NULabs

Project Proposal

Emily Chen and Nathan Gong

CS 3200 Database Design

Northeastern University

Table of Contents

1. Readme

Instructions for Running the Project

1. Install the following software to your directory of choice.

Software/Libraries	**URL**
Python 3.7.x +	https://www.python.org/downloads/
MySQL Workbench	https://dev.mysql.com/downloads/workbench/
Python IDE or Text/Code Editor	https://code.visualstudio.com/download https://www.jetbrains.com/pycharm/download/#section=windows
Web Browser	https://www.google.com/chrome/?brand=YTUH&geo=US&gclid=EAIaIQobChMI_d-Exuri7wIVdeW1Ch3r0gKnEAAYASAAEgIlb_D_BwE&gclsrc=aw.ds https://support.apple.com/downloads/safari
PyMySQL	pip install
Flask	pip install

2. Connect to a MySQL server (through the MySQL Workbench).
3. Download and extract all files from the nulabs.zip file.
4. Import the nulabs/sql_scripts/nulabs.sql SQL dump file to a MySQL database.

5. Open nulabs/app/functions.py within your Python IDE or text/code editor of choice. Under the "MySQL Server User Login Information" section, modify the **db_username** and **db_password** variables to match your personal MySQL username and password.
    a. Ex: db_username = "root", db_password = "password123"
6. Within your Python IDE or via the command line, execute the nulabs/app/main.py script and navigate to the provided local host server address within your web browser of choice.

Notes for the User

- If you are logged in and refresh the webpage or stop the execution of app/main.py, you will need to log back in.
- If you make any changes to the codebase while the application is running, you will need to refresh the webpage and/or rerun nulabs/app/main.py in order for those changes to come into effect.

2. Technical Specifications

All technologies used during the course of this project are listed in the categories below. Only the **bolded items** are necessary to be installed in order to replicate this project.

Software

- **MySQL Workbench**
    - Used to import the SQL dump file to create the schema
- **Python IDE or Text/Code Editor**
    - Used to edit the *functions.py* file to match the user's MySQL login username and password
    - Used to execute the *main.py* file to run the application (can also be run from the command line)
- **Web Browser**
    - Used to view the locally-deployed web server
- Jupyter Notebook

- Microsoft Excel
- Git

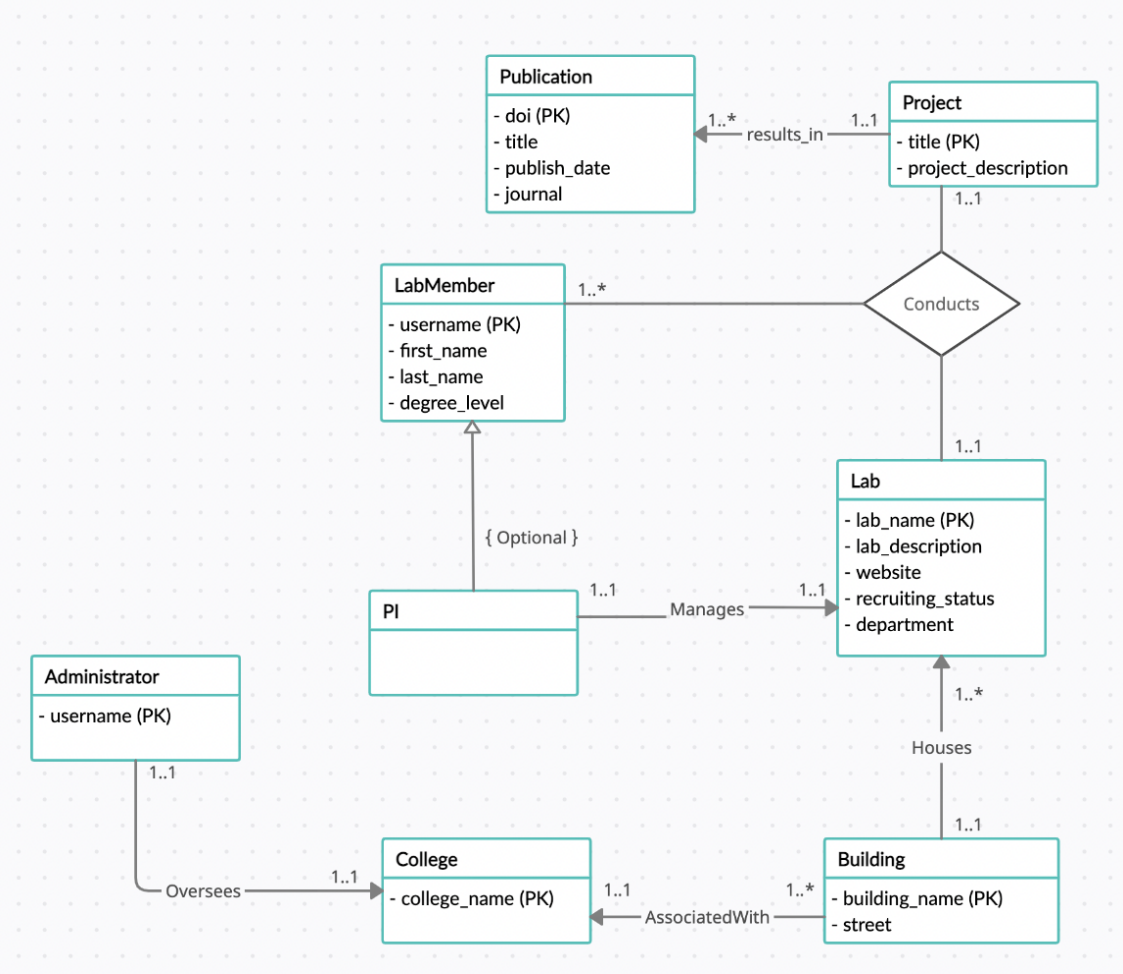
Languages

- **Python (3.7.x +)**
- HTML
- CSS
- SQL

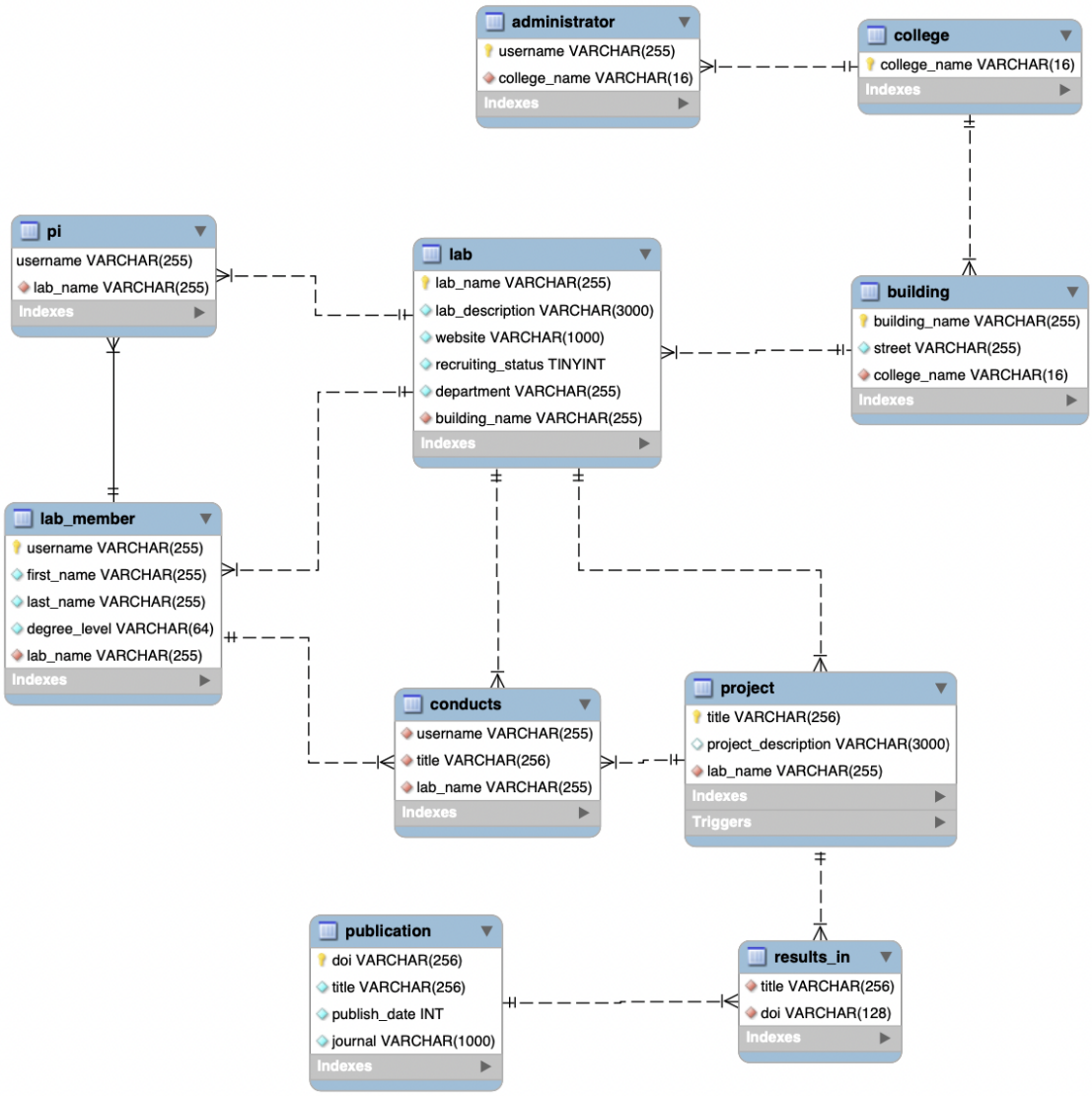
Libraries

- **PyMySQL**
- **Flask**
- Pandas
- Bootstrap

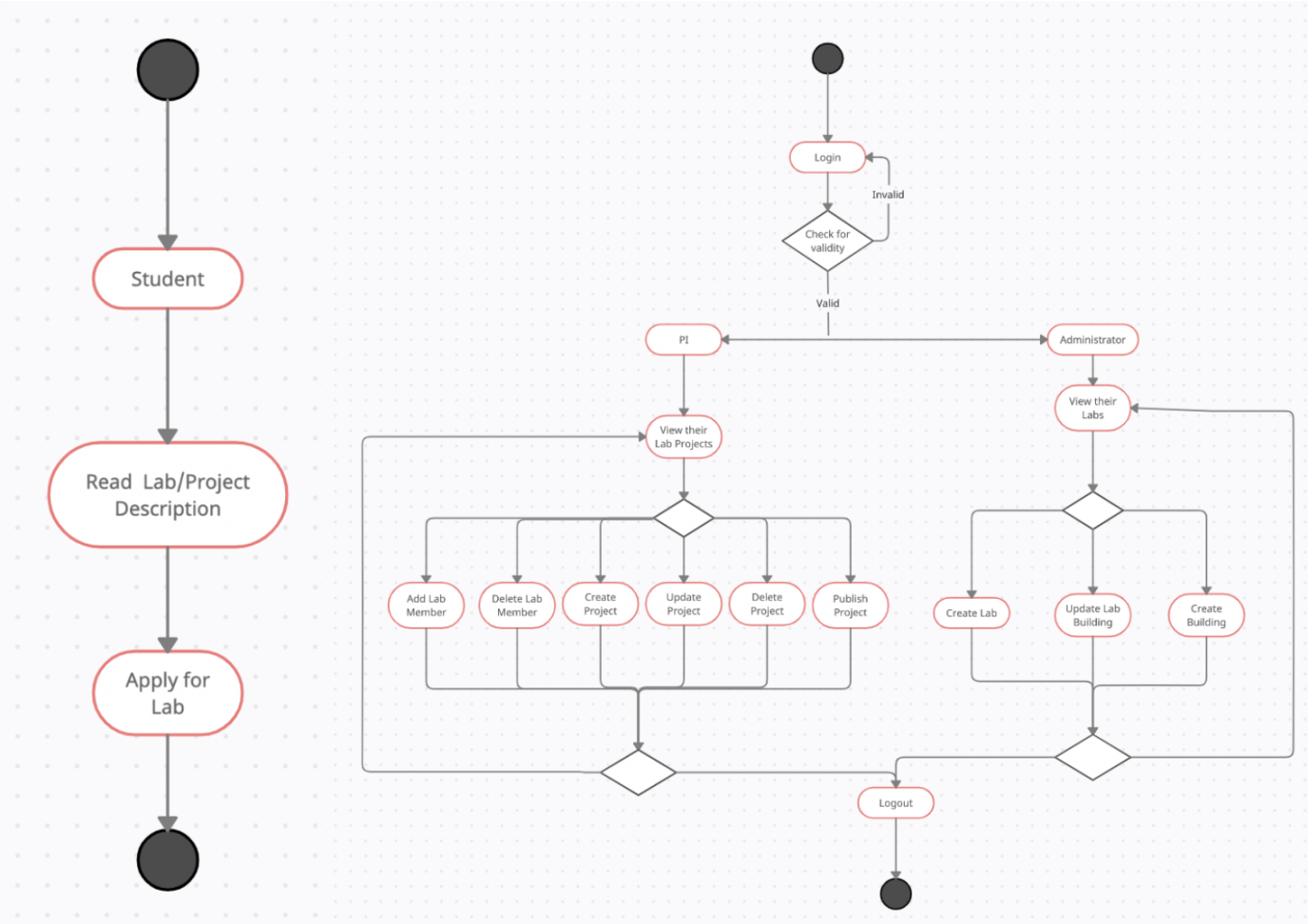
3. Conceptual Design (UML)



4. Logical Design (Reverse Engineered)



5. User Interaction Flowchart



1. A Student can read the lab projects stored in the database.
    a. A Student may choose to apply to a lab that is conducting a project that they find interesting.
2. A user may login to the application using their Northeastern username.
    a. If the username is valid, they may proceed as either a PI or an Admin.
    b. If the username is invalid, they will be asked to try to login again.
3. A PI may view their lab projects associated with their lab.
    a. A PI may choose to add a lab member for their lab.
    b. A PI may choose to delete a lab member for their lab.
    c. A PI may choose to create a new lab project for their lab.
    d. A PI may choose to update an existing project in their lab.

    e. A PI may choose to delete a lab project from their lab.
    f. A PI may choose to publish a project in their lab.
4. An Administrator may choose from three available options.
    a. An Administrator may create a new lab.
    b. An Administrator may update an existing building.
    c. An Administrator may create a new building.
5. A User may log out of the application (navigate back to the home page).

6. Lessons Learned

Technical Expertise Gained

This project has enabled us to expand our knowledge in data collection, SQL and Python. We collected real world data from Northeastern research websites and the lab websites listed. We were not able to include every single lab from each college's website. We were also not able to include all the publications from each lab because we were collecting the data manually. Understanding our limitations and being able to weave through the data to only include the most meaningful points is very valuable experience. As for SQL, we were able to develop a schema and accompanying programming objects. We designed each programming object with the overall functionality of our website in mind. Lastly, upon connecting our SQL database to Python, in order to make a user interface, we decided to make a website using Flask, a library neither of us have used before, and HTML and CSS, which neither of us have in depth experience with.

Insights

We had good time management throughout the project. We met 2-3 times a week to work on the project, discussing our progress and action items to complete before the next meeting. As for data insights, we could have used web scraping to gather some of our data. While we may have not been able to use web scraping for every website because some websites have complex structures, it still would have made the data collection process shorter.

Alternative Design Choices

At first, we were planning on designing a website using Django, a Python web framework. Upon reading the documentation, we realized that Django is usually used for more complex websites. In addition, the documentation itself was a little hard to understand so we went with what we thought was a simpler solution: a GUI. We used Python Tkinter to start creating a GUI, but we decided that in order to achieve the functionality we wanted to achieve with our application, a website is the best choice. We reached a final decision of using a website with Flask, a micro web framework.

Code Not Working

When we tried to import the data into our SQL database using import wizard, most of the data did not import properly. However, when we tried insert statements on a couple inserts, it worked. To accommodate for the un-inserted data, we made a Python script using pandas to generate insert statements for each table in the database.

7. Future Work

Planned Uses of the Database

This application could be used to bolster Northeastern's reputation as a research institution by making it easier for all users to access information related to the research labs at Northeastern in one centralized location. Current and prospective students alike are able to access information pertaining to labs across all colleges at Northeastern without needing to login. Additionally, PI's and Administrators are able to advertise their labs to a greater audience of interested students, which could serve to both match students with research projects that best fit their interests and bolster overall interest and participation in undergraduate and graduate level research. Having an application that displays relevant lab information in a neat and standardized fashion means that students no longer have to go digging around to find a PI's projects, contact information, or if a lab is even recruiting or not. PI's also don't need to worry about creating their own websites to

display lab information, which can lead to confusion about what information to upload and how to organize it, as well as neglect for maintaining the website due to the cost of time. If there is demonstrated interest, this application could grow to encompass research centers in addition to individual research labs, as well as expand to other research-centered universities.

Potential Areas for Added Functionality

If Northeastern or Khoury college were interested in collaborating to take this application further, a number of potential additions could be made:

- Use official Northeastern username and password information and 2-factor authentication to increase security when validating user login.
- Expand the database to include any missing research labs, centers, or colleges at Northeastern, such as the College of Arts, Media, and Design, or the College of Professional Studies.

Several improvements could also be made within the application itself:

- A method of filtering the labs within the Student view (by college, department, recruiting status, etc.) rather than displaying all labs in a single page.
- Purchase a domain name and host the application on the web so that users don't need to download a bunch of files and software in order to access the application.
- Continuously stream data from a remote database server rather than the local server so that users don't need to import a SQL dump file to use the application.
- Improve the overall aesthetic of the website and its pages through front-end development.
- Include a help or contact section for users to reference in case they have questions or encounter errors when interacting with the application.
- Improve the modularity of the program code so that a separate page does not need to be created for each new possible user operation and that code changes only need to be applied to a single file rather than multiple pages.
- Improve error handling so that when exceptions are thrown, they are not displayed on the website, but rather a user-friendly error message and clear instructions are printed.