**D3L**

**System Design Document**

Nick Bashor

Patrick Grzegorczyk

Wesam Jabali

Joey Navarro

Kevin Sass

Luigi Yebra

##### **Original Plan Date: February Fifth, Twenty Twenty-one**

##### **Revision Date:January Fifth, Twenty Twenty-one**

##### **Revision: 1.0**

# **Revision History**

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Comment |
| 1.0 | February 5, 2021 | Original DoIT PMO Document |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

**Scope .......................................................................................................................................... 5**

Audience .......................................................................................................................... 5

Related Documentation ................................................................................................... 5

**System Overview ....................................................................................................................... 6**

Description ....................................................................................................................... 6

System Architecture ......................................................................................................... 6

*Software Architecture* ........................................................................................... 6

*Hardware Architecture (Not Applicable)* ............................................................... 6

**Hardware Design (Not Applicable) ........................................................................................... 7**

Hardware Components (Not Applicable) ......................................................................... 7

*Computer Systems* *(Not Applicable)* .................................................................... 7

*Peripherals* *(Not Applicable)* ................................................................................ 7

*Networks (Not Applicable)* .................................................................................... 7

*Project-Specific Hardware Items* *(Not Applicable)* ............................................... 7

Hardware Integration (Not Applicable) ............................................................................. 7

*Logical Design* *(Not Applicable)* ........................................................................... 7

*Physical Design (Not Applicable)* ......................................................................... 7

*Recovery Design* *(Not Applicable)* ....................................................................... 7

**Software Design ........................................................................................................................ 8**

Software Packages .......................................................................................................... 8

*Software Module* .................................................................................................. 8

Software Integration ......................................................................................................... 8

**Data / Database / Files ............................................................................................................. 10**

Data Flow Diagrams ...................................................................................................... 10

Database Design ........................................................................................................... 12

Files ............................................................................................................................... 13

## 

## **Scope**

**Hardware**

* *Not applicable*

**Software**

* Vue.JS (Vue 2)
* Express.JS
* PostgreSQL
* Knex.JS

## **Audience**

Users/readers must have a fair understanding of code, having at least a bit of experience with or some knowledge involving Vue.JS, Express.JS, PostgreSQL, & Knex.JS.

## **Related Documentation**

Right now we have three other documents that we have already filled out. Provided below are clickable links to them all:

* [D3L Test Plan Document](https://docs.google.com/document/d/1DjujxHx_p4k7PqsVNRhvyu5lxmam0h3VJafZE433vzc/edit?usp=sharing)
* [D3L Requirements Analysis Document](https://docs.google.com/document/d/1_p2Gz5g3ApQst7yBIkK0BB2km-aVxh1T6IBJ5yPVibA/edit?usp=sharing)
* [D3L Use Case Document](https://docs.google.com/document/d/1sgKP_qqYea4amz4RR7q_yItP3y46MbhZgHP2F9nIeC4/edit?usp=sharing)
* [Templates for Documents](http://www.doit.state.nm.us/project_templates.html)

**System Overview**

## **Description**

The purpose of this system is to manage the relationship between students, faculty, and administration in a university. It has the ability to:

* Manage users
  + Create user IDs and faculty IDs
* Manage courses
  + Courses are organized in alignment to DePaul’s course number format.
  + Students are assigned to courses
  + Teams are assigned to courses
  + Content is assigned to courses
  + Manage discussion forums per course
* Manage content
  + Content can be any file upload or text submission by faculty
  + Content can be graded or ungraded
  + Manage discussion forums per piece of content
* Manage teams
  + Students can be assigned to teams
  + Teams assigned to courses
  + Teams can have group submissions, as used in D2L.

## **System Architecture**

### **Software Architecture**

This section outlines the software architecture established for the project. Provide references to the System Architecture Document and a brief summary of the software architectures.

### **Hardware Architectures**

*Not applicable*

# 

# 

# 

# 

# **Hardware Design**

## **Hardware Components**

### **Computer Systems**

* Netlify
* Heroku

### **Peripherals**

* *Not applicable*

### **Networks**

* *Not applicable*

### **Project-Specific Hardware Items**

* *Not applicable*

## **Hardware Integration**

### **Logical Design**

* *Not applicable*

### **Physical Design**

* *Not applicable*

### **Recovery Design**

* *Not applicable*

# **Software Design**

## **Software Packages**

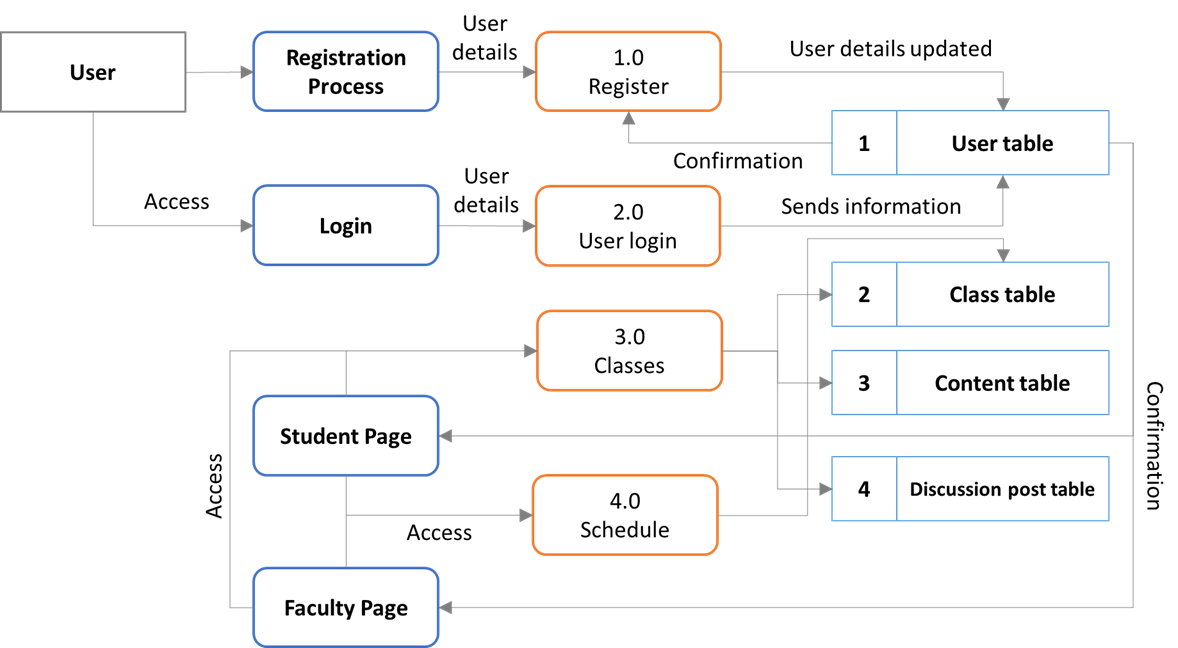
* **Frontend**
  + components/
    - HeaderComponent.vue
    - Snackbar.vue
  + plugins/
    - axios.js
    - vuetify.js
  + router/
    - index.js
  + store/
    - index.js
  + views/
    - Home.vue
    - Login.vue
    - Register.vue
  + main.js
* **Backend**
  + database/
    - migrations/
      * starting\_schema.js
    - seeds/
      * 01\_d3l\_user.js
    - sqlScripts/
      * starting\_schema\_down.pgsql
      * starting\_schema\_up.pgsql
    - knex.js
  + middleware/
    - authMiddleware.js
    - Cors.js
    - rolesMiddleware.js
  + routes/
    - app/
      * account/
        + index.js
        + user.js
      * auth.js
      * index.js
    - index.js
  + app.js
  + knexfile.js

## **Software Integration**

* **Vue.JS** - Frontend framework
* **Node.JS** - Backend framework
  + **Express.JS** - Web application framework
* **Knex.JS** - Querying middleware

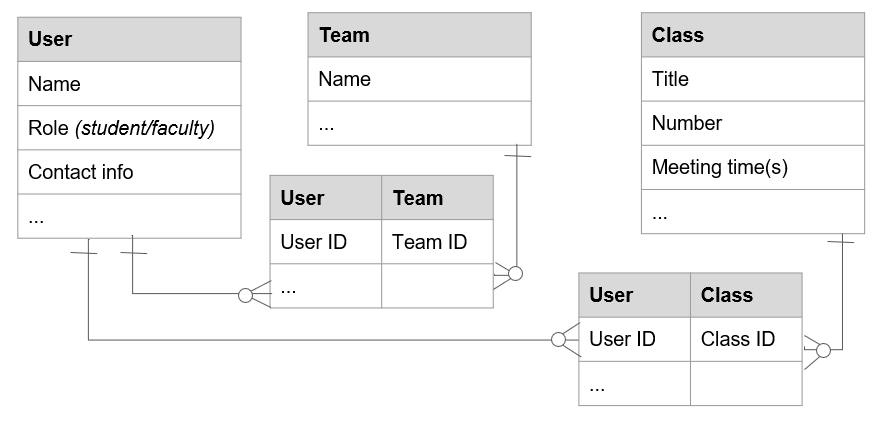
**Data / Database / Files**

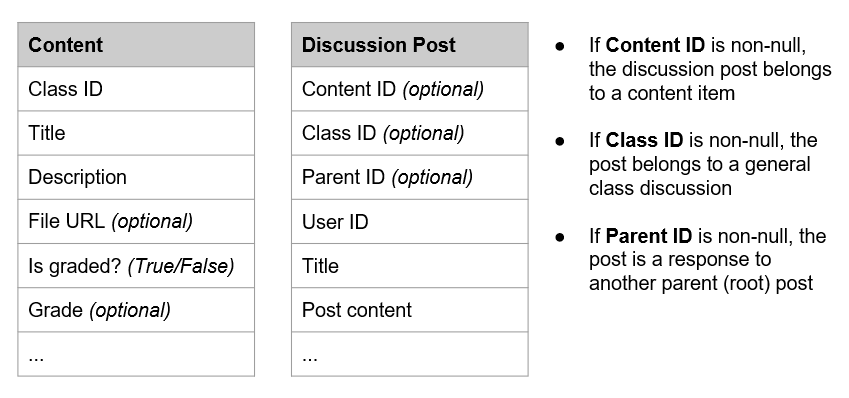
## **Data Flow Diagrams**



## 

## **Data Flow Diagrams (cont’d)**





## 

## **Database Design**

List and describe tables, fields, and entity relationships (also known as data dictionary and logical/physical database design), schema, query language, key and indices, data management functions.

* **Query language:** PostgreSQL
* **Data management**
  + Querying via Knex.JS
* **Tables**
  + User
    - Timestamp
    - ID
    - First name
    - Last name
    - Email address
    - Password (encrypted)
    - Phone number
    - Home address
  + Course
    - Timestamp
    - ID
    - Title
    - Course prefix
    - Course number
    - Section number
  + Team
    - Timestamp
    - ID
    - Team name
    - Course ID (foreign key to Course)
  + Content
    - Timestamp
    - ID
    - Course ID (foreign key to Course)
    - Title
    - Body
    - File URL
    - Graded
    - Points earned
    - Total points
  + Discussion Post
    - Timestamp
    - ID
    - User ID (foreign key to User)
    - Parent ID (foreign key to Discussion Post)
    - Content ID (foreign key to Content)
    - Title
    - Body
  + User-to-Team
    - Timestamp
    - ID
    - User ID (foreign key to User)
    - Team ID (foreign key to Team)
  + User-to-Course
    - Timestamp
    - ID
    - User ID (foreign key to User)
    - Course ID (foreign key to Course)
  + User-to-Role
    - Timestamp
    - ID
    - User ID (foreign key to User)
    - Role

## **Files**

* root/
  + database/
    - migrations/
      * starting\_schema.js
        + Identifies startup PostgreSQL script
        + Identifies shutdown PostgreSQL script
    - seeds/
      * 01\_d3l\_user.js
        + Defines admin user
    - sqlScripts/
      * starting\_schema\_down.pgsql
        + PostgreSQL script to execute when shutting down
      * starting\_schema\_up.pgsql
        + PostgreSQL script to execute when starting up
    - knex.js/
      * Querying framework