Alloy and Formal methods of Web development

Virtual Labs
An MHRD Govt of India Initiative

VLEAD
Virtual Labs Engineering and Architecture Division

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

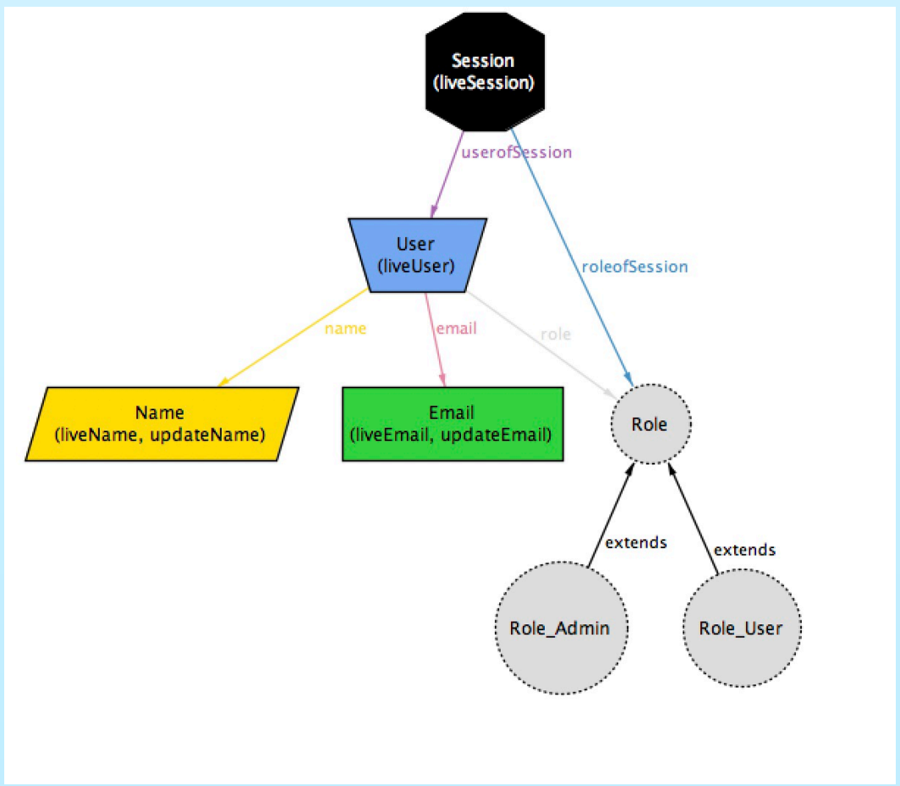
AIM

- To show the advantages of modelling during the development of an application.
- More emphasis on the requirements.
- To remove the bugs directly during development process rather than during the production of the app.
- To show that modular development helps in pipelining and automation.

INTRODUCTION

<u>Alloy and formal methods of web development :</u>

<u>This Web-application</u> is a simple *user* directory.It allows users have role, name, email. This app has many functionalities such as add, update, delete show users etc,. It is written in python.
<u>Alloy</u> is a tool for relational modelling which uses expressive login based notations.
We used alloy to show the various *constrains* that a application model faces while designing it. *e.g.,1.Can users have same name or email? 2.How many roles a user can have?*

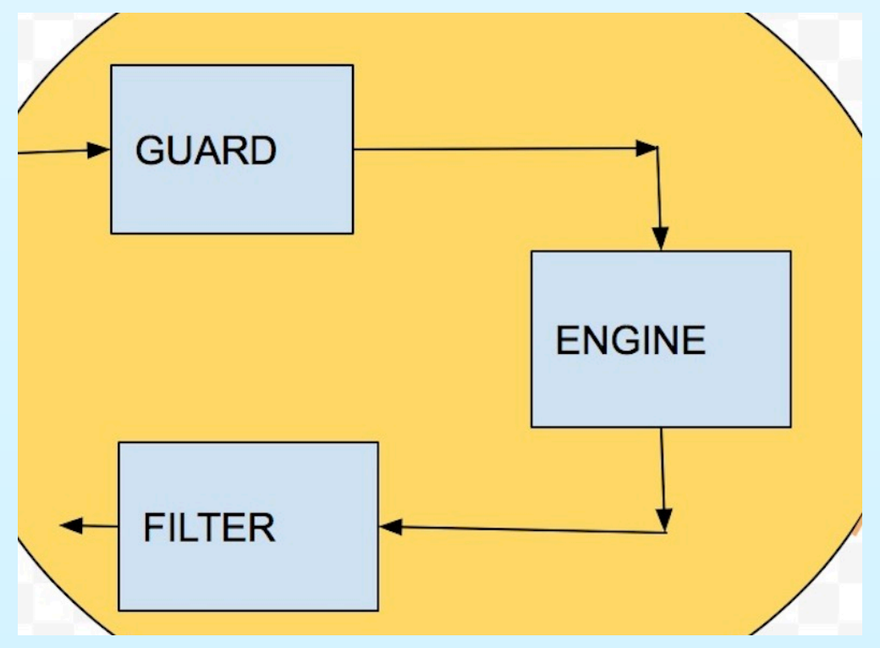


-The Alloy Analyzer generated metamodel of the app showing the relations of all the entities in the app

<u>Motivation:</u> to model our app in a specification language(like *Alloy*) is that it helps us in checking the correctness of our data-model before actually writing the real code. Also, the client may ask us to add new requirements or constraints and there is a necessity to check whether the new constraints don't *break* the constraints of the previously present requirements.

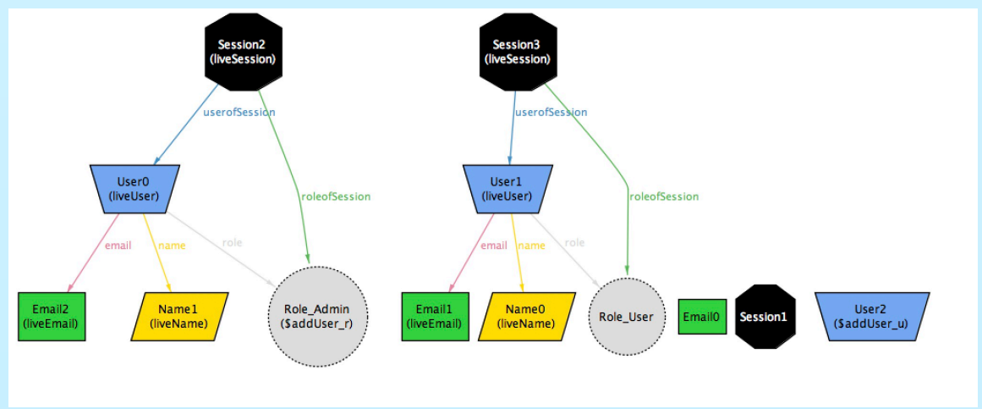
METHODS

<u>Routing:</u> All the operations in the app follow a workflow, i.e, an instruction is generated from the REST API and sent to the components. The app has 2 components,

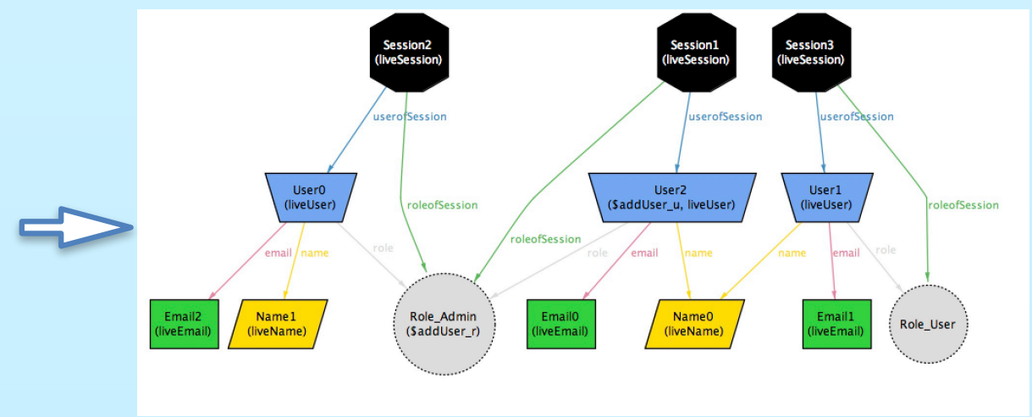


<u>Guard</u>:This does all the checks such as type check, auth check and state check depending upon the type of instruction.
<u>Engine</u> :This does the operations on the entities and returns the result.

<u>Alloy Model</u>
<u>Add user:</u> This example shows user being added by the admin in system.

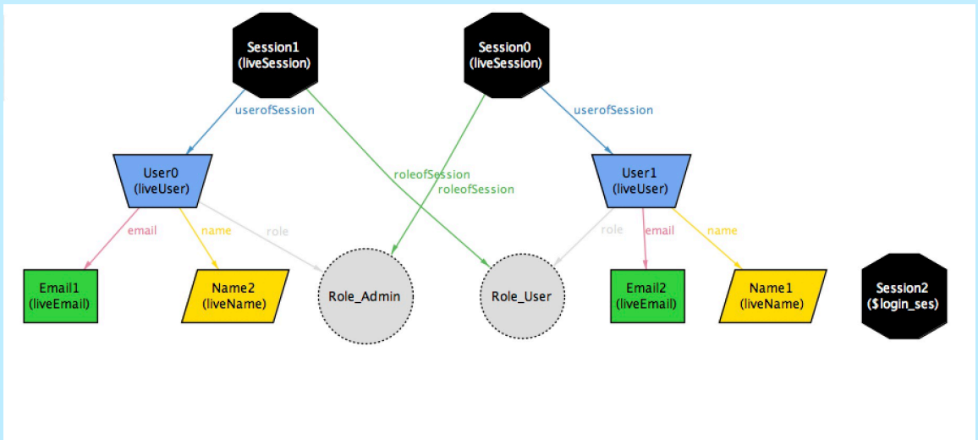


- before the user is added

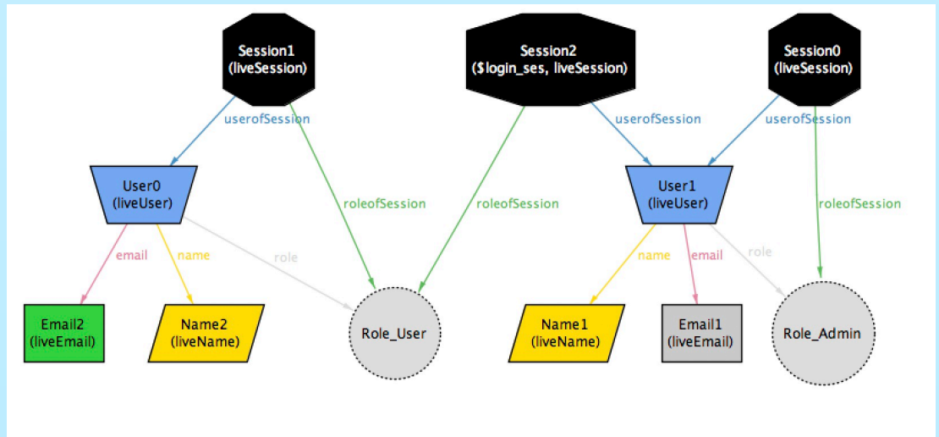


after the user is added

<u>User Login :</u> This example shows user login inside of the system.

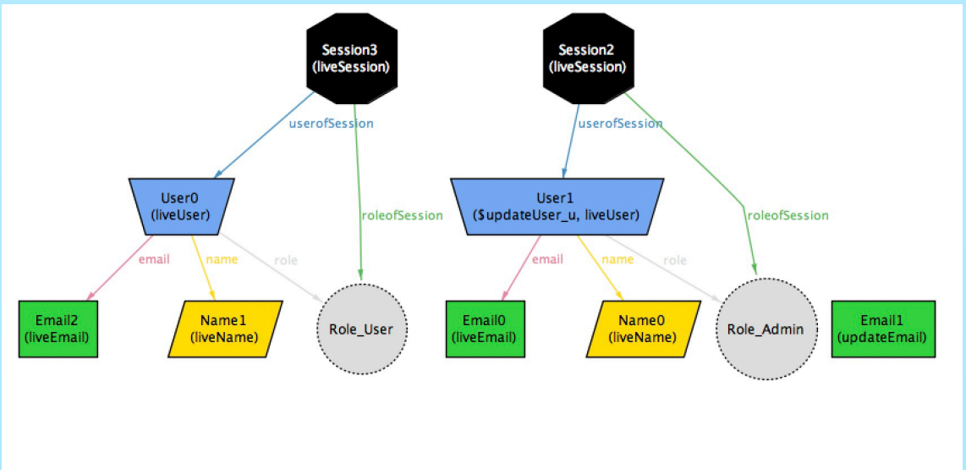


- before the user is logged in

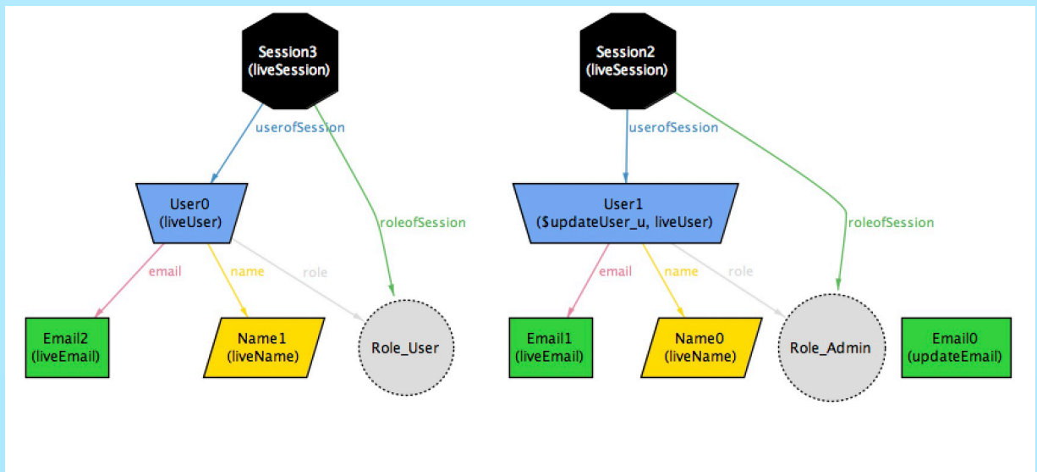


after the user is logged in

<u>Update user:</u> This example shows user credentials being updated by the admin in system.



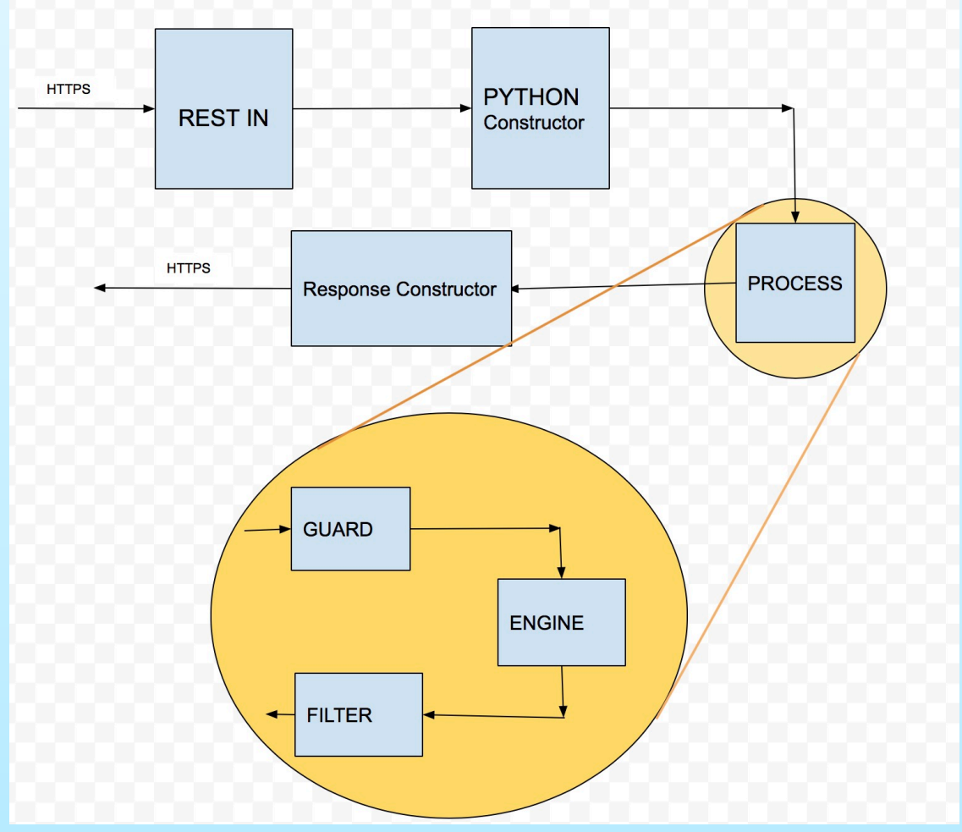
- before user is updated



after user is updated

MODULAR DEVELOPMENT

<u>Modular development</u> is a technique which is used to break down our app into independent modules which can perform a task. We combine all such modules in a certain order(i.e., we generate a *workflow*) to get a certain operation done.



workflow of the app

RESULTS

- Completed all the necessary routes(12 routes) and their tests (totally 114 tests)in the web-app-short-course.
- The routes are successfully built modularly.
- Relational modelling helped us in defining a better database for the app and also helped in achieving the requirements without any bugs.

CONCLUSIONS

- Our work shows how using a specification language to model our app helps us in understanding the requirements better.
- Modelling in Alloy and modular development helps in automating the process of developing an application by giving some inputs and requirements.
- Modular development helps in achieving pipelining as each module is independent and can be used according to the operation.

SERC
Software Engg Research Center

Developers name: Anirudh Reddy, Shubh Maheshwari
Mentors : Venkatesh Choppella, Thirumal Ravula