Cloud Lab Vision

|  |  |
| --- | --- |
| Cloud Lab |  |
| Vision | Date: <28/Mar/21> |

# Introduction

The CSU computer science faculty have commissioned this project. Their aim is to build a system with a drag-and-drop interface where you create and configure a lab, with the ability to make multiple copies of the lab to accommodate the number of students who will be using it. A physical lab has a number of drawbacks such as access, hardware, and software limitations. As well as the time cost associated with setting up multiple systems.

# Positioning

## Problem Statement

|  |  |
| --- | --- |
| The problem of | How difficult it is to create a cloud lab for students to use |
| affects | Teachers, Students |
| the impact of which is | Backing up and restoring of student work consumes valuable lab time.  Difficulty upgrading hardware to meet increased demand.  Installing all possible required software on a lab computer can slow the system down.  The difficulty and expense of licensing software on all lab computers. |
| a successful solution would be | Allow students to rebuild the lab environments inside their AWS account and continue lab exercises.  Integrate with existing tools - Excel for student management, and course scheduling with public calendars.  Automation – Educators don’t have to do anything before or after their classes.  Easy Deploy and Manage – Cloudformation for all deployments, and only use managed AWS Services. |

## Product Position Statement

|  |  |
| --- | --- |
| For | *Educators* |
| Who | *Need to create cloud labs for their students* |
| The Cloud lab | *lab creation webapp* |
| That | *is easy and fast to use* |
| Unlike | *cloud formation designer* |
| Our product | *Doesn’t have a steep learning curve* |

# Stakeholder Descriptions

## Stakeholder Summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| CS Faculty | Commissioning the system | Ensures that the system will be maintainable.  Ensures that there will be a market demand for the product’s features.  Monitors the project’s progress.  Approves funding. |
| Teachers | High level user | Participates in testing and requirements analysis. |
| Students | Low level user | Participates in testing and requirements analysis. |

## User Environment

* Number of people involved in a task can range from 1 to 200 and can vary greatly.
* Task cycles will typically last 1 to 2 hours.
* Each activity will typically last 1 to 10 mins.
* There are no unique environmental constraints the system will be accessed exclusively from laptops and desktops.
* What other applications are in use:
* Does your application need to integrate with them:
* Outline the task and roles involved, and so on:
  + Administrator: Authorization and role assignment.
  + Teacher: Template creation, deploying labs.
  + Student: Deploying labs.

# Product Overview

## Needs and Features

|  |  |
| --- | --- |
| Need | Feature |
| Teachers need to deploy duplicate infrastructure to all their students | Creation of cloud lab templates |
| Students need to deploy a replica of their teachers infrastructure | Deployment of cloud lab templates |
| Teachers need a way to evaluate a students progress | Viewing a students changes / progress |
| Not all classes can be completed in time, so they need a way to save that class for later | Backing up a class for later |
| Teachers need a way to avoid duplication of similar work | Syncing templates and sharing them with other teachers |
| Teachers need to make changes on the fly, correct mistakes, introduce new concepts etc... | Dynamically updating the lab with the teachers changes |

# Other Product Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
| Easy to use interface | MAX | ASAP |
| Quick processing and deployment of labs | MAX | ASAP |
| Security so nothing can interrupt the lesson (Think zoom bombing) | HIGH | Always ongoing |
| Low resource usage to keep hosting costs low | MEDIUM | Always ongoing |
| Advanced overview and dashboards | LOW | Last feature if it can be made in time |