MICHIGAN STATE UNIVERSITY

Project Plan Presentation
SmartSat™ Satellite App Store

The Capstone Experience

Team Lockheed Martin Space

Cody Lowen
Sirena Ly
Matthew Harper
Quinton Farrar
Kaleb Koebel
Mike Kilmurray

Department of Computer Science and Engineering
Michigan State University

Spring 2022

From Students…
…to Professionals

Functional Specifications

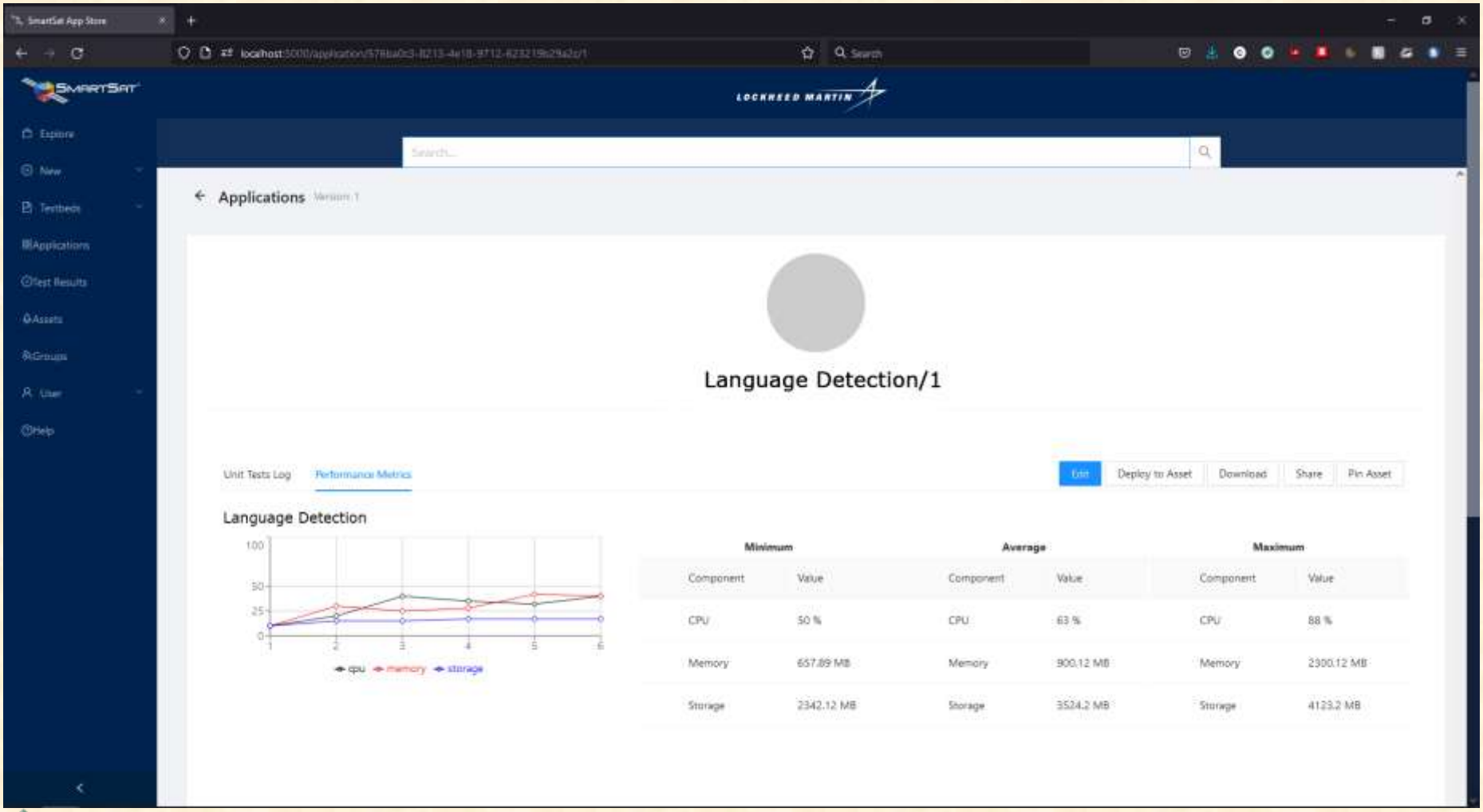
- Extend testing capabilities against new SDKs, monitor resource utilization and performance of applications
- The SmartSat™ App Store provides a platform to host and manage SmartSat™ applications and SDKs, automated testing and version control. When a new SDK or application is uploaded, all relevant tests are automatically run. The planned enhancements to the SmartSat™ App Store can be defined by two primary goals.
- Extension of the automated testing capabilities. Performance profiling and resource utilization measurement provides essential information to consider prior to deployment. This information will help operators make informed decisions when deploying software to satellite hardware.

- The ability to track and deploy dependencies. SmartSat™ applications can contain dependencies that require the proper SmartSat™ packages. This enhancement identifies an application's dependencies, creates associations in the web front end, and verifies that the required dependencies are in place before a package is installed.

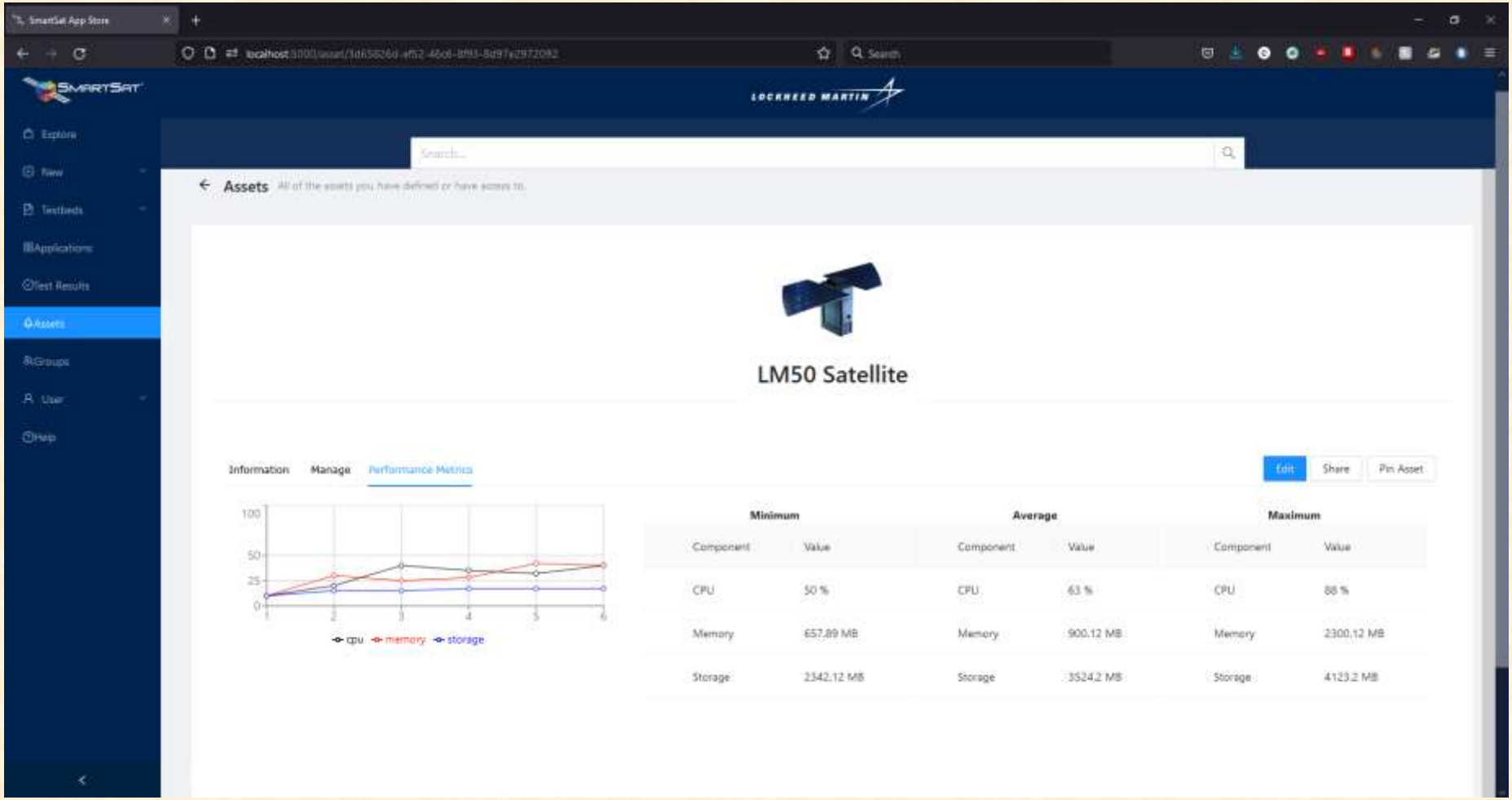
Design Specifications

- After a user deploys an application, they can access the "Performance Metrics" tab which will display CPU, Memory, and Storage utilization in the form of maximum, minimum, and average as well as time series data for these metrics.
- When uploading an application users will be notified of missing application dependencies, which will be automatically installed. If the required dependency does not exist in the app store, then the user will be prompted to upload the dependency.

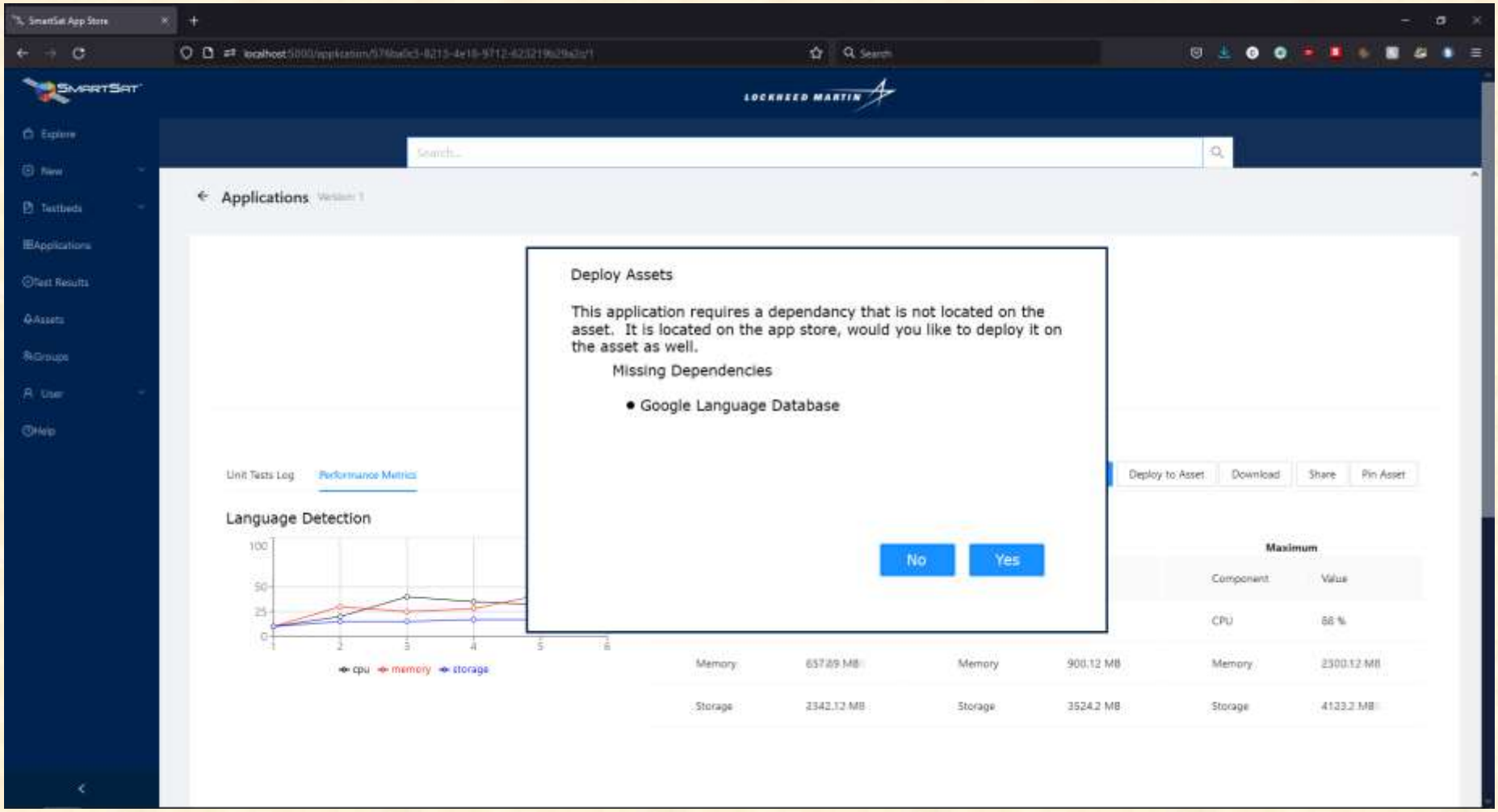
Screen Mockup: Performance Metrics for deployed application.



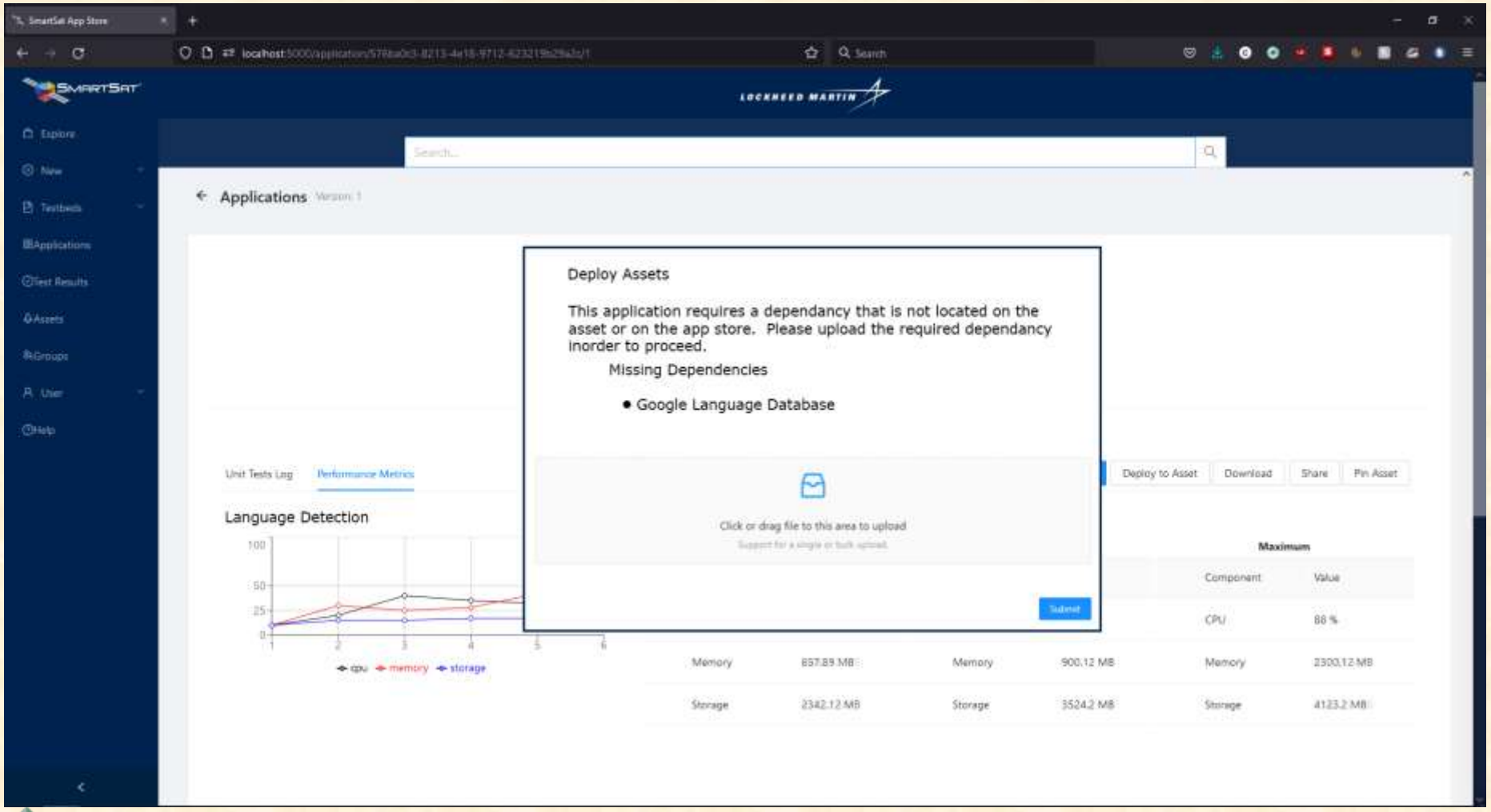
Screen Mockup: Performance Metrics for asset



Screen Mockup: Dependencies missing, but exist in app store.



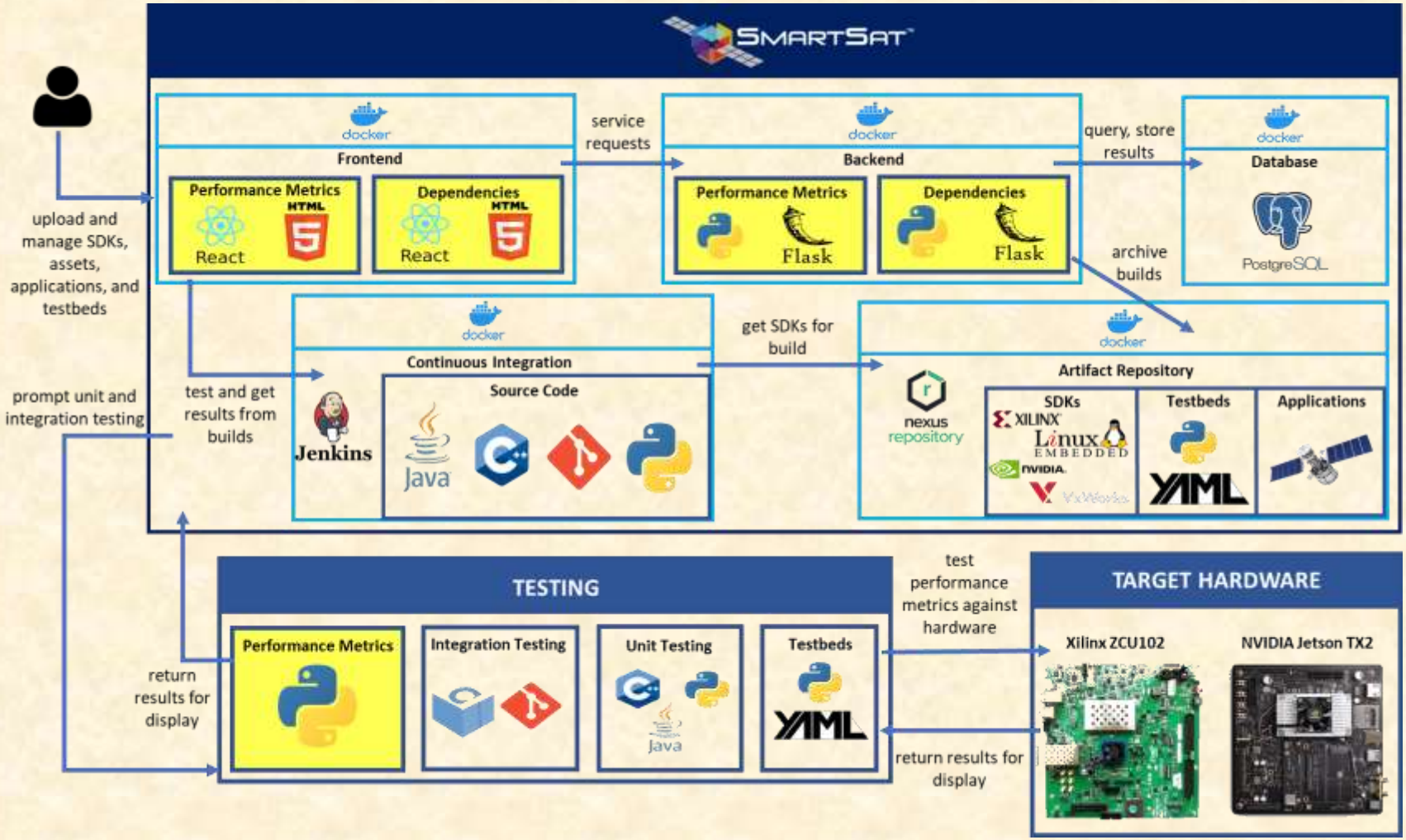
Screen Mockup: Dependencies missing, and do not exist in app store



Technical Specifications

- UI developed in ReactJS which is supported by Flask backend.
- Dependencies will be tracked by applications YAML file.
- Tinker tests utilized in backend to facilitate capture of performance metrics on target hardware.
- Flask backend will manage the installation of missing dependencies for applications.

System Architecture



System Components

- Hardware Platforms
  - Xilinx ZCU102
  - NVIDIA Jetson TX2
- Software Platforms / Technologies
  - Docker
  - Linux Ubuntu 20.04
  - VxWorks
  - Nexus
  - OpenLDAP
  - PostgreSQL
  - ReactJS
  - Flask
  - Jenkins
  - Tinker

Risks

- Fresh Installation fails to Operate
  - Several issues occurred while setting up the provided codebase on each team member's local environment. Tables within the PostgreSQL database were either missing or outdated.
  - Team members will access and develop the project on MSU servers instead of their independent Linux environments. Repairs to the database will be made on these servers, documented, and then replicated in the form of SQLAlchemy migrations
- Receiving Application Information from Asset
  - The SmartSat™ Appstore currently treats each application independently, not having to rely on dependencies from other packages. Team members need to track which dependencies correlate to the selected application to verify and deploy given dependencies.
  - Through looking at APKs metadata within packages, team members will be able to record which packages need which dependencies.
- Recreating Nexus Repository
  - The backup nexus repository was not able to be setup due to team members not being able to access the backup.
  - Team members can recreate the nexus_backup.tar file and will have to manually configure the user admin to upload desired SDKs through the app store.

Questions?