MICHIGAN STATE UNIVERSITY

Project Plan Presentation
SmartSat™ Satellite App Store

The Capstone Experience

Team Lockheed Martin Space

Aidan Delfuoco
Matt Heilman
Will Teasley
Valentino Dore
Colin Williams

Department of Computer Science and Engineering
Michigan State University

Fall 2021

From Students…
…to Professionals

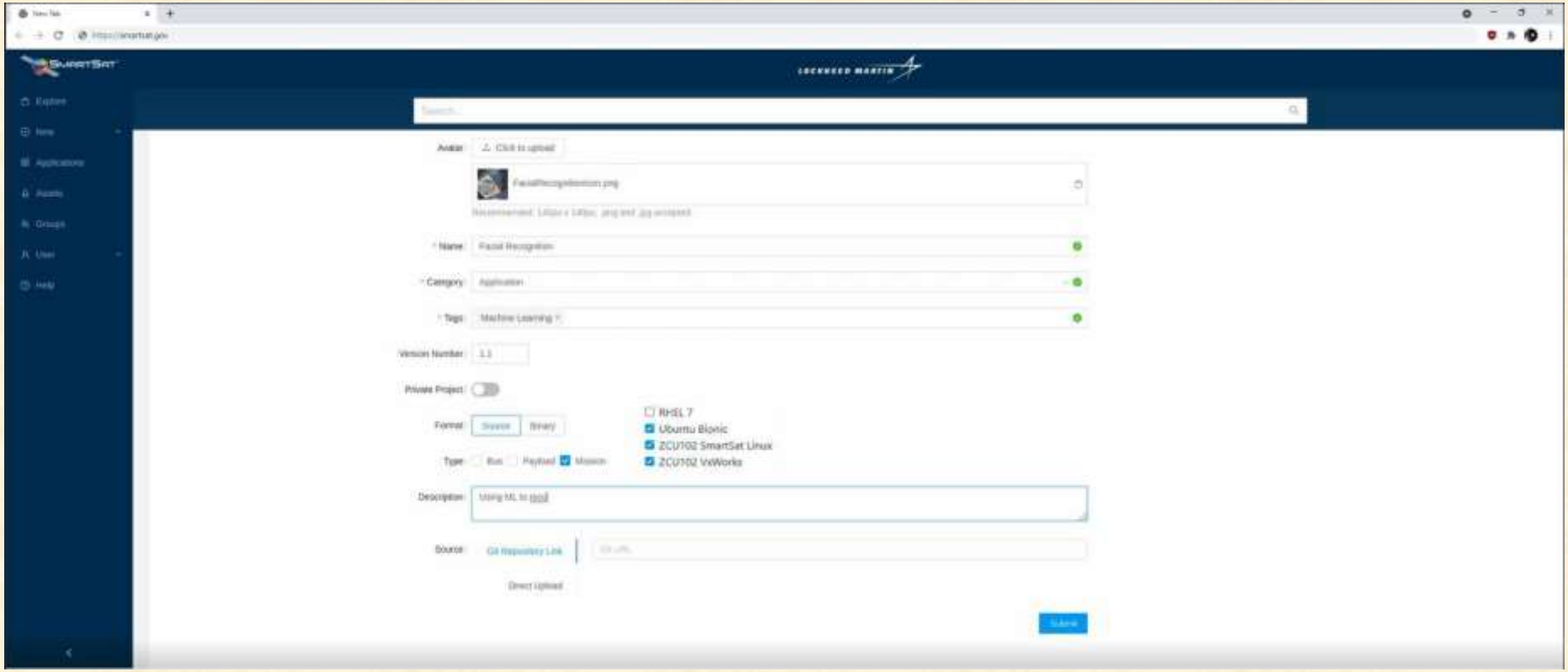
Functional Specifications

- This project will provide automated black-box hardware testing to ensure SmartSat™ application compatibility with hardware/software versions.
- The existing web application UI will be updated to specify target hardware and software configuration for SmartSat™ applications.
- Unit tests will be uploaded on the web application and test status will be shown.
- (Strech goal) RESTful API with token authentication for App Store automation and integration.

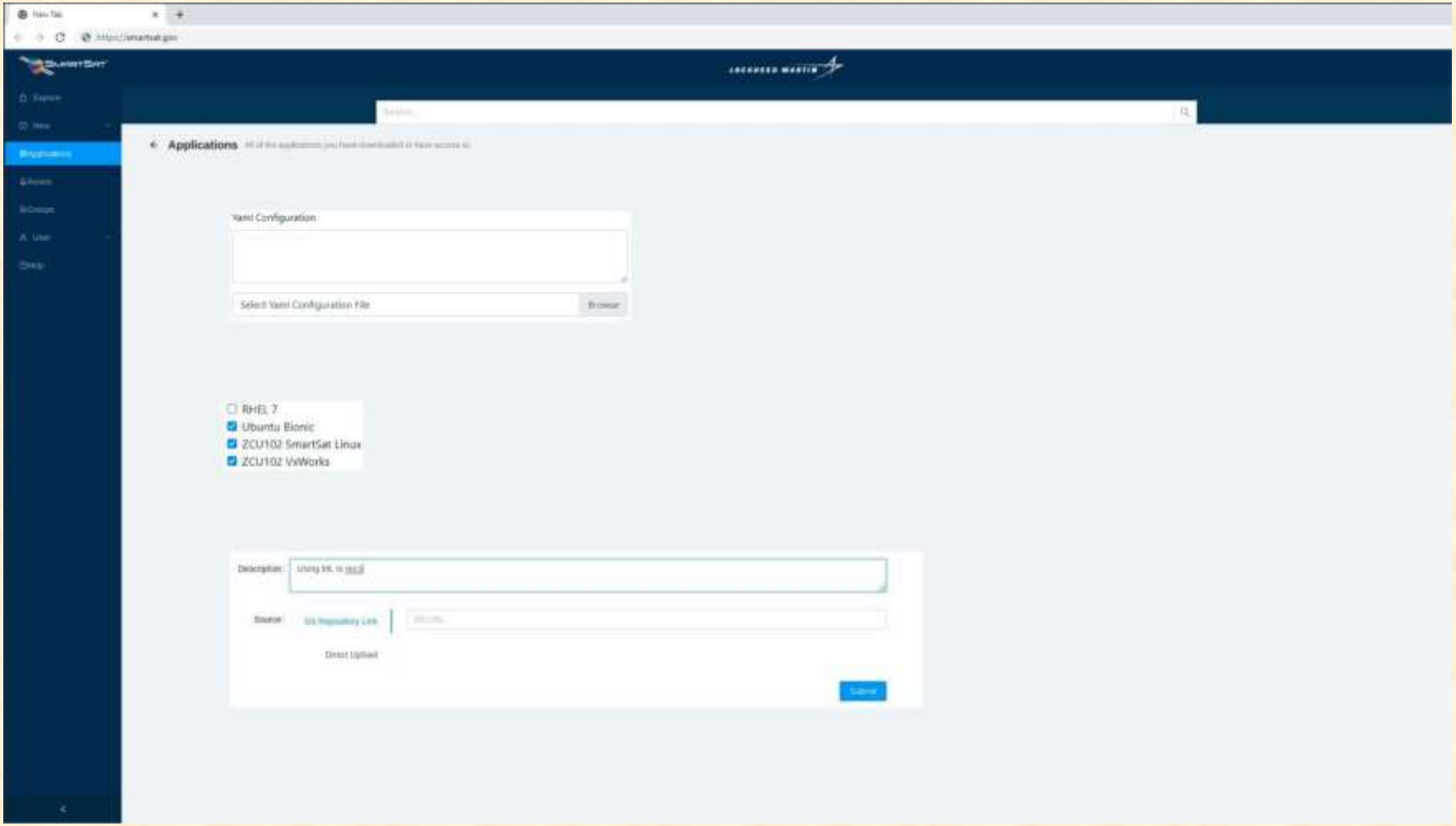
Design Specifications

- When uploading a new application to the App Store, target hardware/software configurations will be chosen from a check-box menu, with all versions selected by default.
- The web app will also have a new feature to upload Tinker Testbed forms (test classes).
- When a new app is uploaded, tests are automatically run, and the status can be viewed under the app's description.

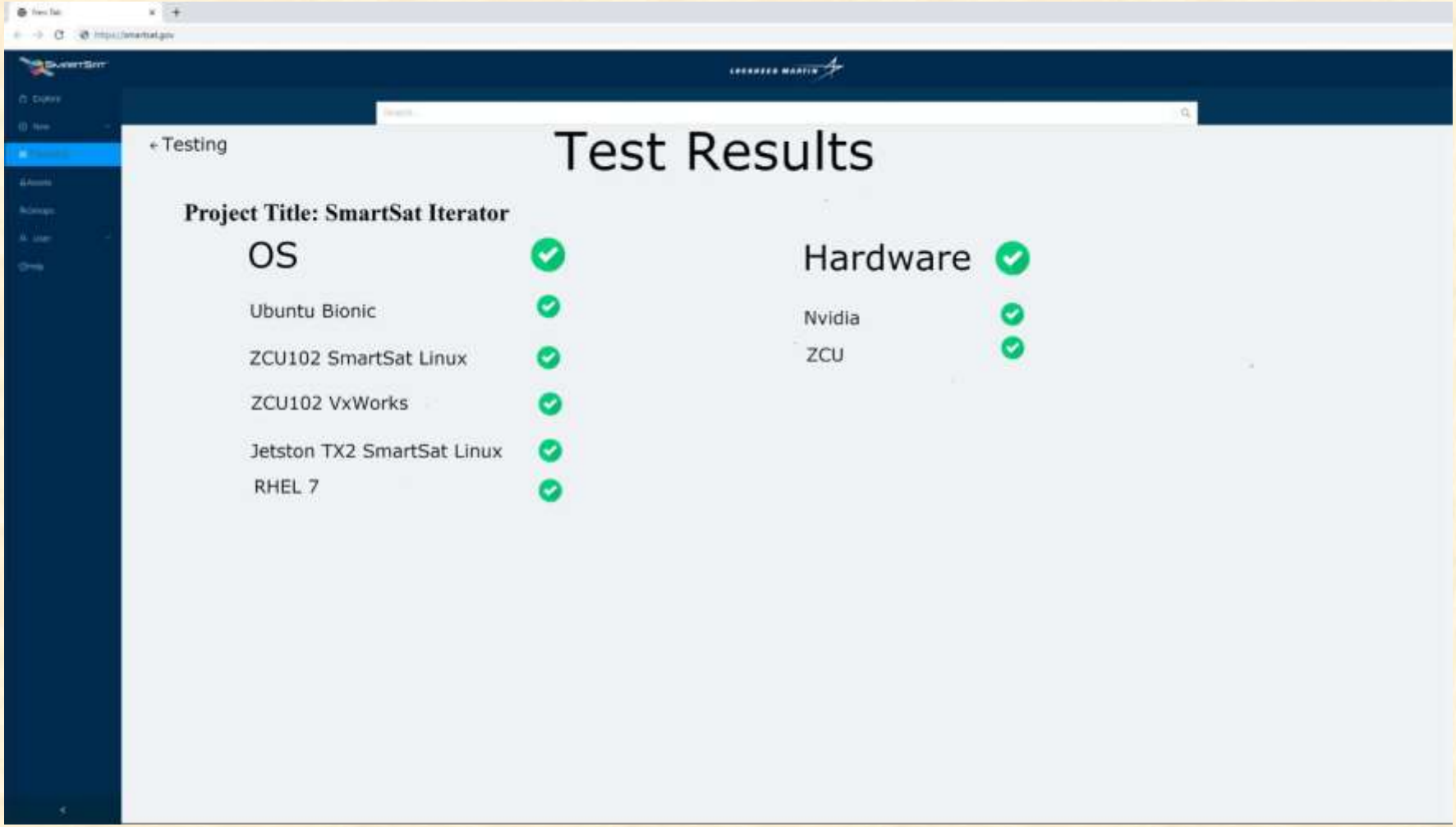
Screen Mockup: Upload New or Versioned Application Form



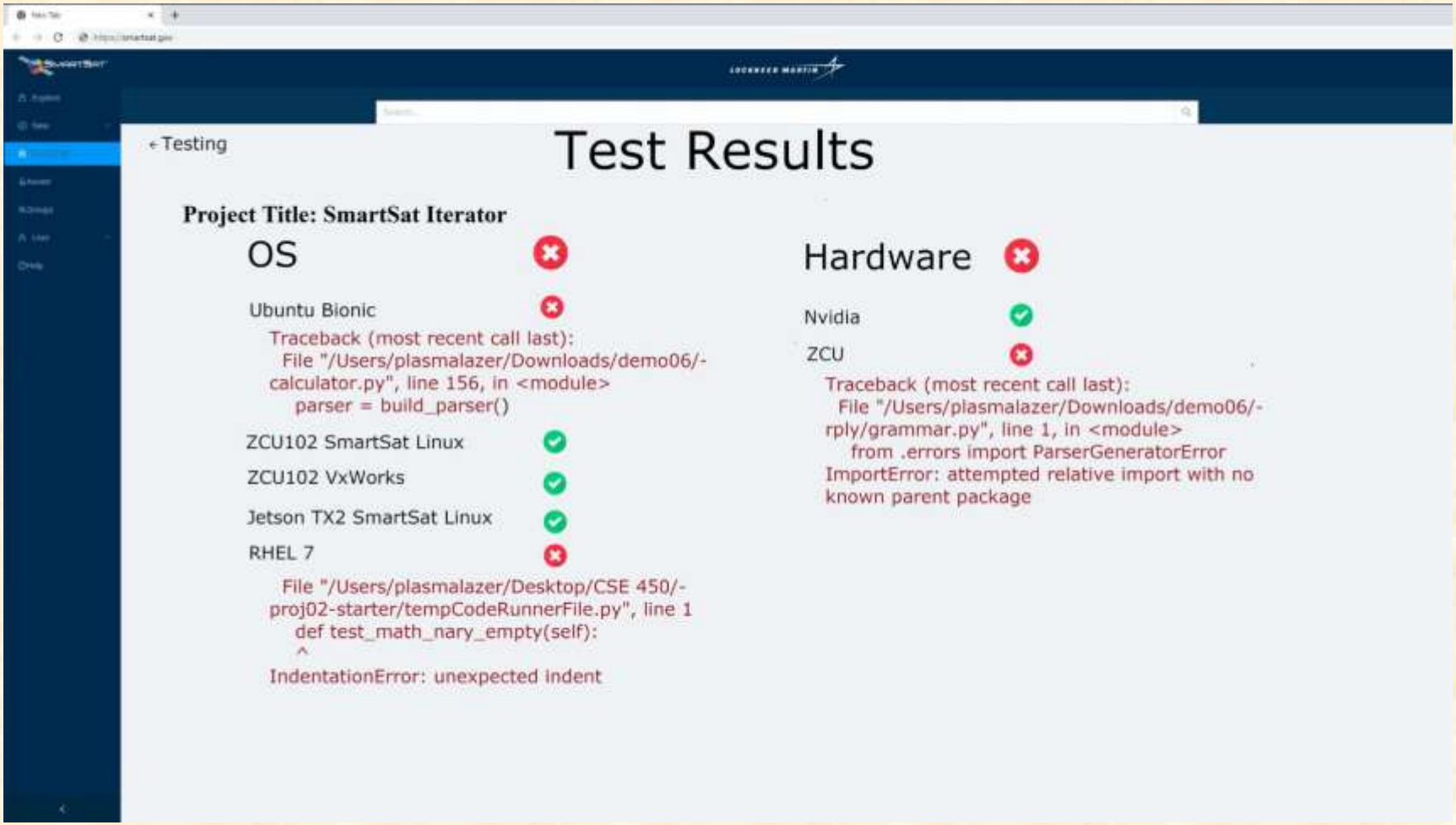
Screen Mockup: Upload Test Bed Form



Screen Mockup: Executed Test Results Positive



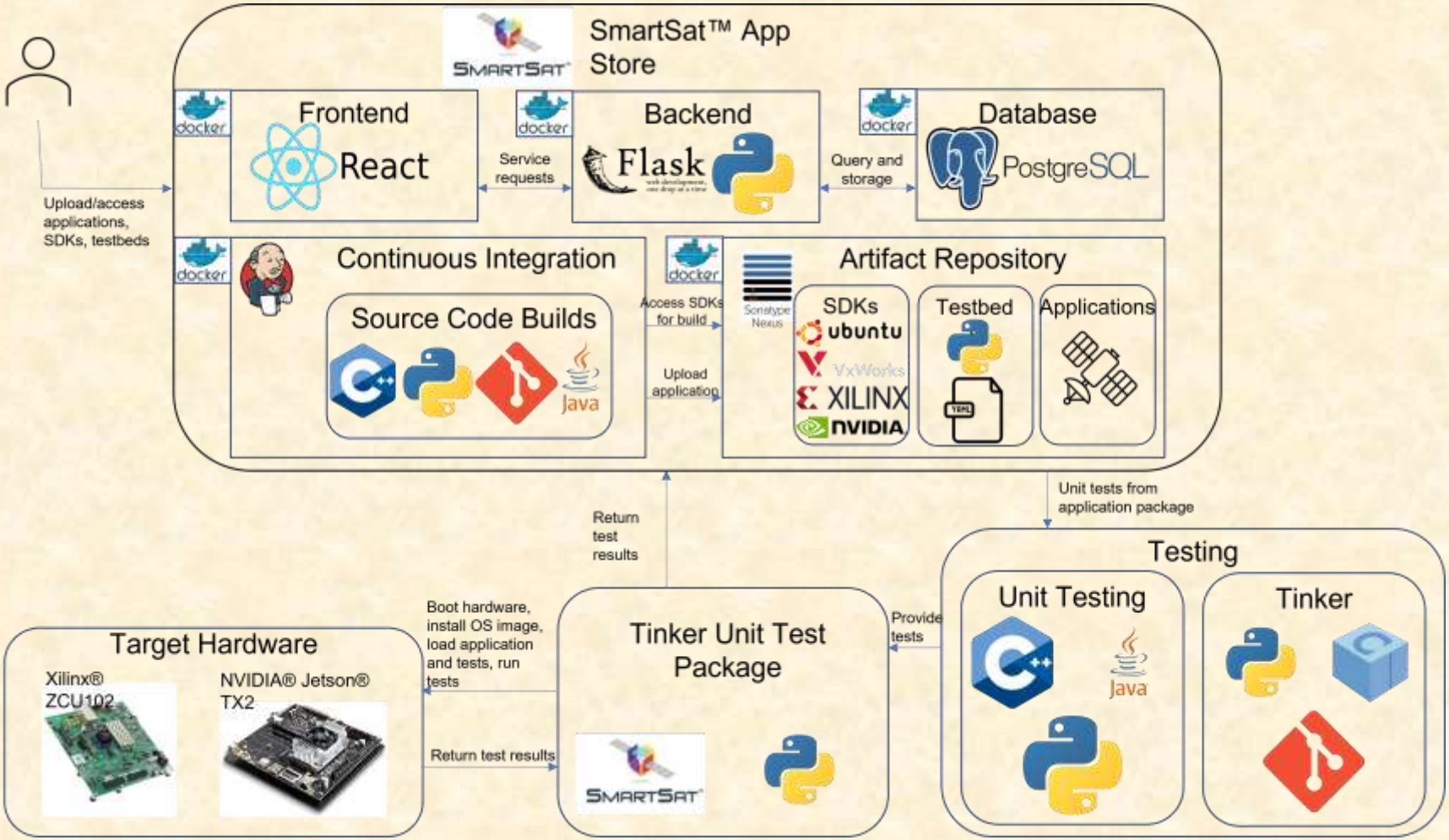
Screen Mockup: Executed Test Results Negative



Technical Specifications

- Application developers create tests using Python that interface with the applications executing on the target hardware using built-in SmartSat™ Messaging.
- The SmartSat™ app store will support the creation of new types of testbeds and provide facilities to define instances of the testbed through a YAML configuration file.
- Python package to manage the interactions between the target hardware, unit and Tinker tests, and the App Store.
- The UI changes will be made in React and Flask.

System Architecture



System Components

- Hardware Platforms
  - Xilinx ZCU102
  - NVIDIA Jetson
  - Power Unit to connect hardware in a flight-like configuration
- Software Platforms / Technologies
  - Jenkins
  - Docker
  - PostgreSQL
  - React
  - Flask
  - Conan
  - Tinker

Risks

- Applicable SDKs to test
  - Not all applications are compatible to run with all SDKs on all hardware. How can we keep track of what SDKs an application can run on?
  - We are planning to tell each Tinker testbed which SDKs are compatible. In the UI for app creation, we will ask the user what SDKs/Operating Systems are compatible in a drop-down menu and then we will have to figure out how to configure the Tinker testbed accordingly.
- Building with Tinker
  - Tinker is an Automated Black Box Integration Testing Framework for SmartSat™ Application Packages. It was developed by Lockheed Martin Space internally, so familiarizing ourselves with it will take some time. Before we can begin to incorporate Tinker functionality into the App Store, we need to understand how the tests are written and run.
  - We have received presentation slides outlining the basics of Tinker development. We will have to get familiar with the example test classes developed in Python that incorporate Tinker. The Tinker codebase has been given to us to study and start writing some simple local tests.
- Running Automated Tests
  - Unit tests must be run on all applicable SDK versions whenever an app is updated, as well as on all applicable applications whenever an SDK version is updated. This should be an automated process, and the tests need to all be run on the physical hardware. Figuring out how to push out the unit test to all the hardware serially is a risk because we are unsure how to hook up all the target hardware at one time, and how to store the compatible versions in a configuration file.
  - Using Tinker and ssh to communicate with all the different hardware versions (ZCU102 and NVIDIA Jetson), we can push out the unit tests serially. Once the response comes back from one, we can send the next test. The applicable SDK versions will be specified in a new file that will be stored with the YAML and testbed configuration files.
- Setting all hardware up together
  - We have booted up the ZCU102 and connected to it through the command line interface, but we have not determined the best way to set up the ZCU102 and NVIDIA Jetson at the same time with the power unit.
  - We will need to work with our clients more to figure out the best way to set up all the hardware at the same time.

Questions?