## Milestone 2

### Changes

As mentioned in my previous milestone update, my project has completely changed since originally proposed. Instead of inferring policies in code, I am writing a web framework that will provide strong static safety and security guarantees by using LiquidHaskell's refinement type system. When errors are found in the code, we hope to repair them automatically by extending the work done on LIFTY. I include an updated list of milestones at the end of this update.

### Progress

At this point, my goal was to have finished writing the SIGBOVIK conference management system, as well as get an idea of how the web framework we are using generates queries to the database. I am currently on track, and you can find the SIGBOVIK conference management system here: http://sigbovik.herokuapp.com/. I am currently writing unit tests to ensure that the security policies are enforced in the conference management system.

### Updated Milestones

| | |
|---|---|
| 1/30/2017 | ● Finish SIGBOVIK conference management system<br>● Pick a few examples of the places where queries are generated, and figure out what we need to check<br>○ find an easy, medium, and hard query |
| 2/13/2017 | ● Write tests for Policies in conference management system<br>● Try to verify Yesod projects with LiquidHaskell<br>● Understand how database queries are generated in Yesod |
| 2/27/2017 | ● Understand how to express a small number of policies in Liquid Haskell (not in Yesod)<br>● Deploy conference management system<br>● Tests for conference management system |
| 3/20/2017 | ● Figure out what policies would look like in Yesod w/ Liquid Haskell<br>● Figure out what it means to be a policy in the database and how these policies will be verified |
| 4/3/2017 | ● Figuring out how the database queries fit into the type system of LIFTY |
| 4/17/2017 | ● Understand how repair would work with the database<br>● Repair using liquid haskell |

| | |
|---|---|
| 5/1/2017 | ● Generate code for accessing DB that has records with tagged fields |