Senior Design

Week 1 Report

Interactive Embedded Systems Learning using the Prairie Learn framework

9/13 - 9/20
Faculty Advisor: Phillip Jones

Team Members:

- Ben Stroup
- Caden Last
- Jack Kennedy
- Emmanuel Paz
- Ryan Dela Merced
- Cody Prochaska
- Ryan Bumann

Summary of Progress this Week:

- Download VirtualBox and PrairieLearn
- Read PrairieLean Documentation

Questions:

- Ryan D - Any formal requirements? (our in class lightning talk is on this)
  - Needs vs wants
- Manny - So based on off looking through prairieLearn, I came across different classes that are already made based off a GitHub. Is this essentially what we are creating in our GitLab repo? If so the next steps after creating the questions would be creating automatic graders and parameters or is this in sync with creating the questions?
- Manny - Connecting prairieLearn to Canvas: Use canvas api to create assignments from the prairieLearn server aka the gitLab repo?
- Manny - Is it okay to just use docker instead of VirtualBox?
- Ryan B - Or other VM software besides VirtualBox
- Cody - How will this eventually be hosted? Through an ISU server or something else?
- Jack - how are we deploying the server?
-

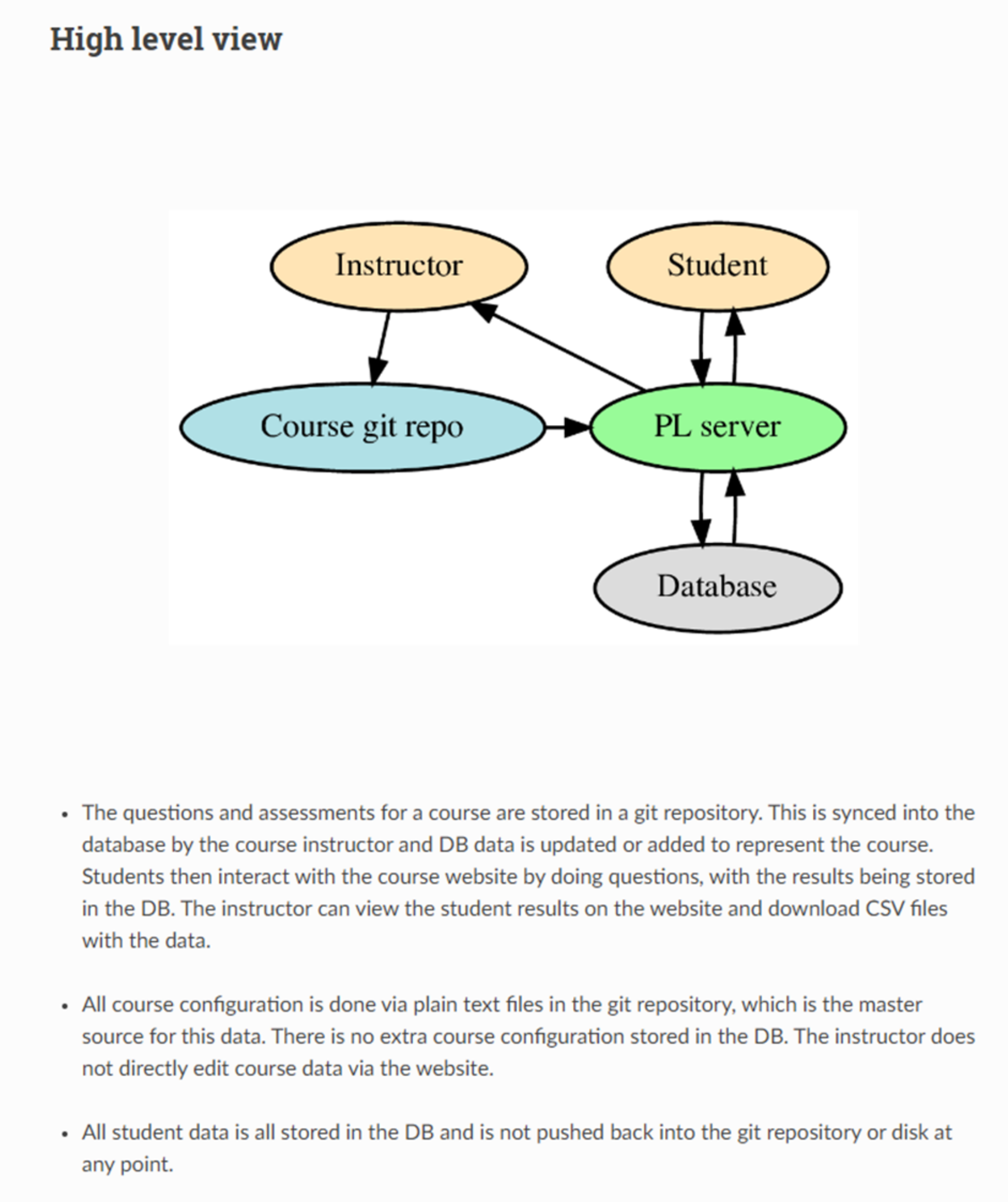
Team Member	Contributions	Hours	Total Hrs
Ben Stroup	Downloaded 20.04.5 Ubuntu server and started it in VirtualBox, Downloaded PrairieLearn and played around with it.		
Caden Last	Download Ubuntu Download Docker Download Prairielearn		
Jack Kennedy	Downloaded 20.04.5 Ubuntu server and started it in VirtualBox, Installed docker on it, Tried to install Prairielearn		
Emmanuel Paz	Downloaded Docker and setup prarieLearn local server.		
Ryan Dela Merced	Quick 288 refresher, Contacted 288 student, Downloaded Docker & Prairielearn, ran the example.		
Cody Prochaska	Set up Ubuntu 20.04 VM, installed docker and got a local version of PrairieLearn running on it.		

Ryan Bumann	Setup Ubuntu 20.04 virtual machine and downloaded docker and ran an example from prairielearn. Began looking through prairielearn docs.	2	2

Additional Information:

- Ryan B

High level overview of prairie view structure



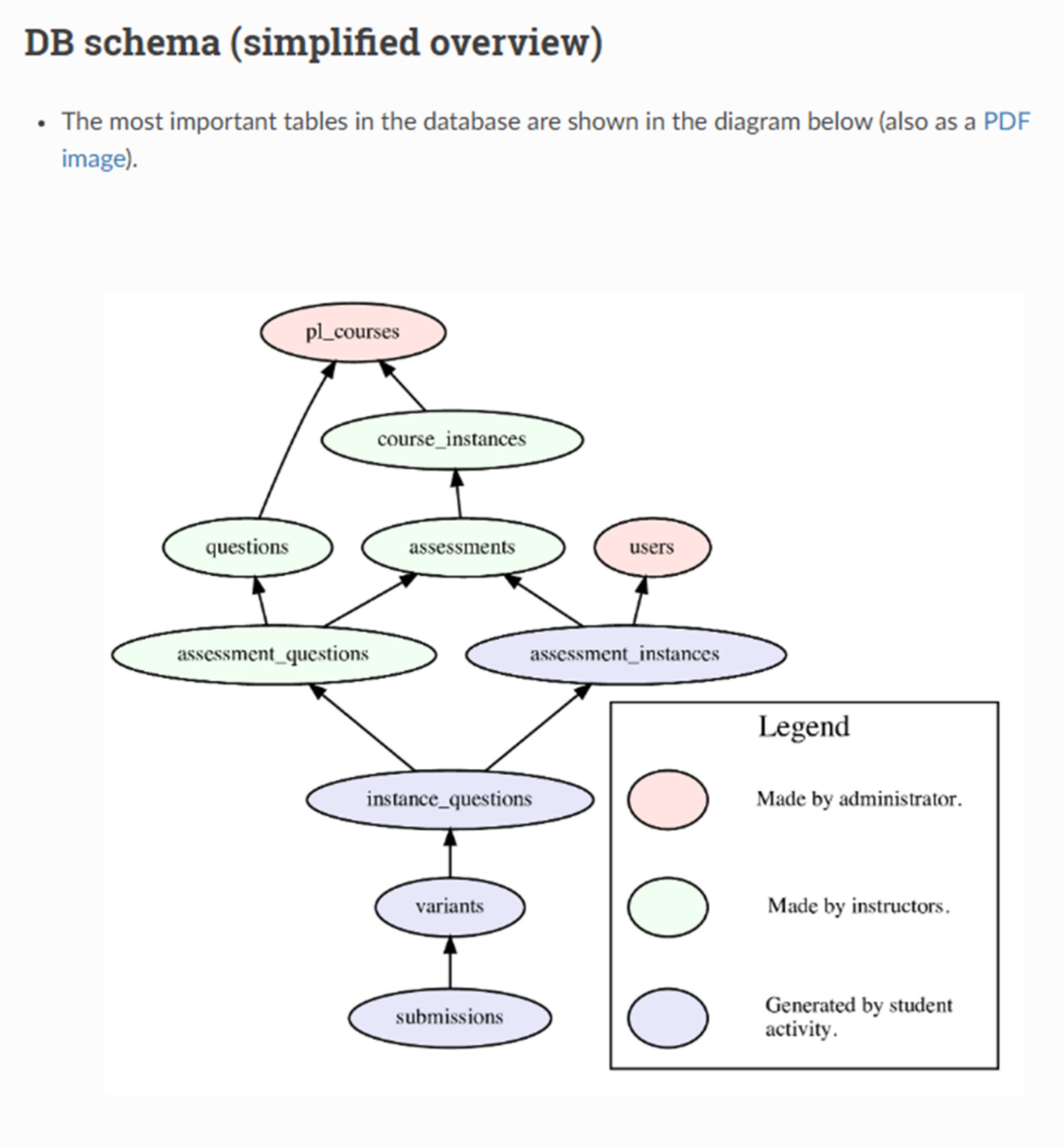
High level view

- The questions and assessments for a course are stored in a git repository. This is synced into the database by the course instructor and DB data is updated or added to represent the course. Students then interact with the course website by doing questions, with the results being stored in the DB. The instructor can view the student results on the website and download CSV files with the data.

- All course configuration is done via plain text files in the git repository, which is the master source for this data. There is no extra course configuration stored in the DB. The instructor does not directly edit course data via the website.

- All student data is all stored in the DB and is not pushed back into the git repository or disk at any point.

Important database structures



DB schema (simplified overview)

- The most important tables in the database are shown in the diagram below (also as a PDF image).

Docs contain multiple pages on autograding C/C++ code. How to setup, and available test options, etc...

Docs contain drawing capabilities. (possible problem to program “robot” to do something). Make sure all students know their fair share of programming.