How Data Model Changes Impact Project Development of Web Application

Snehitha Mamidi
Snehitha.Mamidi@unh.edu
Master Project in Information Technology
Department of Applied Engineering and Sciences
University of New Hampshire
Manchester, New Hampshire, USA

ABSTRACT

The Academic Advising department at UNH Manchester uses Excel spreadsheets to manage evaluations of courses students transfer from other institutions towards their UNH programs of study. The Excel workbook contains transfer evaluation records organized in Excel worksheets that correspond to the student's major. To avoid Excel processing and improve operations with managing transfer evaluations, two former graduate students in the Information Technology program built a web application. The limitations of their web application motivated my study. The Academic Advising department has changed the Excel workbook since the previous version of the project. The main goal of my project is to change the database model of the web application, update the transfer evaluation methods, and improve the essential features of the web application. In the new Excel workbook, two new columns are added that are state name, UNHM course title and a column named sem/year taken is removed. I updated the data model diagram by adding state-name in the School model and major-req-course-title in the UNHM course model. Based on the updated data model, I changed backend transfer methods and front end designs for all the form components. I improved essential features such as implementing sorting functionality and dependent drop-down menus. I informed the director of Academic Advising of the status of the project. I evaluated the project by testing the backend implementation using Postman and getting the feedback from the end-user.

CCS CONCEPTS

• **Information systems → Web applications**; • **Software and its engineering → Application specific development environments**.

KEYWORDS

Django backend, ReactJS frontend

1 INTRODUCTION

All the transfer course evaluations at UNH Manchester are maintained using Excel spreadsheets. Maintaining transfer evaluations manually is very difficult and leads to many mistakes. This will also result in transcription errors and data duplication. To avoid all these errors, having a web application to maintain all the transfer course evaluations is very important. The main unique features of this project are creating, viewing, updating, and deleting transfer evaluations. There are many limitations in the previous version of this web application which motivated my project study. The main limitations in the previous version are the create form to create new transfer evaluation is not working.

This project uses Django REST API for the backend and ReactJS for the frontend [4]. The Django app for the back-end will function as application programming interface (API) as shown in figure 1. This web application was already built by two graduate computing students. I took over this project for my internship and continuing it for my master's project. I should upgrade the older version of the web application by improving the required features like implementing dependent dropdowns and sorting functionality which will sort the tables in ascending and descending order. Then deploy the application to Heroku and present it to the UNHM Advising department. Once the web application is reviewed the main aim of this project is to get the new data source that is a spreadsheet from the Academic Advising office which contains a lot of tabs. This data source has a different organization than the original source which is used in the previous version of this web application. Based on the data source, I proposed a strategy to update the existing data model and according to the changes in the data model, I updated the transfer methods.

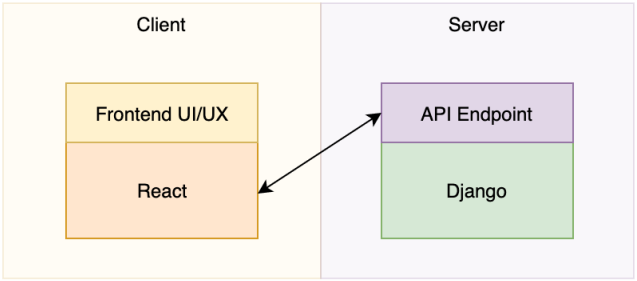

Figure 1: Integrating ReactJS with Django REST API

2 OBJECTIVES

The goal of this project is to improve essential features of the Transfer Evaluation Service web application which were not working in the previous version of this project and making the web application user-friendly. The main aim of the project is to study the impact of changing the data model of an existing Transfer Evaluation web service on the

- Development process which can be essentially broken down to three phases: design, implementation and testing. The first stage is design which is basically building a new database

model diagram of the project. The second stage is implementation which is very crucial where all the transfer methods will be changed based on the updated database model diagram and the final stage is testing where need to run the project and check for the errors. If there are any errors, the next step would be to debug the project and rectify those errors
- Web service architecture and its components
- Evaluation process: effective communication with the client and end-user that is UNH Manchester Academic Advising Department who manages the transfer evaluations

3 APPROACH

The dataset used in this project is an Excel spreadsheet that contains lots of tabs compared to the previous version. Once the data source is imported, the data will be stored in the particular data models based on the design of the data model and using the backend python script. The data model will be changed because this data source is completely a new organization that contains huge data. Below are the steps that are carried out in this project and are explained in detail further:

- Implement search functionality in Transfer Course page based on school name
- Implement search functionality in Major Requirement page based on major name
- Implementing sorting functionality in both ascending and descending order for all the columns in all the pages
- Implementing three searchable dependent drop-downs one for major, school and course number. Once the major is selected, only schools of that particular major will be displayed in the drop-down. After selecting major and school, only transfer courses of that combination will be displayed in courses drop-down
- Contact director of Academic Advising Department and get the new Excel spreadsheet which contains the latest version of the data source
- Based on the new data, update the database model diagram or design a newer version of the data model
- Propose a new strategy of all the changes that to be made based on the data model diagram
- Make the changes to transfer methods based on new Excel spreadsheet
- Design a logic for create form to create new Transfer Evaluation
- Change the front end design for create form component to create a new Transfer Evaluation
- Change the front end design for update form to update the existing Transfer Evaluation
- Test the backend implementation using Postman
- Inform the client about the current status of the project

3.1 Sorting Functionality in Ascending and Descending Order

I implemented sorting functionality based on ascending and descending order for every column in all the pages which makes the user experience easy. I wrote two functions one is to sort in ascending order and the other is to sort in descending order [1]. I used two buttons for each column named asc and desc. When a user clicks on asc button it will call the function that is compareaescMajor which takes two values from the list of Majors as arguments named as a and b and compares them alphabetically and sorts based on the function. If a's name is less than b's name then it returns a negative value which means a's name should be before b's name. If a's name is more than b's name then it returns a positive value which means a's name should be after b's name. If both have the same name then it returns zero value which means they both have the same order. When a user clicks on the desc button it will call the function that is comparedescMajor which takes two values from the list of Majors as arguments named as a and b and compares them alphabetically and sorts based on the function. If a's name is less than b's name then it returns a positive value which means a's name should be after b's name. If a's name is more than b's name then it returns a negative value which means a's name should be before b's name. If both have the same name then it returns zero value which means they both have the same order. If the user sorts one column, automatically all the column data will be sorted based on the selected column.

Listing 1 shows the code for sorting functionality based on major name:

Listing 1: Sorting by major

```
function compareaescMajor( a, b ) {
    if ( a.major_name < b.major_name ){
      return -1;
    }
    if ( a.major_name > b.major_name ){
      return 1;
    }
    return 0;
  }

function comparedescMajor( a, b ) {
    if ( a.major_name < b.major_name ){
      return 1;
    }
    if ( a.major_name > b.major_name ){
      return -1;
    }
    return 0;
  }
```

3.2 Updated Database Model Diagram

The data file used in this project is an Excel spreadsheet that contains data of transfer evaluations for different majors. I updated the database model diagram because the Excel sheet which was used earlier is different than what I used in this project. This data model contains 6 models which are School, TransferCourse, Approver, Major, MajorRequirement, TransferEvaluation. Compared to the previous spreadsheet two important columns are added and one column which is not necessary is removed. state-name is added in the School model. The combination of school-name and state-name should be unique. major-req-course-title is added in the MajorRequirement model. The combination of major-req-course-no and major-req-course-title should be unique. One attribute which is

named sem-year-taken is removed from the TransferEvaluation model. Figure 2 shows the updated database model diagram.

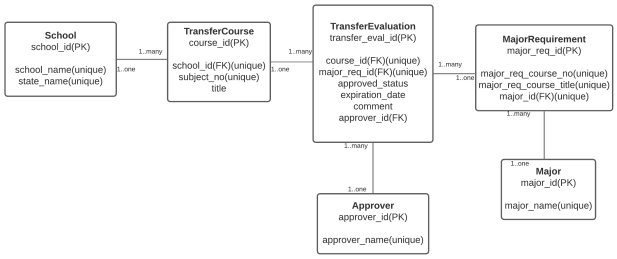

Figure 2: Updated database model diagram

3.3 Updated School Page

In the School model based on the latest version of the spreadsheet, I added a state name which is a character field. The combination of state-name and school-name should be unique. For the backend part, while importing the spreadsheet I implemented two functions in load.py to import school name and state name based on the school id. While importing schools, it also imports the state name of that particular school name from column 1 in the spreadsheet. When we click on the state name or school name, we can see the details of that particular school that are school name, state name, and school id as shown in figure 3.

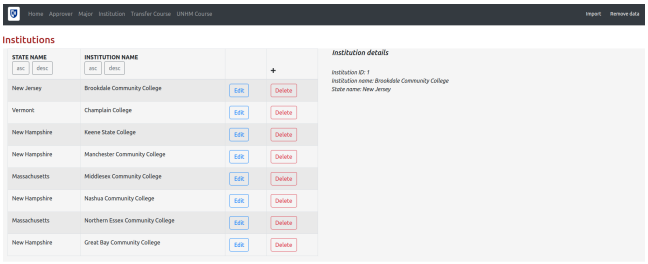

Figure 3: Detail page of the selected school

For editing a school or creating a school, the same form component is used which is imported from react-bootstrap. The difference between creating and editing forms is that in the create form all the fields are empty but in the edit form, all the fields will be filled based on selected school or state.

3.4 Updated UNHM Course Page

From the latest version of the spreadsheet there is a new column added which is named as UNHM Coure title. So, in the MajorRequirement model, I added major-req-course-title which is a character field. The combination of UNHM course title and UNHM Course number should be unique. For the back-end part, in the previous version of the project only major-id and major-req-course-no combination is imported from spreadsheet. I made changes to back-end so that now it takes major-id, major-req-course-no and major-req-course-title.

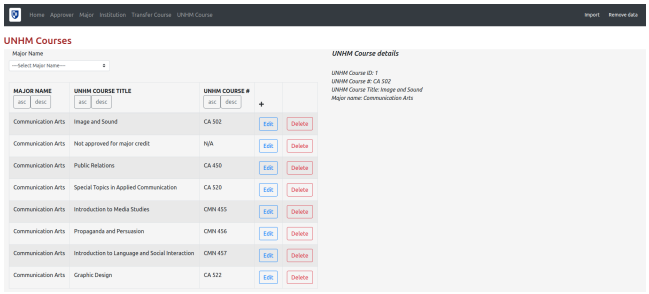

Figure 4: Detail page of the selected UNHM course

For the front-end part, In the form component I have added UNHM Course title field for editing a selected course or for creating a new UNHM Course. When the user clicks on any major name or UNHM course title, it displays details of that selected UNHM Course. The details it displays are major name, UNHM Course title and UNHM Course number as shown in figure 4. When the user clicks on edit, automatically the details of that selected course will be populated automatically and can be updated.

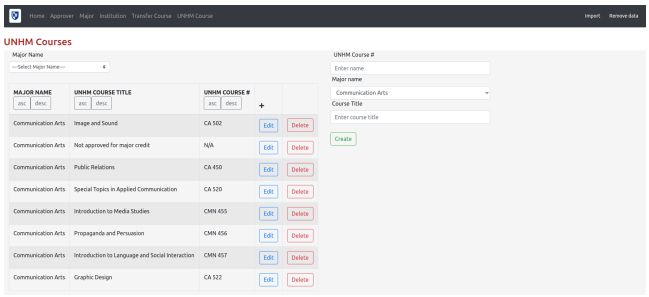

Figure 5: Create form of the selected UNHM course

3.5 Upgraded Transfer Evaluations Page

Based on the latest version of the Excel spreadsheet there is no column for the sem/year taken. From the TransferEvaluations model, I removed the sem/year taken field. In the front end, the Home page which is the Transfer Evaluations page I removed the sem/year taken column from the table. One more column is added in the Transfer Evaluations page that is the UNHM course title as displayed in figure 6.

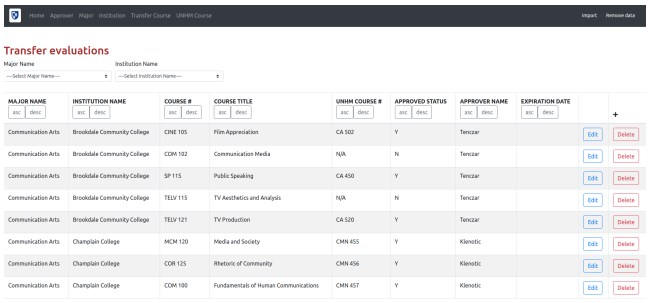

Figure 6: Transfer Evaluations page

3.6 Project Workflow: Project Board, Pull Requests, Issues

The Project board that I used on GitHub is Automated Kanban where the cards will be moved automatically between different columns like To do, In progress and Done. Project board is used to create notes. After notes or cards are created we should convert it to issue to automate the project. We can create works that are to be done in To do column. Once the work is started we should move the particular note or card to In progress column. Once we work on our local machine, we should make sure that we are in developer branch not in master branch. After the code is done, we should add, commit and push the work to developer branch. Below are the steps and commands:

(1) Checkout a new branch **git checkout -b <branch-name>**
(2) Add all the files that are changed **git add -A**
(3) Commit all the files that are added **git commit -m 'commit-message'**
(4) Push to the remote branch **git push origin <branch-name>**

Now, we need to compare and generate a pull request in GitHub. We can compare the changes from master branch to developer branch. Once all the code is reviewed and compared we should link pull request to issue so that the issue that is created in kanban board will be moved to Done column automatically after the pull request is merged to master branch. In order to link pull request to isssue, there are few keywords that are close, closes, closed, fix, fixed, fixes, resolve, resolves, resolved. The structure that is to be used in the comment box is keyword #issue-number [2]. Now we can merge the pull request to master branch. Once this is done, the linked issue will be automatically moved to Done column as shown in figure 7.

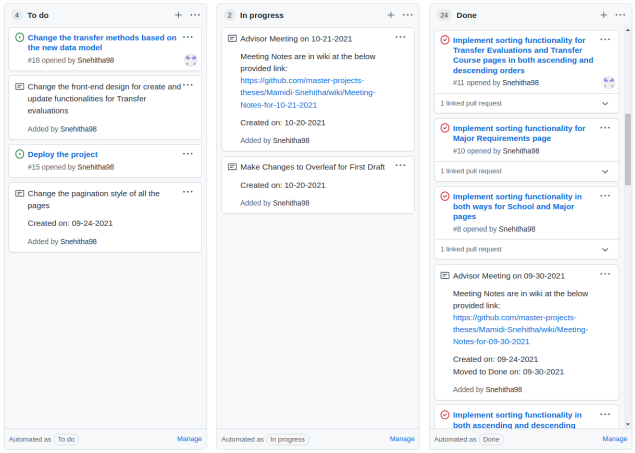

Figure 7: Automated Kanban Project board

4 RESULTS

The end result of this project will be successfully functioning a web application to maintain transfer evaluations. Many important features like search functionality, dependent drop-downs and sorting functionality will be added to this application. A new data model will be designed based on the new data source. According to the data model, the transfer methods will be upgraded to a newer version. The client or the end-user will be informed regarding the current status of the project.

4.1 Searchable and Cascaded Drop-downs

In this project, I added search functionality for Transfer Evaluations, Transfer course, and Major Requirements page. In the Transfer Evaluations page, I implemented searchable cascaded drop-downs for major, school and course number as shown in figure 8. Once the major is selected, only schools of that particular major will be displayed in the school drop-down. After selecting major and school, Transfer evaluations of this combination will be displayed on the Home page. In the Transfer Course page and Major Requirement page, I implemented a searchable drop-down. In the Transfer Course page, Once the user selects a specific school, transfer course number and transfer course title of that school will be displayed. In the Major Requirement page, Once the user selects a certain major, UNHM equivalent of that major will be displayed.

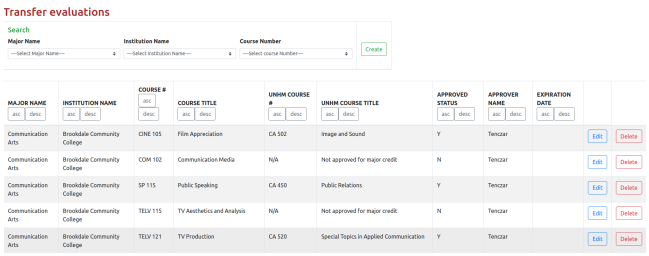

Figure 8: Cascaded drop-downs

4.2 Sorting All Columns by Ascending and Descending Order

In the previous version of the web application, there is no option for sorting all the columns based on ascending and descending order. When the user uses the web service, In the Transfer Evaluations page there are 9 columns which will be very difficult for the user to search for a particular transfer evaluation. In this project, I added sorting functionality for all the columns in the table to sort in both ascending and descending order which makes the user job easier. I added two buttons for each column headers named asc and desc as shown in figure 9. If we sort any one column, respective columns will also be sorted based on the criteria.

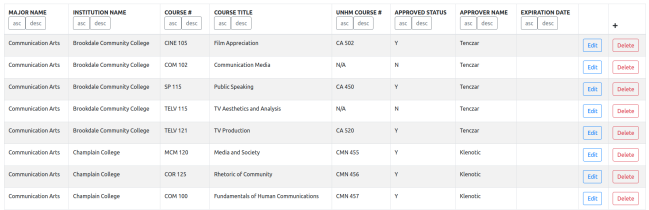

Figure 9: Sorting all columns

4.3 Create Form of New Transfer Evaluation

One of the main important step is to implement create form of new transfer evaluation. I designed the logic for creating a new transfer evaluation. The user can create a new transfer evaluation under few conditions that are major is unique or school name should be unique or the combination of transfer course and UNHM course should be unique. If any of those conditions doesn't meet then the user cannot create a new transfer evaluation and error message that is Error while creating new transfer evaluation will be displayed in warning color that is red color.

There are three drop-downs in create form, one is for major, other is for school and last is for approver name. All the majors, schools and approver names that are imported from spreadsheet will be displayed in the drop-downs. If the user needs to create a new school then the user can go to School page to create. The user can create a new transfer course by entering transfer course number and transfer course title. There is one more drop-down for UNHM course number which will display all the UNHM course numbers. The user can select the existing UNHM course number or the user can create a new UNHM course by entering UNHM course number and UNHM course title. In the Approved status drop-down, there are two options that are yes and no. The user can select any of the option and also select the expiration date as shown in figure 10. When the user clicks on create, a new transfer evaluation will be created successfully and redirects to the Home page or if there is any error it will display an error message. When the user clicks on cancel then the operation to create new transfer evaluation will be cancelled and redirects to transfer evaluations page.

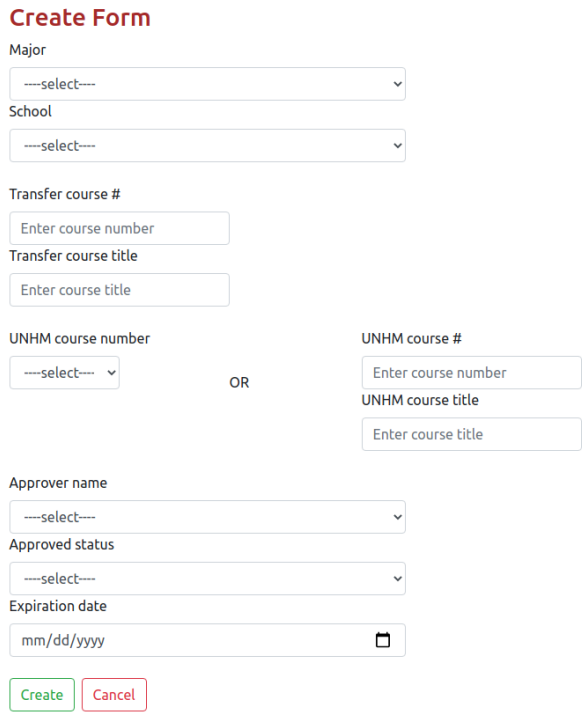

Figure 10: Create form

4.4 Update Form of Existing Transfer Evaluation

I updated the front end design for edit form of transfer evaluation. In the home page, when the user clicks on edit it redirects to update form. The user can update Approver name, Approved status and expiration date. In order to update Transfer course number or Transfer course title, there is a link added named as Transfer Course which redirects to Transfer Course page. To update UNHM Course details, there is a link added named as UNHM Course as shown in figure 11. When the user clicks on UNHM Course it redirects to UNHM Course page and the user can update the details there. If the user clicks on cancel button instead of update form then the transfer evaluation will not be updated and will be redirected to transfer evaluations page.

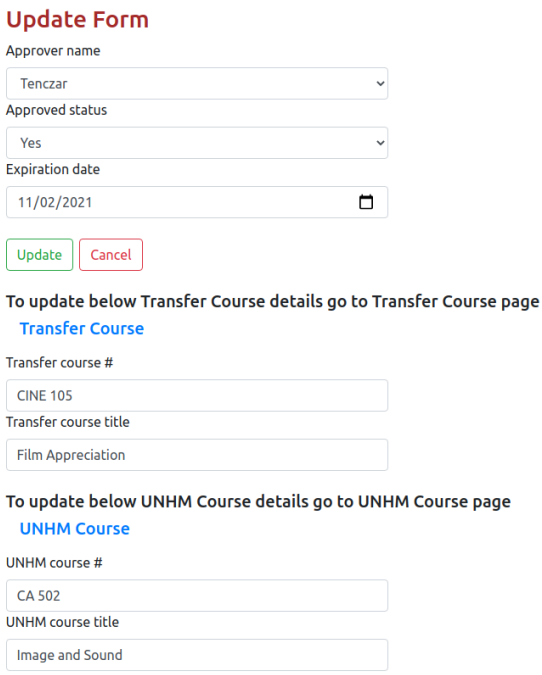

Figure 11: Edit form

5 EVALUATION

The best way to evaluate this upgraded version of Transfer Evaluation service web application is getting feedback from the client or the end-user. Presenting the successfully functioning web service and testing the backend implementation using Postman which is a separate service that talks with the backend.

5.1 Testing Backend Implementation Using Postman

I used Postman to test the backend implementation [3]. There is an option to enter the url for the backend server. There are many options like PUT, DELETE, POST, GET. Once the url is entered, if I select GET option then all the details will be displayed in response

section. If I select POST option then in the body section all the details should be entered and then we need to click on send then the entered details will be posted as shown in figure 12. We can check in the backend server url whether the details is posted or not.

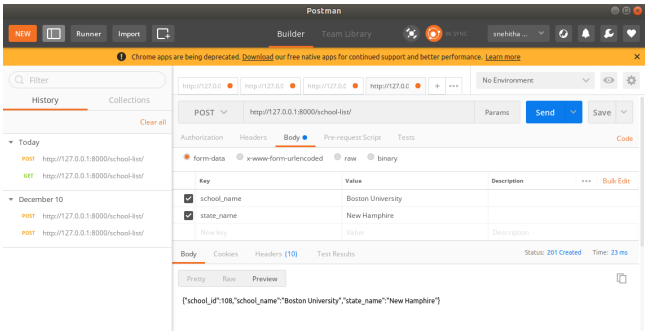

Figure 12: Backend testing with Postman

5.2 Current Limitations and Further Development

(1) One of the processes that will ensure a more comprehensive evaluation of the project is User Interface testing which is used to check whether the user interface is working properly. Selenium is one the UI testing tools that is used to verify the web application UI. Testing UI for the React application will be very useful.

(2) Another important step would be deploying the web application so that the client can check all the functionalities and become familiar with the web application. Deploying to AWS or GCP will be very effective.

(3) In home page, after searching transfer evaluations based on major and school drop-downs, when the user clicks on create button to create a new transfer evaluation major and school values should be directly populated in create form which will be very easy for the user to use the web application.

REFERENCES

[1] DevPedrada. 2021. sort data in ascending and descending toggle in reactjs Code Example. https://www.codegrepper.com/code-examples/javascript/sort+data+in+ascending+and+descending+toggle+in+reactjs

[2] GitHub, Inc. 2021. Linking a pull request to an issue. https://docs.github.com/en/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue.

[3] Thomas Hamilton. 2020. Postman Tutorial: How to Install and use Postman for API Testing. https://www.guru99.com/postman-tutorial.html

[4] Devil Moya. 2021. How to Create Simple App using React and Django. https://morioh.com