Milestone 5

Changes

There have been no more changes to my project since my last milestone update.

Accomplishments Since Last Update

Since my last update, I have made strong progress towards understanding how to add refinement types to database queries. I have split the types of queries into two distinct problems: queries without joins, and queries with joins. Queries without joins have type signatures that made refinements very easy-- they usually look something like:

```
getUsersWithUsername :: name:Text -> [{u:User | userUsername == name}]
```

So we have access to every part of the query in the type signature.

For queries with joins, we are usually looking at some table that relates two other objects. An example of such a query would be:

```
getPapersForUser :: u:User -> [{p:Paper | isAuthor u p}]
```

The problem is that our “isAuthor” function needs to check if there is a relationship in our database that relates this paper and this user. In SQL databases, relationships like this are stored in a separate table. This means that this table won’t appear in our type signature, and we can’t just use isAuthor in a refinement (since only certain functions can be lifted into the refinement logic). I have not yet figured out how to deal with this problem, but know that this is my next step.

Meeting Milestones

I am slightly ahead of where I expected to be at this point in time. I am already well into my next set of milestones, which is figuring out how to express policies in the LiquidHaskell refinements. If needed, I will revise my milestones after Spring break to reflect my current progress.

Looking Ahead

Over the next two weeks, I will be figuring out how to express more complex policies in the refinement logic. The first of which is the second type of problem described in my accomplishments section. After that, I want to figure out how to propagate assumptions about the current session user as assumptions in the Liquid Haskell refinements. Since webpages in Yesod can only be accessed if an isAuthorized check returns Authorized, we know certain things about the user if we are in the body of a function that responds to that request. Both of these problems should occupy a good amount of my time.

Resources Needed

I require no additional resources to continue working on my project.