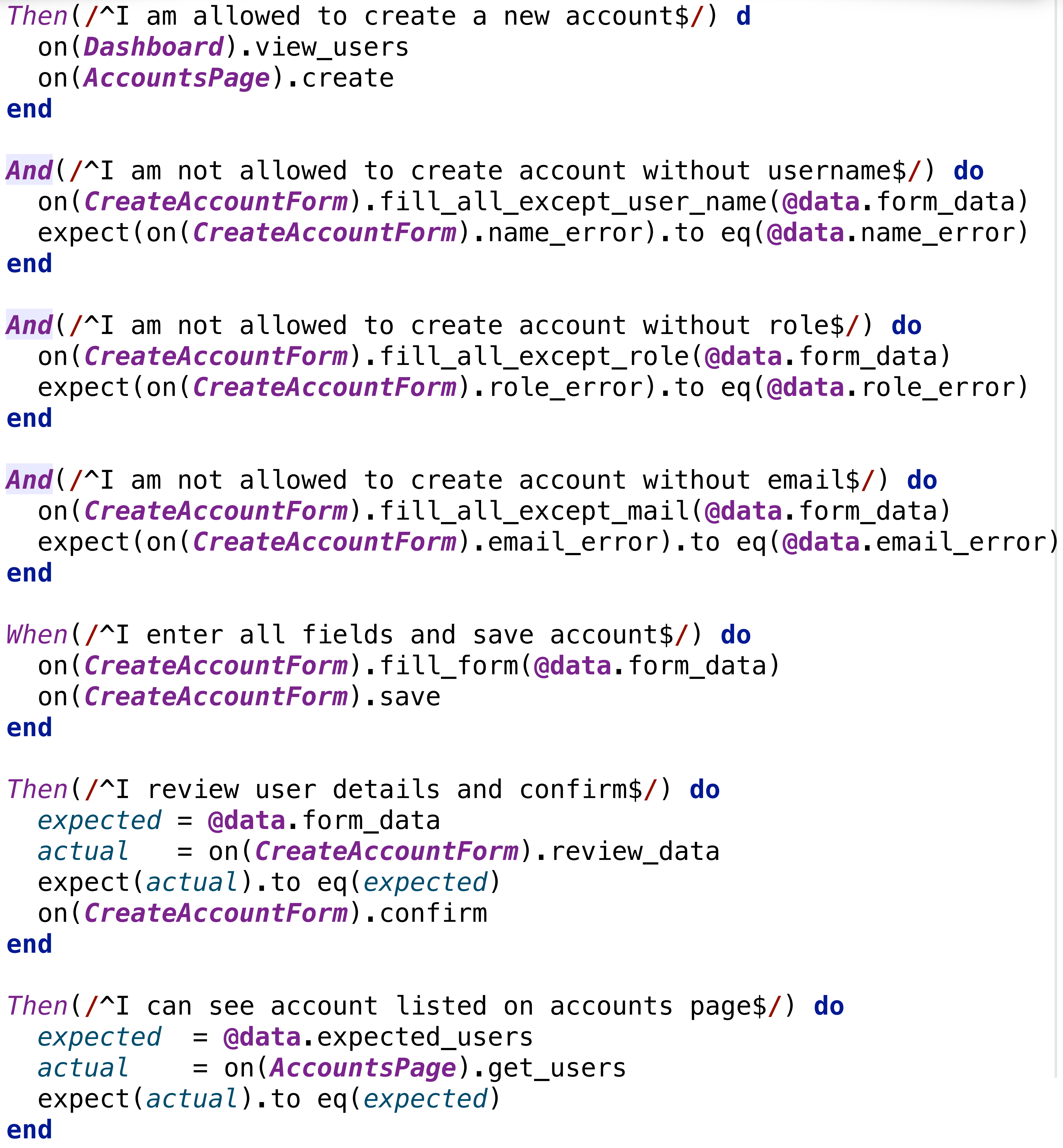
I review user details and confirm

- verify if user details are same as what was entered on the form

I can see account listed on accounts page

- Go to users page, and check if newly created user is present in the table.

feature/steps/accounts_steps.rb



```
Then(/^I am allowed to create a new account$/) d
  on(Dashboard).view_users
  on(AccountsPage).create
end

And(/^I am not allowed to create account without username$/) do
  on(CreateAccountForm).fill_all_except_user_name(@data.form_data)
  expect(on(CreateAccountForm).name_error).to eq(@data.name_error)
end

And(/^I am not allowed to create account without role$/) do
  on(CreateAccountForm).fill_all_except_role(@data.form_data)
  expect(on(CreateAccountForm).role_error).to eq(@data.role_error)
end

And(/^I am not allowed to create account without email$/) do
  on(CreateAccountForm).fill_all_except_mail(@data.form_data)
  expect(on(CreateAccountForm).email_error).to eq(@data.email_error)
end

When(/^I enter all fields and save account$/) do
  on(CreateAccountForm).fill_form(@data.form_data)
  on(CreateAccountForm).save
end

Then(/^I review user details and confirm$/) do
  expected = @data.form_data
  actual   = on(CreateAccountForm).review_data
  expect(actual).to eq(expected)
  on(CreateAccountForm).confirm
end

Then(/^I can see account listed on accounts page$/) do
  expected  = @data.expected_users
  actual    = on(AccountsPage).get_users
  expect(actual).to eq(expected)
end
```

Interpretation

I am allowed to create a new account
- I click the **create** button to view form

I am not allowed to create account without username
- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without username””

I am not allowed to create account without role
- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without role””

I am not allowed to create account without email
- If I fill all fields and not name
- and I click create button
- then I should see error “cant proceed without email””

I enter all fields and save account
- fill all fields and click create

I review user details and confirm
- verify if user details are same as what was entered on the form

I can see account listed on accounts page
- Go to users page, and check if newly created user is present in the table.

features/accounts.rb

```
Feature: Accounts

  @functional @smoke @admin @accounts @create
  Scenario: Admin creates a guest user
    Given I am "admin"
    Then  I am allowed to create a new account
    And   I am not allowed to create account without username
    And   I am not allowed to create account without role
    And   I am not allowed to create account without email
    Then  I enter all fields and save account
    When  I review user details and confirm
    Then  I can see account listed on accounts page
```

config/data/accounts.yaml

```
Admin creates a guest user:
  name_error      : "Can't proceed without username"
  role_error      : "Can't proceed without assigning role"
  email_error     : "Can't proceed without email"
  expected_users  : [
    ["User One", "user-one@mails.com", "10 Minutes"],
    ["User Two", "user-two@mails.com", "9 hours, 10 Minutes"],
    ["Guest Singh", "guest@one.com", ""]
  ]

  form_data:
    email     : "guest@one.com"
    full_name : "Guest Singh"
    role      : "guest"
    username  : "guest"
```

src/steps/accounts_steps.rb

```
Then(/^I am allowed to create a new account$/) d
  on(Dashboard).view_users
  on(AccountsPage).create
end

And(/^I am not allowed to create account without username$/) do
  on(CreateAccountForm).fill_all_except_user_name(@data.form_data)
  expect(on(CreateAccountForm).name_error).to eq(@data.name_error)
end

And(/^I am not allowed to create account without role$/) do
  on(CreateAccountForm).fill_all_except_role(@data.form_data)
  expect(on(CreateAccountForm).role_error).to eq(@data.role_error)
end

And(/^I am not allowed to create account without email$/) do
  on(CreateAccountForm).fill_all_except_mail(@data.form_data)
  expect(on(CreateAccountForm).email_error).to eq(@data.email_error)
end

When(/^I enter all fields and save account$/) do
  on(CreateAccountForm).fill_form(@data.form_data)
  on(CreateAccountForm).save
end

Then(/^I review user details and confirm$/) do
  expected = @data.form_data
  actual   = on(CreateAccountForm).review_data
  expect(actual).to eq(expected)
  on(CreateAccountForm).confirm
end

Then(/^I can see account listed on accounts page$/) do
  expected  = @data.expected_users
  actual    = on(AccountsPage).get_users
  expect(actual).to eq(expected)
end
```

feature/steps/accounts_steps.rb

```ruby
Then(/^I am allowed to create a new account$/) d
  on(Dashboard).view_users
  on(AccountsPage).create
end

And(/^I am not allowed to create account without username$/) do
  on(CreateAccountForm).fill_all_except_user_name(@data.form_data)
  expect(on(CreateAccountForm).name_error).to eq(@data.name_error)
end

And(/^I am not allowed to create account without role$/) do
  on(CreateAccountForm).fill_all_except_role(@data.form_data)
  expect(on(CreateAccountForm).role_error).to eq(@data.role_error)
end

And(/^I am not allowed to create account without email$/) do
  on(CreateAccountForm).fill_all_except_mail(@data.form_data)
  expect(on(CreateAccountForm).email_error).to eq(@data.email_error)
end

When(/^I enter all fields and save account$/) do
  on(CreateAccountForm).fill_form(@data.form_data)
  on(CreateAccountForm).save
end

Then(/^I review user details and confirm$/) do
  expected = @data.form_data
  actual   = on(CreateAccountForm).review_data
  expect(actual).to eq(expected)
  on(CreateAccountForm).confirm
end

Then(/^I can see account listed on accounts page$/) do
  expected  = @data.expected_users
  actual    = on(AccountsPage).get_users
  expect(actual).to eq(expected)
end
```

lib/pages/create_account_form.rb

```ruby
class CreateAccountForm < Page
  def fill_form(form)
    @browser.text_field(:name =>  "email").when_present.set(form["email"])
    @browser.text_field(:name =>  "username").when_present.set(form["username"])
    @browser.text_field(:name =>  "full-name").when_present.set(form["full_name"])
    @browser.select_list(:name =>  "select-role").when_present.select(form["role"])
  end

  def fill_all_except_mail(data)
    fill_form(data)
    @browser.text_field(:name =>  "email").when_present.set("")
    save
  end

  def fill_all_except_role(data)
    fill_form(data)
    @browser.select_list(:name => “select-role").option(:index => 0).select
    save
  end

  def fill_all_except_user_name(data)
    fill_form(data)
    @browser.text_field(:name =>  "username").when_present.set("")
    save
  end

  def name_error
    @browser.element(:class => "username-error").when_present.text
  end

  def role_error
    @browser.element(:class => "role-error").when_present.text
  end

  def email_error
    @browser.element(:class => "email-error").when_present.text
  end

  def confirm
    @browser.link(:name => "confirm-new-account").when_present.click
  end
```

Work In Progress...