

FYP Diary

Junaid Rasheed

Contents

1 Meeting 1 - 3rd October 2017	3
1.1 Meeting Notes:	3
1.1.1 Books	3
1.1.2 Frameworks / Tools	3
1.1.3 Comparing the two frameworks	3
1.1.4 Plan for next meeting	4
2 Meeting 2 - 12th October 2017	4
2.1 Meeting Notes:	4
3 Project Definition Form	4
3.1 14th October 2017	4
3.2 15th October 2017	4
3.3 19th October 2017	4
4 Meeting 3 - 19th October 2017	4
4.1 Meeting Notes:	4
5 Meeting 4 - 24th October 2017	5
5.1 Meeting Notes:	5
6 Meeting 5 - 10th November 2017	5
6.1 Meeting Notes:	5
7 Meeting 6 - 16th November 2017	5
7.1 Meeting Notes:	5
8 Meeting 7 - 23rd November 2017	6
8.1 Meeting Notes:	6
9 Meeting 8 - 14th December 2017	6
9.1 Meeting Notes:	6

10 Meeting 9 - 1st February 2018	7
10.1 Meeting Notes:	7
11 Meeting 10 - 15th February 2018	8
11.1 Meeting Notes:	8
11.1.1 What needs to be done	8
11.1.2 What to look at for evaluation	8
12 Meeting 11 - 22nd March 2018	9
12.1 Meeting Notes:	9
12.1.1 Plan for the holidays and beyond	9

1 Meeting 1 - 3rd October 2017

1.1 Meeting Notes:

1.1.1 Books

Real World Haskell

Haskell from first principles (haskellbook.com)

Web application development with Haskell and Yesod (out of date)

1.1.2 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.

1.1.3 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

1.1.4 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

2 Meeting 2 - 12th October 2017

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

3 Project Definition Form

3.1 14th October 2017

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

3.2 15th October 2017

Tutor feedback implemented

3.3 19th October 2017

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

4 Meeting 3 - 19th October 2017

4.1 Meeting Notes:

Carry on with the Haskell Programming from First principles book

Have some planning for the twitter clone ready

5 Meeting 4 - 24th October 2017

5.1 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

6 Meeting 5 - 10th November 2017

6.1 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.

7 Meeting 6 - 16th November 2017

7.1 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.

8 Meeting 7 - 23rd November 2017

8.1 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.

9 Meeting 8 - 14th December 2017

9.1 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.

10 Meeting 9 - 1st February 2018

10.1 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.

11 Meeting 10 - 15th February 2018

11.1 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

11.1.1 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?

11.1.2 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 experienced

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

12 Meeting 11 - 22nd March 2018

12.1 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.

12.1.1 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