

# NXTFolio – Iteration 1

## Team Members

- Balmaseda del Campo, Vicente – Scrum Master
- Fletcher, Lance
- Jain, Ayushri – Product Owner
- Li, Baichuan
- Li, Wenyan
- Price, Niah

## Links

**GitHub:** <https://github.com/vibalcam/match-my-fashion-public-CodeCreators>

**Pivotal Tracker:** <https://www.pivotaltracker.com/n/projects/2630237>

**Heroku Deployment:** <http://nxt-folio.herokuapp.com/>

## Client Meeting

**Customer Meeting:** Zoom, Friday, 6pm

## Completed User Stories & Tasks

### **Project Ruby & Dependency Update**

This legacy project was using Ruby 2.4 and old versions of Rails and other gems. This resulted in problems setting it up for development and not being able to deploy it to Heroku. Moreover, some dependencies were no longer maintained, such as “therubyracer”, which made it difficult to find a fix to the problems that arose. After trying to fix the existing issues, we decided to upgrade the versions. Although this would entail upgrading a major version, and thus breaking some of the functionality which needed to be fixed plus dealing with dependency hell, we think that it was the right choice to create new features and to make deployment easier for future groups. Moreover, not doing so would result in the problems becoming larger over time. Instead, doing so makes it easier to gradually upgrade the dependencies as new versions become available. Furthermore, many of the app’s features were not

working in the current Ruby 2.4 and the tests were failing. Upgrading and fixing allow us to set up correctly the work environment. We upgraded the Ruby version to 3.2.0 and upgraded the gems being used. We also substituted “therubyracer” for “nodejs”, which has the same functionality to embed javascript using “Execjs” and is well maintained.

### **Migration and Installation Fixes**

We found some issues running the existing migrations and setting up the Postgres database which they were using. The migration issue consisted of duplicate columns being created. We isolated the migration that was creating the duplicated column and, after checking the schema, removed it since those columns already existed. The database issues mainly came from the usage of Postgres and unhelpful installation instructions. To make it easier for future groups to set up all the dependencies, we created a Dockerfile running Ubuntu 18 (same system as Cloud9 to allow using Docker or AWS) and a script that automizes the process of downloading dependencies, RVM, ruby 3.2.0, installing and setting up Postgres, and bundling the app. By using Docker together with a script we hope to allow setting up the app in a few simple steps, thus easing the installation process for future groups so they can get to get the app working faster. We also modified the Readme accordingly.

### **Gallery Fix**

In this iteration, a user should be able to create, delete and add own reviews in the profile page and gallery page, in this way, users can add several images as a gallery to show their project. Also, several users can add ratings, and the average rating will be shown in the user’s profile page. In the future, some features can be improved like enabling users to delete an individual picture. Generally, the functions in the gallery page are working well.

### **Database Seeding:**

To better facilitate development of the application, the database needed to be populated with data to better display the application's features. For example, when searching for users, the database needs to have users in the database to fully see this feature. The seeded data will likely be able to help in acceptance testing. For now, the seed file populates the LoginInfo, GeneralInfo, and Gallery tables. These tables store data which is core to the functionality of the application. In the future, we may look at seeding the database with other data such as ratings, comments, etc. During this process it was discovered that the passwords were being stored in plain text in the database, which is a large security risk. In the future, we plan to address this issue.

### **Heroku Deployment**

An important objective of iteration 1 is to deploy the application. We utilized Heroku for this deployment because of its ability to easily deploy ruby applications. The deployment took longer than was desired, but a delay was caused by the ruby version upgrade and the subsequent library upgrades. When initially deploying to Heroku, it failed due to a missing the storage.yml configuration file. Once this file was added the application was able to be deployed

and was working properly. All the features which have been fixed up to this point have been pushed to deployment and are working as expected.

### **Fix sign up page : #184489925**

After upgrading to Ruby 3.2.0 and Rails 6.1.4.2, the sign up and login page stopped working. Since we were unable to sign up, we could not login either and see other user details. To fix this problem, `general_info_controller.rb` file was updated. To test the changes, existing test cases were used. It was difficult to figure out which test cases should be used since there were multiple test cases which were not working completely, so it took time. Later, test case for `user_story_2.feature` was updated with a minor change and was run via cucumber. Note – this user story was named "Fix test cases" earlier which was only a placeholder. Also, the original pull request was not merged into the master branch but it was merged with another branch for seeding and then that branch was merged with master branch.

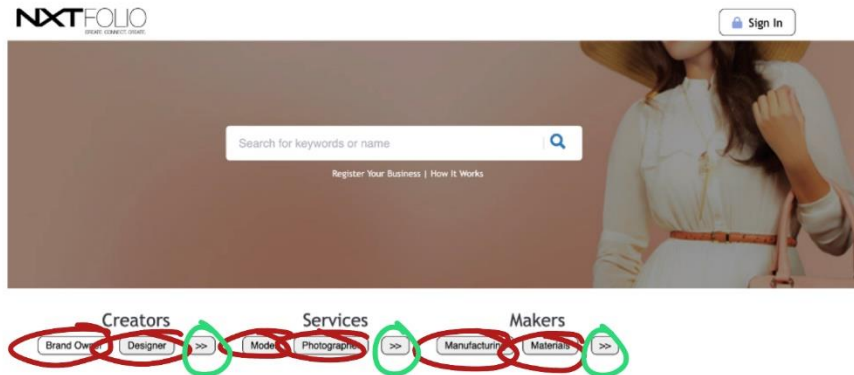
### **Rspec Tests**

All the RSpec test files were initially not runnable. We made the necessary modifications and deletions to make them executable. However, some tests are still failing though runnable, and we need to investigate the root cause of these failures. We can either modify the non-functional tests or modify the codes to add the required functionality in order to make the tests pass.

### **Search Function**

Regarding the search function, I tested it using seeded data and found that fashion professionals can be easily located by searching their first and last names. The filters also work. Furthermore, I added several minitests to the test folder and confirmed that they all passed. To enhance the user experience, I made some changes to the UI, such as modifying the color of the search box and adding instructions to prompt users to search more efficiently. Currently, the search results are not displaying properly, and the information of the professionals only appears when the user hovers the mouse over the relevant item. It is hard to get brief information about professionals in this way. We will modify it soon.

## User Stories Moved to Iteration 2



**Red: Non-functioning**

**Green: Functioning**

The buttons on the home screen are missing critical data from the seeds.rb file, which leaves the buttons without a destination to navigate to. The buttons circled in red as well as the categories that are made visible after pressing the “>>” button are non-functioning. The buttons circled in green are functional and are used to extend the categories listed under each topic. Using the files from the schema.rb file is taking longer than expected but is expected to be resolved by the next iteration.

## Tests

- Existing test case used – user\_story\_2.feature.

```
2023-02-24 17:02:13 WARN Selenium [DEPRECATION] [:capabilities] The :capabilities parameter for Selenium::WebDriver::Chrome::Driver is deprecated. Use :options argument with an instance of Selenium::WebDriver::Chrome::Driver instead.
Given I am a professional and want to create my profile # features/step_definitions/user_story_2_steps.rb:1
When I click on "Register Your Business" in the home page # features/step_definitions/user_story_2_steps.rb:5
Then I should be able to provide username and password 2 times to register # features/step_definitions/user_story_2_steps.rb:9
And I should be redirected to the first profile data entry page # features/step_definitions/user_story_2_steps.rb:15

Scenario: Provide data on profile Data Entry page # features/user_story_2.feature:17
Given I am a professional and want to enter personal data # features/step_definitions/user_story_2_steps
.rb:21
When I submitted all the required personal information # features/step_definitions/user_story_2_steps
.rb:31
sessions current_user_key:a305d96ce92d90f92265
#a305d96ce92d90f92265"
#<GeneralInfo id: 1, userKey: "a305d96ce92d90f92265", first name: "Test", last name: "User", month_ofbirth: nil, day_ofbirth: nil, year_ofb
irth: nil, gender: nil, company: "Test Company", highlights: "highlights", industry: "Creators", country: "United States", state: "Texas",
city: "College Station", compensation: "Salary", facebook_link: "My site address", linkedin_link: "", instagram_link: nil, personalWebsite
link: "", bio: "About me", created_at: "2023-02-24 23:02:33.886650000 +0000", updated_at: "2023-02-24 23:02:35.603629000 +0000", specific_p
rofile_id: nil, profile_picture: nil, cover_picture: nil, gallery_picture: nil, template_id: nil, specific_profile: nil, phone: nil, latit
ude: 30.5955289, longitude: -96.3071042, job_attr: {}, job_name: "Admin", is_admin: true, email: nil, specialization: "my specialization",
profdetails: "ProfessionalDetails", experience: "2", emailaddr: "fashionxt@gmail.com", notification: false, notification_from: []>
Unknown profile type! Profile type given:
nil
{}
Then After save I redirect to professional detail page and provide information and submit # features/step_definitions/user_story_2_steps
.rb:49
And I will be redirected to profile detail page # features/step_definitions/user_story_2_steps
.rb:64

2 scenarios (2 passed)
8 steps (8 passed)
0m43.257s
```

**Design diagram for this iteration**

Most of the work done this Iteration1 has been focused on fixing features from the legacy code that did not work, refactoring, and upgrading Ruby and the gems. Thus, the design diagram for the app has not changed on this Iteration1. To make it easier to access, we are including the design diagram provided by the previous team, which is also the design diagram for the current iteration.