Client/Advisor: Dr. Myra Cohen

Automated Configuration Testing Framework for KBase

SDMay 21-26

Jake Veatch, Daulton Leach, Daniel Way, Sergey Gernega, Hunter Hall, Caleb Meyer

Motivation:

- Our senior design project is focused on automating interactions in a predictive biology tool named Kbase.

Problem:

- Manual configurations through the KBase interface take minutes to run and have no elements of concurrency.
- An exponential amount of input combinations make KBase difficult to test.

Solution:

- Use Selenium to automate KBase interactions
- Generate output files to flag errors for KBase Developers to easily reproduce.

Intended Users:

- KBase Developers
- Biologists

Intended Uses:

- Run numerous tests and flag errors for replication
- Automate Biologist's simulations to increase efficiency

Functional Requirements:

- Configure Selenium to interface with KBase apps
- User sets parameters to automate using Selenium
- App will sample the configuration space from set parameters
- Automate the testing of sample configurations
- Collect output data from KBase
- Users will be able to randomize inputs
- All inputs from the KBase GUI will be available

Non-Functional Requirements:

- Will not noticeably disrupt KBase traffic flow
- App GUI will have similar layout as KBase GUI

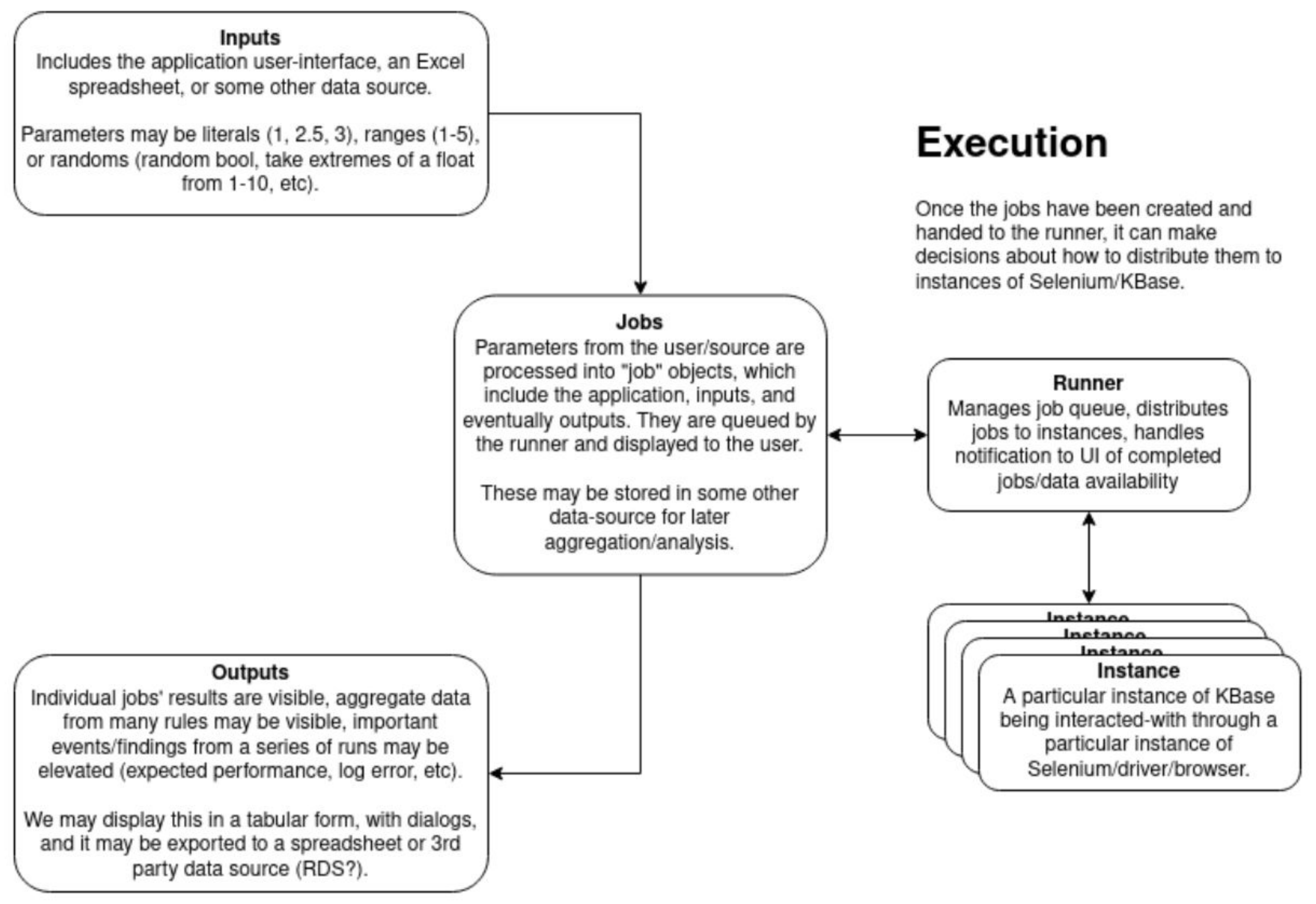
Engineering Constraints:

- Have to use selenium to manipulate GUI, cannot use KDA.
- Unable to modify KBase in any way
- Must support all Operating Systems and other Platforms.

Operating Environment:

- Will require a valid and active KBase account
- Must be run on device which supports Selenium
- Standalone application that can be executed from the users desktop

Block Diagram:



Inputs
Includes the application user-interface, an Excel spreadsheet, or some other data source.

Parameters may be literals (1, 2.5, 3), ranges (1-5), or randoms (random bool, take extremes of a float from 1-10, etc).

Execution

Once the jobs have been created and handed to the runner, it can make decisions about how to distribute them to instances of Selenium/KBase.

Jobs
Parameters from the user/source are processed into "job" objects, which include the application, inputs, and eventually outputs. They are queued by the runner and displayed to the user.

These may be stored in some other data-source for later aggregation/analysis.

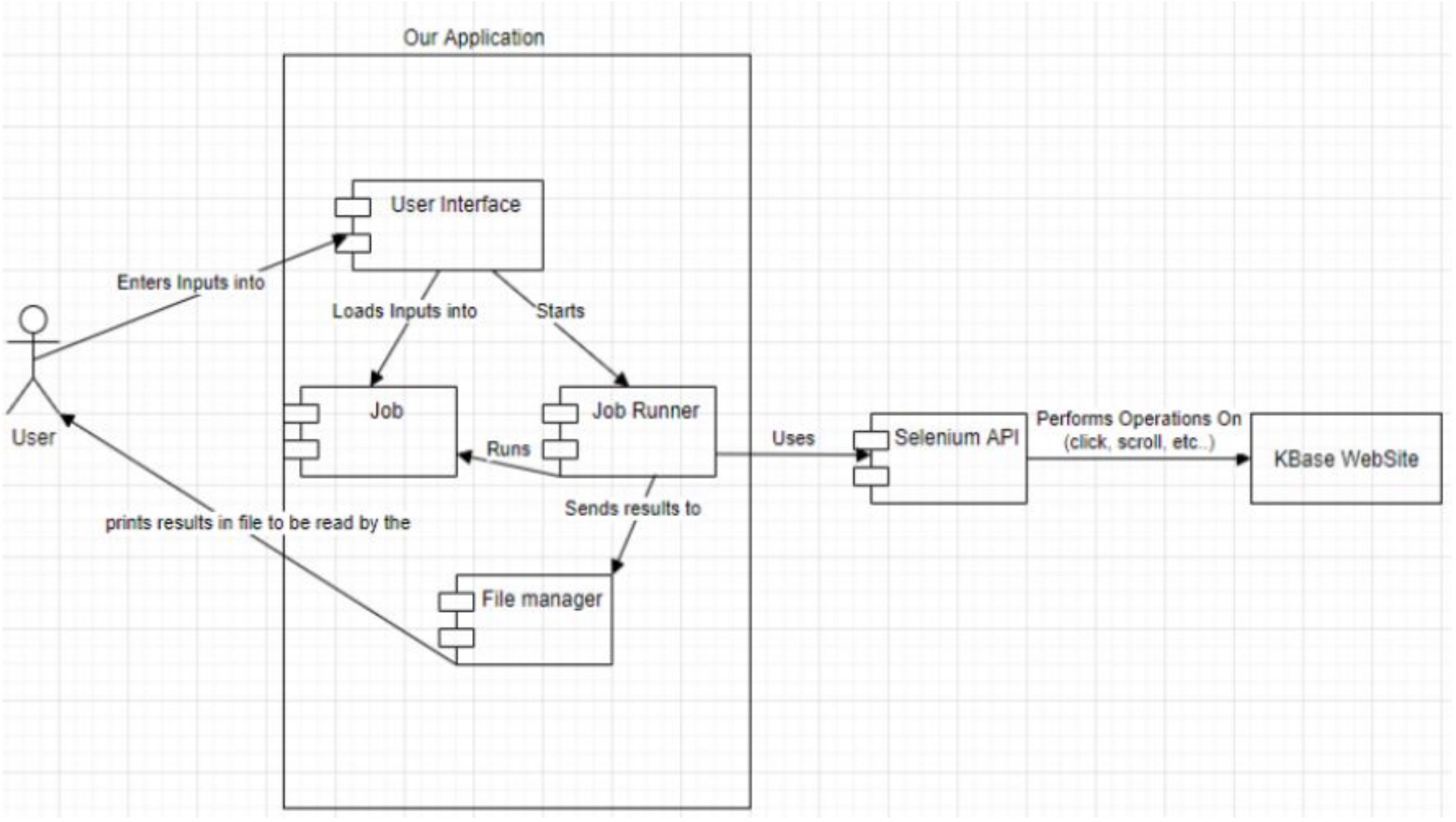
Runner
Manages job queue, distributes jobs to instances, handles notification to UI of completed jobs/data availability

Instance
A particular instance of KBase being interacted-with through a particular instance of Selenium/driver/browser.

Outputs
Individual jobs' results are visible, aggregate data from many rules may be visible, important events/findings from a series of runs may be elevated (expected performance, log error, etc).

We may display this in a tabular form, with dialogs, and it may be exported to a spreadsheet or 3rd party data source (RDS?).

Use Case Diagram:



Technical Details:

- Programming Languages: Java, Java Swing for GUI
- Framework: Selenium
- Development Environment: IntelliJ, Docker, Github (for version control/code review)

Testing:

- Environments: System Integration, User Acceptance
- Testing Strategies: Unit (JUnit5), Interface, Acceptance, Results

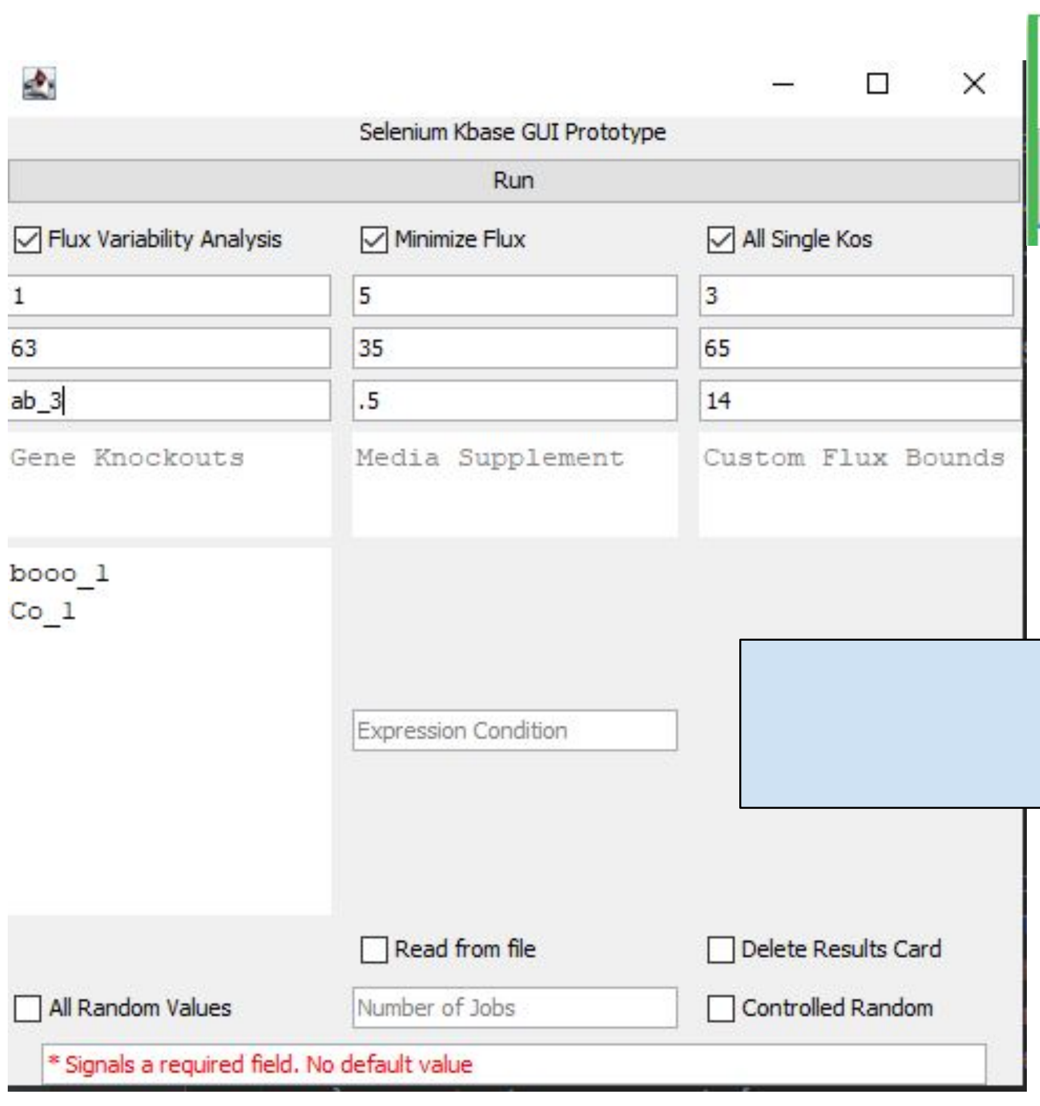
Resources:

- No external resources needed

IEEE Standards:

- IEEE 12207-2017 - ISO/IEC International Standard - Information Technology - Software Life Cycle Processes
- IEEE 1028-2008 - IEEE Standard for Software Reviews and Audits
- IEEE 2430-2019 - IEEE Trial-Use Standard for Software Non-Functional Sizing Measurements

System Screenshots



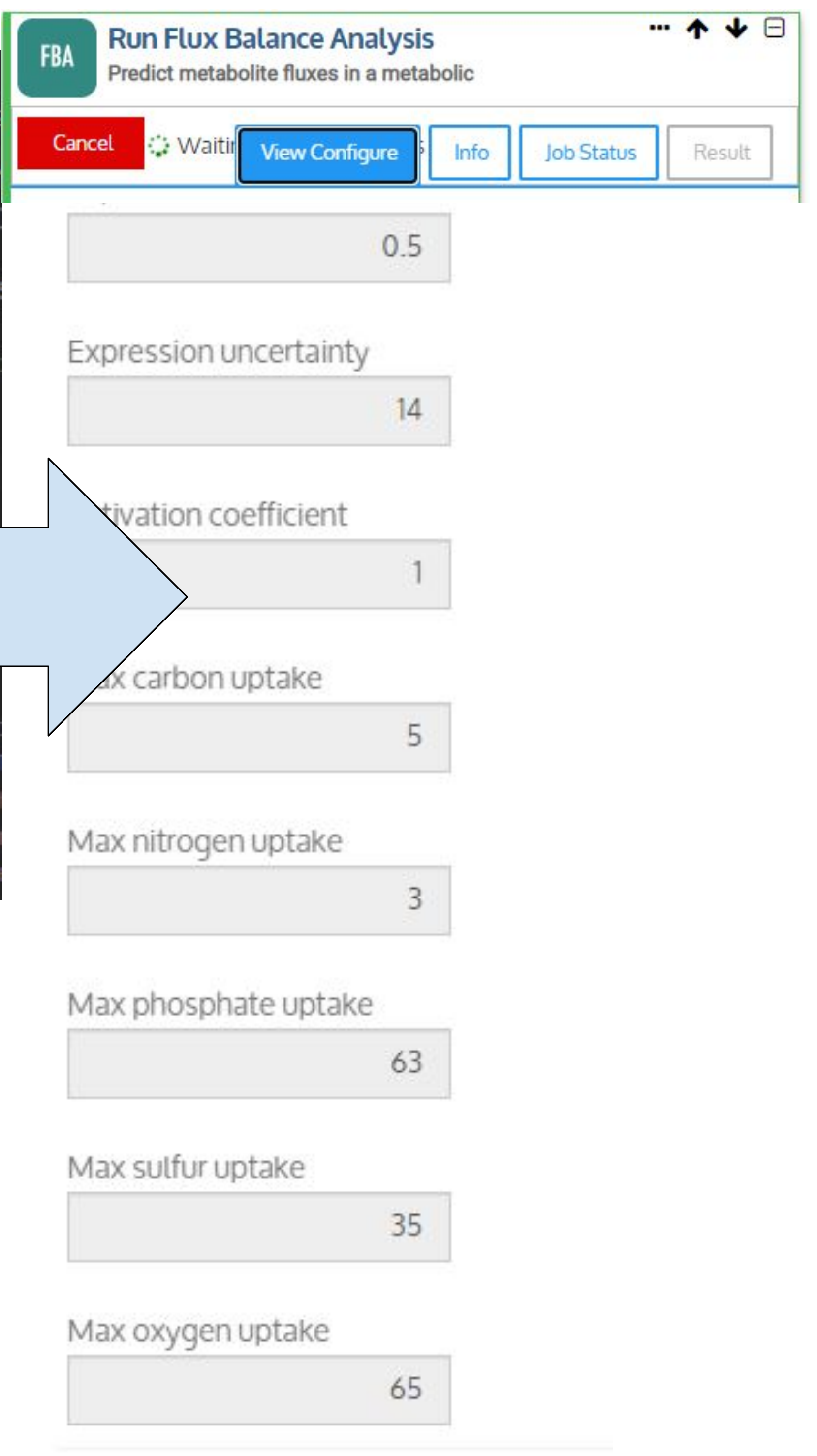
Selenium Kbase GUI Prototype

Run

☑ Flux Variability Analysis ☑ Minimize Flux ☑ All Single Kos

1	5	3
63	35	65
ab_3	.5	14
Gene Knockouts	Media Supplement	Custom Flux Bounds
booo_1 Co_1	Expression Condition	
	□ Read from file	□ Delete Results Card
□ All Random Values	Number of Jobs	□ Controlled Random

* Signals a required field. No default value



Run Flux Balance Analysis
Predict metabolite fluxes in a metabolic

0.5

Expression uncertainty: 14

ivation coefficient: 1

x carbon uptake: 5

Max nitrogen uptake: 3

Max phosphate uptake: 63

Max sulfur uptake: 35

Max oxygen uptake: 65

The user enters their experiment details (or inputs a file) and selenium automatically runs the FBA experiment and exports the output for the user