**Project Plan, Requirements Specification**

CS 499: Senior Design Project, Team 2: Tirth Patel, Jared Payne

Stakeholders: Dr. James Griffioen, Dr. Jurek Jaromczyk,

Charles Carpenter, Tony Elam

17 September 2018

<https://cs499team2.github.io>

[**1. Introduction**](#_i7t6jk30ehdp) **2**

[1.1. Project Overview](#_rbhhdee5vx0) 2

[1.2. Project Scope](#_18h5cc28jjdv) 2

[1.3. Document Preview](#_pjxwfn66niur) 2

[**2. Project Overview**](#_7cif7vyey8x6) **2**

[**3. Development and Target Environments**](#_f4p027p2x7vx) **3**

[**4. System Model**](#_o31xi26i7l5a) **3**

[**5. User Interaction**](#_3mqw85gowfrv) **4**

[**6. Functional Requirements**](#_87u3nelxdvx0) **4**

[**7. Nonfunctional Requirements**](#_d50maclwk8cr) **4**

[**8. Feasibility**](#_9kuvu1mbker) **4**

[**9. Conclusion**](#_ayaw60de6bqs) **4**

[**10. Appendices**](#_x9gqr2hbpy63) **4**

# 1. Introduction

## 1.1. Project Overview

A Jupyter notebook is an open-source product developed by a non-profit organization called Project Jupyter. It is an interactive web application which allows users to create documents which can contain code, text, visualizations, etc. Code can be compiled real-time within the document and the results are displayed for the users to view. The name Jupyter itself comes from the three core languages which are Julia, Python and R. However since its creation, Jupyter notebooks have been expanded and now support over 40 different programming languages. The versatility of the notebooks make them a powerful tool for a large number of communities, however a problem arises when sharing these notebooks. An environment has to be created on individual machines to properly load the notebook. This includes installing the necessary kernels. The purpose of this project is to solve the problem of recreating the necessary environment and make sharing notebooks easier.

## 1.2. Project Scope

Our project entails learning about technologies consisting of namely Jupyter Notebooks, JupyterHub, BinderHub, Docker, Google Cloud, and Kubernetes (among others) and to investigate how to utilize these softwares in order to easily create reproducible and shareable environments for executing Jupyter Notebooks. Although this system would initially run in the cloud (for example, on Google or Amazon’s servers), it is desired to eventually have the system run exclusively on the University of Kentucky’s resources, specifically within our high-powered computing cluster. To what degree we are expected to complete this latter task remains to be seen.

## 1.3. Document Preview

The purpose of this document is to provide information about the project and the solution that will be implemented. It will describe the various uses and potential constraints. The document is intended for the customers and potential users of the system.

# 2. Project Overview

This section will provide an overview of the solution. As mentioned before, the problem in sharing these notebooks is that an environment that can support the document has to be recreated on individual machines. Each programming language that the notebooks support has its own Kernel which requires to be installed in order for viewing. Due to their versatility and flexibility the notebooks can be a valuable tool to even non-coding communities however issues may arise when attempting to re-create the necessary environment and this process can also lead to the loss of valuable time.

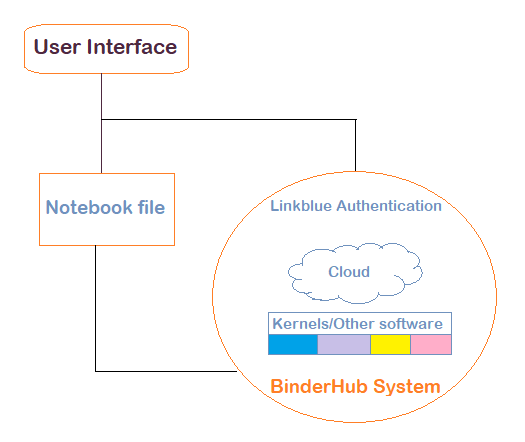
The solution that will be implemented will involve deploying a BinderHub system to make the notebooks easily shareable. BinderHub makes use of various technologies such as Kubernetes, Google cloud and JupyterHub to create a custom environment to access a notebook. The eventual goal of this project will be to deploy a BinderHub system using local cloud infrastructure rather than the Google cloud and to simplify the construction of customized environments.

# 3. Development and Target Environments

This section will list the technological resources that are necessary. Namely, they are: BinderHub, JupyterHub, Kubernetes, Docker, Google Cloud, UK cloud infrastructure, various Jupyter kernels.

# 4. System Model

The existing model is such that users recreate the environment for a specific notebook on local machines. The proposed model is that the BinderHub system constructs the environment necessary for the user and simplifies the process.



*Figure 4.1*

# 5. User Interaction

## Access

* The user can access the BinderHub system via any standard web browser.
* The user passes through authentication to access the system.

## Usage

* Upon access, the user can select a customized environment tailored to a specific type of notebook.
* Upon creating the necessary environment, the user can edit, view, and share notebooks.

# 6. Functional Requirements

* The solution should be a web application that is accessible via any modern web browser.
* Users should be able to sign in to the system via their linkblue accounts.
* Users should be able to create environments for their notebooks with respect to the required hardware, operating system, and language environments.
* Users should be able to share their notebooks with others as either as copies of the original or as by editing the original.

# 7. Nonfunctional Requirements

* The system should be heavily compatible with existing Jupyter Notebook related softwares.
* The system should be capable of maintaining long periods of uptime.
* The system should be scalable for future additions and advancements.

# 8. Feasibility

The barebones version of this project is that the system will be able to produce a specialized environment for a single type of notebook. The enhanced version would include the ability to create multiple types of environments.

# 9. Conclusion

Upon conclusion of the project, the system shall be capable of generating many different types of environments that are capable of running Jupyter Notebooks containing various programming languages. The scope of our project is somewhat flexible, and to which degree we are responsible for getting these environments runnable on UK’s systems remains ambiguous to a degree.

# 10. Appendices

Appendix a:

Figure 4.1 System Block diagram, Proposed system model for project