Evaluation of a Haskell Web Framework - Progress Report Computer Science

Junaid Ali Rasheed

April 20, 2018

Contents

0.1. Introduction

0.1.1. What is this project about?

The goal of this project is to compare two different frameworks, one written in Haskell and one written in another, more popular framework in an Object Oriented language.

In both of the chosen frameworks, I will write a simple clone of twitter, a social media website. The two websites will be functionally identical to allow a fair comparison between the chosen frameworks. When evaluating the frameworks, I will compare the reliability, maintainability, speed, functionality, and the efficiency of both websites.

0.2. The Chosen Frameworks

Yesod is a Web Framework written in Haskell. The reason I chose Yesod is because, from my initial impressions, it seemed to be a fully featured and modular web framework. Yesod also claims to use features of the Haskell language to provide a fast, modular, and type safe web framework. One of the goals of the project is to see if Haskell's type safety, referential transparency, and lazy compiling is an advantage for web developers, so Yesod seems like a good choice to get the most out of Haskell's features.

> Yesod attempts to ease the web development process by playing to the strengths of the Haskell programming language. Haskell's strong compile-time guarantees of correctness not only encompass types; referential transparency ensures that we don't have any unintended side effects. Pattern matching on algebraic data types can help guarantee we've accounted for every possible case. By building upon Haskell, entire classes of bugs disappear. (Snoyman 2012, Introduction)

Django is a Python Web Framework. Django, like Yesod, is a "batteries included" web framework, "instead of having to open up the language to insert your own power (batteries), you just have to flick the switch and Django does the rest." (George 2017). I chose Django for a number of reasons

- "Batteries included", like Yesod
- Modular, like Yesod
- Python is now more common than PHP, second only to node.js (George 2017)

The Django and Yesod web frameworks have a similar set of features. I believe that it will be possible to make a functionally identical site in both of these frameworks with a similar amount of effort. This would give us a fair comparison between Django and Yesod, allowing us to come to a conclusion on whether a Haskell web framework may be a good choice for a developer rather than a more tradition web framework.

0.3. Learning The Frameworks

Haskell is very different than any language that I have used before, coming from an Object Oriented Background and being used to languages like Python, Java, and PHP. To ensure that my lack of knowledge of Haskell does not give too much of an advantage to Django, I read the book "Haskell Programming from First Principles" (Allen and Moronuki 2016). I used this book to get a general understanding of Haskell and the principles of Functional Programming. After a few weeks of reading through the book, I started reading "Developing Web Applications with Haskell and Yesod" (Snoyman 2012), a tutorial for the Yesod framework written by the original developer for Yesod, Michael Snoyman. After going through some of the Yesod book, I started to develop my twitter clone in Yesod. Developing the website, referring to the books when I needed to, and getting help on some concepts that I was struggling to learn from my tutor, greatly improved my knowledge of both Haskell and Yesod in a short amount of time.

I was already familiar with the Python language before starting this project, and being familiar with Python is one of the reasons that I chose the Django framework. To learn the basics of Django, I went through the tutorials available on the Django website (Django 2018). This introduction guided you through writing a simple Django app, teaching you some of the basic concepts of the Django.

One thing that I noticed straight away is that, because of the high popularity of the Django web framework, if I needed any help, I could just type my query into a search engine, and most likely, someone else has had the same problem which has already been resolved in a forum post. This is in contrast to Yesod, where most of the time, if I encounted any difficulties, I would have to go through the documentation to try and find a solution. The popularity of Django does give it an advantage in this respect, any problem you come across has most likely already been encountered and resolved by another person. However, the documentation of Yesod is excellent in my opinion, and most problems I have come across were resolved rather quickly. I have also recently discovered that Yesod has an IRC channel which I plan to use if I encounter an error that I am having difficulty solving in the future.

0.4. Progress So Far

I have started development on both the Django and Yesod websites. To ease the setup of a development environment, I have created vagrant boxes for both frameworks. Vagrant is a tool that allows you to automatically setup a virtual machine and install the operating system and a set of programs specified by you in a script. The vagrant box I have created automatically installs Ubuntu and the necessary tools required to develop for both Django and Yesod. Both sites are using a PostgreSQL database to ensure that any speed or database integrity comparisons are as fair as possible.

Bootstrap is being used for styling for both websites. Using bootstrap allows me to easily ensure that both websites look attractive and identical from the frontend without much effort on my

part. This allows me to focus on the functionality of the websites.

Login and Signup functionality has been implemented on both websites. Users can visit the website homepage, create an account, and login to an existing account. The Yesod website also has functionality to create messages. These messages are displayed on the profile page of a user. I am currently in the middle of implementing the same functionality in Django.

0.4.1. Tests

The testing syntax for both frameworks is actually pretty similar. I have implemented tests for the login and signup functionality in both of the frameworks. The two code blocks below show a test in both frameworks. Both of these tests are ensuring that the correct error messages are displayed when a user does not provide any input in the login form.

Code 1: "A Django Test"

```
def test_create_account_no_input(self):
  """
  If no data is input, a valid message is displayed and the user is redirected
      to the sign up form
  """
  response = self.client.get(reverse('base:register'), follow=True)
  self.assertEqual(response.status_code, 200)
  self.assertRedirects(response, reverse('base:signup'))
  messages = list(response.context.get('messages'))
  self.assertEqual(len(messages), 3)
  for message in messages:
    self.assertEqual(message.tags, 'danger error')
  self.assertEqual(str(messages[0]), 'Please enter a username')
  self.assertEqual(str(messages[1]), 'Please enter a password')
  self.assertEqual(str(messages[2]), 'Please enter an email address')
```

Code 2: "A Yesod Test"

```
it "redirects to signup page with messages when no input is given" $ do
  get SignupR
  statusIs 200

  request $ do
    setMethod "POST"
    setUrl SignupR
    addToken
    byLabel "Username" ""
    byLabel "Email" ""
    byLabel "Password" ""

  statusIs 303
  _ <- followRedirect
```

```
statusIs 200

htmlAnyContain ".alert-danger > span" "Value is required"
htmlCount ".alert-danger > span" 3
```

0.4.2. Plans for the Immediate Future

The immediate plans for the future is to finish the message functionality in Django. Once this is finished, extensive tests will be written in both Django and Yesod for this functionality. Once I have finished working on these tests. I will start working on using AJAX in both frameworks. For AJAX, the plan is to refactor the post message functionality to use AJAX and to refactor the profile page to automatically update the messages posted by the owner of the profile page.

References

Allen, Christopher and Julie Moronuki (2016). *Haskell Programming from first principles*.

Django (2018). *Getting Started | Django Documentation | Django*. URL: https://docs.djangoproject.com/en/1.11/intro/ (visited on 19/01/2018).

George, Nigel (2017). *Why Django? The Django Book*. URL: https://djangobook.com/tutorials/why-django/ (visited on 19/01/2018).

Snoyman, Michael (2012). *Developing Web Applications with Haskell and Yesod*. O'Reilly Media.

Appendices

A. Project Diary

A.1. Meeting 1 - 3rd October 2017

A.1.1. Meeting Notes:

Books

Real World Haskell
Haskell from first principles (haskellbook.com)
Web application development with Haskell and Yesod (out of date)

Frameworks / Tools

Haskell Servant package
Snap is alternative to Yesod
ghcjs haskell to js
haskell stack tool
hackage is like npm. Stack can use hackage.
Stackage is like stack on top of hackage
Use the latest LTS version of haskell from stackage
Atom could be useful with their plugins, compare with plugins available for code
ghc-mod available for haskell in atom, helpful when developing
ide-haskell, linter
There is a Haskell plugin for intellij which may work. Good because I would be familiar with the IDE.

Comparing the two frameworks

- Maintainability
  - Make a change to both
- Performance
- Scalability - could use tools, hard to do on your own
- People say Haskell is easier to write code with, less time debugging, once learnt

- We could test this. How much the type checking helps. The different tools available
- Can't use line by line debugging

Plan for next meeting

Do as much as possible for now
Come up with rough project definition form
Go through some haskell tutorials, haskellbook.com is recommended

A.2. Meeting 2 - 12th October 2017

A.2.1. Meeting Notes:

Look into getting GHC mod compile on save
Get the project proposal doc ready for next week
Learn Django and get it installed on the laptop
Make a basic page in Django and Haskell

A.3. Meeting 3 - 19th October 2017

A.3.1. Meeting Notes:

Carry on with the Haskell Programming from First principles book
Have some planning for the twitter clone ready

A.4. Meeting 4 - 24th October 2017

A.4.1. Meeting Notes:

Set up a basic homepage in Yesod and Django. Do this over the weekend.
Have a play around with the yesod site that's provided to see what you can focus on.
Carry on with the book
Setup Docker/Vagrant if you have time at the end, for instructions on setting up the repo
Topics important for yesod

- Quasi quotes, provided by yesod
- Yesod Typeclass could be useful to know

A.5. Meeting 5 - 10th November 2017

A.5.1. Meeting Notes:

I've created the homepages in both yesod and django. I've used tests in django to test a basic app not related to the project

Next week, I want to ensure both home pages are the same and to create tests in both frameworks. I want to progress more through the yesod and haskell book. Create User models in both yesod and django and create tests for them.

A.6. Meeting 6 - 16th November 2017

A.6.1. Meeting Notes:

I've created the homepages in yesod and django and ensured that they both have the same content and styling.

For django, I have added the functionality to allow users to create accounts and log in. I have added unit tests for this and they all pass.

For yesod, I have added the latest version of jquery and bootstrap to the project. I have tried to complete the user account functionality but I am blocked. I am trying to import yesod-auth-hashdb but cannot figure out how to do it. There is some documentation showing how to edit the cabal file but this is overwritten during the build, I believe the data comes from package.yml. Editing package.yml causes strange errors when I try to build the project but I don't think I am doing it in the correct manner. Need to figure out how to edit the package.yml, edits would result in errors on my computer.

For next week, I want to fix the weird error and get some tests up.

Things to try to resolve the error, try to reproduce it on normal ubuntu. If you can't resolve it, report it to yesod.

A.7. Meeting 7 - 23rd November 2017

A.7.1. Meeting Notes:

I've resolved the random error we had last week.
I've imported hashdb and have added functionality for users to create accounts and login on the yesod site.
Yesod forms rely on bootstrap 3, so downgraded from bootstrap 4 (beta) to 3.

For next time...
I want to figure out how to concatenate a Text data variable in Yesod. Have to figure out how to deal with overloaded strings?
Finish the user authentication functionality. Show appropriate messages and add extra validation to the yesod form (unique user and email, min and max length of fields).
Create tests for the user authentication functionality.
Change the forms on Django to use their form model rather than a HTML form. This will let me compare the pros and cons of Django's and Yesod's forms.
If there is time, add functionality to allow users to post messages. These messages should be saved in the database so that the user can see all the messages they've posted when they log in.

The user post message page should use ajax so when they post a message, the part of the div will just reload rather than the whole page.

A.8. Meeting 8 - 14th December 2017

A.8.1. Meeting Notes:

On the yesod site:
Have some tests working
Users can post messages, be signed up, see other users messages
Have some tests working, this is WIP

For next time...
Get Django messages working
Try to get ajax working on both sites, see https://www.yesodweb.com/blog/2013/02/ajax-with-scaffold

Interim report plan

- Intro
- Explain the choices of yesod and django
- Do some initial comparisons of the site
- My experiences with developing on both sites, what I found easy and hard on the different frameworks.
- Advantages and disadvantages of both frameworks.

A.9. Meeting 9 - 1st February 2018

A.9.1. Meeting Notes:

Worked mainly on the Django site. I have the messages working and have began comparing features between two sites such as

- The implementation of Handlers/Routes
- The way you can pass variables to templates
- How Haskell's 'maybe' reduces the number of errors you need to catch
- the ways you can implement AJAX in both frameworks

In the near future, refactor the messages implementation to use AJAX for retrieval of messages and creating new messages. This refactoring will help compare the ease of modifiability of both of these frameworks.
Whenever you come across a difficult error, try to compare the process of debugging in both frameworks.
Remember to focus on using different parts of the framework than just implementing new features on the site.
Try to resolve the textarea problem. If you can't send a screenshot of the error.

A.10. Meeting 10 - 15th February 2018

A.10.1. Meeting Notes:

Created AJAX functionality for getting messages

Resolved issue with using single template for profile by declaring the form stuff even if isCurrentUse is false, this is fine

What needs to be done

Add more tests this weekend for both frameworks. Does Haskell's type checking mean we need fewer tests compared to Python?

What to look at for evaluation

Evaluate:

- The ease of writing tests

- In python, you need lots of testing because there's no static type checking
  - Does this mean you need less tests in Haskell
  - Does this mean tests are easier to write in Haskell, or in Python because tests are more important in Python so they'd be easier to use
- What types of tests are important in the haskell world
- Some tests are unneeded for Haskell
- Is it cheaper to build a bullet proof app in Haskell or Python, maintainability? etc

Amount of users in Django makes it easier to find problems that others have experineced

Amount of users in Django means more tools for Django but this is improving on the Haskell on the side

The Yesod book written by the creator of Yesod is pretty good

Some of the problems written by people using Yesod are more detailed, users are probably more experienced in the programming world? more academic?

Evaluate the ease of adding a new feature after the site is complete

Evaluate how quick it is to debug something

Built in compiler and type checking in Haskell is very useful when debugging

Evaluate page load speeds, is Python slower because it runs the interpreter every time?

Scalability if you can

Some quantitative data?

Explain to the reader why some things make a big difference

- How long it takes to get a reliable application
  - Length of tests? Number of tests? Instances where types catch important errors? These are very useful
  - Times where the type checking or other similar features got in your way?
    - E.g. profile page was easier to code in Python, this was because of Haskell checking the scoping

A.11. Meeting 11 - 22nd March 2018

A.11.1. Meeting Notes:

Implemented type safety for some URLs

- How much does it help with testing / debugging?
- Does this reduce the amount of code you need (to check parameter types, etc.)

Joins cannot be done with Yesod Persistent (alternative pseudo SQL available) but restructuring logic may be more efficient than joins.

Plan for the holidays and beyond

- Finish the functionality of both websites by the end of week 2 of the holidays
- Produce a report by the end of week 3 of the holidays
- Get feedback and produce another version of the report the first week back from the holidays
- Get more feedback and produce a final version of the report the second week after the holidays

A.12. Meeting 12 - 19th April 2018

A.12.1. Meeting Notes:

Report draft, hand in by this weekend.

Book in a meeting on Tuesday to discuss the report.

Simulate mistakes and see how the haskell compilers help

Argue against how the compiler may make you write unnecessary code

- Haskell version, I couldn't do something like if not null, ignore the block
- You can try to use an undefined and force an exception
- You can do something like "Error - (loc) shouldn't happen"

Type safety saved a lot of time with checking variable types

But people like python because of the lack of type checking

It may be faster to ignore types when developing but it could cause problems later on

In Yesod, there's types for entity Ids, and if the entity Id does not exist, it will 404

Think about seperating language vs framework

For this report, it's more useful to focus on the framework, but think about anything to do with the language

For example, static type checking is to do with Haskell rather than Yesod

In the yesod-test package, you can use HTML selectors. That makes testing static content easy.

Django, you can test HTML page but selectors are not implemented. (If html page contains this string).

If you have time after getting the report done...

Load testing, deploy both apps to the same environment. Is there a tool for it. Send multiple queries from multiple machines, time responses.

If no tool, you can try requests from just your own machine. If you can't get it on Heroku, set up a VM.

Test the ease of deployments for both frameworks. Test the actual production mode, not dev mode.

A.13. Project Definition Form

A.13.1. 14th October 2017

First draught written up and sent to tutor via email for feedback

A.13.2. 15th October 2017

Tutor feedback implemented

A.13.3. 19th October 2017

Tutor and I signed form. Form is submitted electronically via Turnitin

A.14. Interim Report

A.14.1. 18th January 2018

Interim report started and then finished

A.14.2. 19th January 2018

Sent Ethics form to tutor

Proof-read and submitted the interim report

A.15. Software Development

This section records the development lifecycle of the Django and Yesod sites. More detailed commit messages can be found by running a *git log* on the git repo for both sites.

A.15.1. Yesod Site

1st November 2017

Created a basic homepage and a vagrantfile.

23rd November 2017

Implemented Login and Signup functionality

11th December 2017

Created a profile page

14th December 2017

Implemented message creation functionality and added some tests

15th February 2018

Modified profile page to retrieve messages using AJAX

19th March 2018

Implemented functionality to show other registered users on the profile page

20th March 2018

Basic functionality to follow users implemented

21st March 2018

Refactor profle page and follow functionality to use type safe URLs

22nd March 2018

Added functionality to allow users to follow eachother through normal usage of the site. Previously, users had to type in a URL to follow a user

31st March 2018

Refactored profile page. Separated pages used for logged in and anonymous users, reducing complexity. Refactoring also increased use of type safe URLs in the profile page and when posting messages.

1st April 2018

Updated stack resolver to latest point release for current resolver, ensuring we have the latest compatible updates for our packages. Fix some bugs and add tests.

2nd April 2018

Added tests for all route handlers and integrate tests with Travis CI.

4th April 2018

Implemented search functionality

5th April 2018

Tested the search functionality

6th April 2018

Added latest posted messages to the homepage and used a previously created datatype to help display posted messages on the frontend.

7th April 2018

Added hashtag functionality and fixed some bugs on the profile page. Hashtagged messages appear on the homepage.

10th April 2018

Added yesod prefix to database names so that the Yesod server can be ran the same time as the Django server.

11th April 2018

Removed unneeded visit button on profile page.

13th April 2018

Improved ordering of messages posted by other uses on your feed.

A.15.2. Django Site

9th November 2017

Created the homepage and a vagrantfile.

14th November 2017

Implemented login and signup functionality

15th November 2017

Refactored authentication functionality to use Django's built in models rather than custom models.

31st January 2018

Implemented profile page and message creation functionality.

10th April 2018

Downgrade Django bootstrap version to bootstrap 3. This is the version used by Yesod so using version 3 reduces the amount of template work to do which does not provide much insight into the benefits of either framework.

Implemented breadcrumbs and a base template file that all other template files use.

11th April 2018

Added AJAX functionality to the profile page.

12th April 2018

Completed the profile page and added search functionality

13th April 2018

Added tests and fixed any bugs discovered. Latest message and hashtagged messages added to the homepage.

A.16. Project Diary Final Report

A.16.1. 20th April 2018

Started write-up of the final report

A.16.2. 22nd April 2018

Completed report and sent to tutor for feedback

B. Ethics Form

Ethics form for student projects

SEAS group: Computer Science

Project title: Evaluation of a Haskell Web Framework

Supervisor name and email: Michal Konecny m.konecny@aston.ac.uk

Ethics questions

Please answer Yes or No to each of the following four questions:

1 - Does the project involve participants selected because of their links with the NHS/clinical practice or because of their professional roles within the NHS/clinical practice, or does the research take place within the NHS/clinical practice, or involve the use of video footage or other materials concerning patients involved in any kind of clinical practice? ***No***

2 - Does the project involve any i) clinical procedures or ii) physical intervention or iii) penetration of the participant's body or iv) prescription of compounds additional to normal diet or other dietary manipulation/supplementation or v) collection of bodily secretions or vi) involve human tissue which comes within the Human Tissue Act? (eg surgical operations; taking body samples including blood and DNA; exposure to ionizing or other radiation; exposure to sound light or radio waves; psychophysiological procedures such as fMRI, MEG, TMS, EEG, ECG, exercise and stress procedures; administration of any chemical substances)? ***No***

3 - Having reflected upon the ethical implications of the project and/or its potential findings, do you believe that that the research could be a matter of public controversy or have a negative impact on the reputation/standing of Aston University? ***No***

4 - Does the project involve interaction with or the observation of human beings, either directly or remotely (eg via CCTV or internet), including surveys, questionnaires, interviews, blogs, etc? *Answer "no" if you are only asking adults to rate or review a product that has no upsetting or controversial content, you are not requesting any personal information, and the adults are Aston employees, students, or your own friends.* ***No***

Student's signature: _____

Supervisor's signature: _____

C. Final Year Project Definition Form

Final Year Project Definition Form

Student's name
Junaid Rasheed

Course
Computer Science

Project title
Evaluation of a Haskell Web Framework

What is the project about?
Comparing a Haskell Web Framework with a more traditional framework. The two frameworks that will be compared are Yesod (Haskell) and Django (Python). A website similar to Twitter will be created with both frameworks and then both websites will be compared. The comparisons will include differences in page load speed, safety, reliability and whether type checking during compile time vs runtime helps reduce bugs, maintainability by comparing the ease of adding a new feature to both websites, and the ease of testing in both frameworks.

What is the project deliverable?
Two websites that are functionally identical, one developed using Django, and the other using Yesod. Then a report will be written comparing the reliability, maintainability, speed, safety, and possibly the scalability of both frameworks. The report will evaluate the advantages and disadvantages of making a website using Yesod.

What is original about this project?
There has not been any detailed comparisons between a Haskell Web Framework and a Web Framework in a more traditional object oriented language like PHP or Python. This project will provide enough detail to people looking into using a Haskell Web Framework to help them inform their decision.

Timetable showing main stages in work plan
End of October: Comfortable with Yesod and Django, create a simple website with both frameworks. Create tests for the simple website.
November: Start to create a simple Twitter clone in both frameworks. Ensure tests are created for new features. Debug any errors.
End of January: Simple twitter clone finished, users can make posts, follow each other, make 'hashtags' (any word with a '#' preceding itself is linkable to other posts containing the 'hashtagged' word and looking up the word will show all posts containing the 'hashtagged' word in a paginated results view.). Tests created for all features, bugs debugged.
End of February: Add a new feature. The feature that planned is a way to send private messages directly to other users. Users will have an area displaying all private messages sent and received and will be able to reply to other people's messages. Add tests for this feature and debug any bugs encountered.
End of March: Record and resolve any bugs and errors in the framework, ensure that each framework has a sufficient number of unit tests and that all unit tests pass. Compare both frameworks, with a focus on speed, reliability, and safety.
April: Begin and finish the Final Project report, prepare for live demos.

Student's signature _____ *Date* _____

Supervisor's signature _____ *Date* _____