CSUDH

DEPARTMENT OF
COMPUTER SCIENCE

Spring 2019

Senior Project Report

Department of Computer Science

California State University, Dominguez Hills

Advising Database

Prepared By

German Bello and Carlito Jude

In
Partial Fulfillment of the requirements

For
Senior Design – CSC 492

Department of Computer Science
California State University, Dominguez Hills
Fall 2019

Committee Members/Approval

Amlan Chatterjee		
Faculty Advisor	***Signature***	***Date***
Committee Member	***Signature***	***Date***
Committee Member	***Signature***	***Date***
Mohsen Beheshti		
Department Chair	***Signature***	***Date***

ABSTRACT

Being a college student comes with many responsibilities. One of these responsibilities is taking care of the classes needed in order to graduate. Students are urged to meet with an advisor each semester before registration to ensure that the requirements are met. During these academic advising, the advisor uses a physical folder containing the student's progress to guide them through graduation. This is good and all, however; having a physical document for such an important information would be problematic in the long run. With the possibilities of unexpected events and a cluttered file cabinet in mind, we are taking the action by digitizing all the documents as well as creating a safe storage for all the required information.

ACKNOWLEDGEMENT

We would like to acknowledge Dr. Chatterjee as well as our fellow students for guiding our group until the end of our project. At first, we were short of ideas during the first presentations, but we managed to get through. Everyone's input during the passed presentations helped us fill up the holes in our project. An example of this was the suggestion for a registration form. The use of registration made more sense for us due to our project's heavy reliance of storing information. We cannot thank them enough for all the things they did. Last but not least, we would also like to thank our group for the co-operation as well as the effort given until the end of this class.

Table of Contents

1 Introduction

This application will create a digital database for secure storage of advising data for Computer Science students. The implementation of a safe and secure login for students to view information related to their advising data. Such data will include information about the students' requirements, classes taken, classes needed and other data related to the students advising information. The target users will be students and advisors, students will be able to register with their school provided "Toromail" and "Student ID". Once registered, students will be able to view their academics. Advisors will have the option to search students academics by student IDs' and be able to update information such as: grades, semester and leaving comments.

2 Background

The language used for the implementation of this project is Java. As for the IDE, we are using Netbeans IDE to take advantage of the built in window builder. And since the idea of this project is storing and retrieving information, we used MySQL for local database. Using this, we created all related tables, their constraints, design of schemas and relationships between the tables.

2.1 Applications

Application of the program can be implemented in educational institutions for the purpose of securely storing and retrieving data related to a students' academics. The end users will be both educators and students, with both having access to academic records or folders. With both students and advisors able to login and view their courses, we can facilitate the advisement process. This will allow schools to digitize physical school records for safer storage, accessibility and recovery.

2.2 Database

Using MySQL database management system for storing and retrieving data. Implementing a relational database in order to relate each corresponding data entry and linking it to unique IDs'. Every student and advisor has a unique ID in order to facilitate tracking of entities within the database. The relation between classes and courses also each have unique IDs, therefore creating their own tables in the database for better organization of data.

3 Research

3.1 Java

Using the java language, we implemented all the required APIs. One of these APIs is JDBC which establishes a link between our program and databases. We have also used JavaMail which allowed our program to send an email.

3.2 Netbeans IDE

When choosing an IDE to use, we were set on using Eclipse until we found out that it does not have a built in window builder. Upon our research, we came upon gui builder turorials where most of them used Netbeans. Instead of downloading the required materials for eclipse to work, it made more sense for us to take advantage of the built in features.

3.3 MySQL 8.0

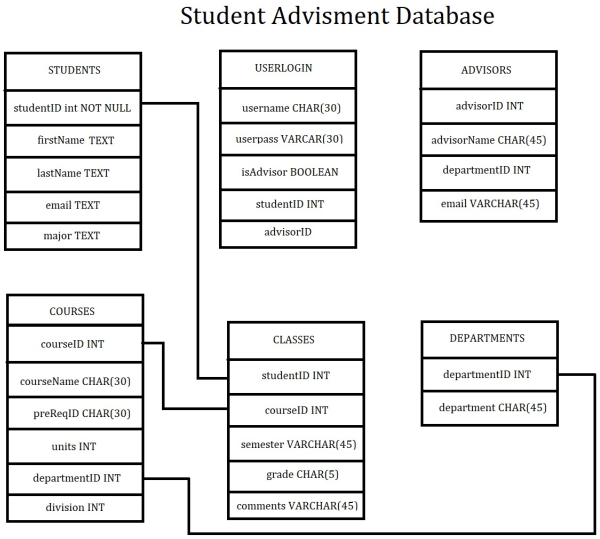
MySQL is an open-source database management system which will be storing all of our data. This relational DBMS helped in the implementation of the application simply by being user-friendly and having a simple learning curve. We were able to retrieve all data and secure fields such as passwords using mySQL two-way encryption.

3.4 Databases and Tables

The design and creation of tables within the relational database is useful for finding entities within the database. These entities can be a students grades received, classes in progress or taken, and other information related to particular students. It can also help track a students total units taken or for learning how many students are currently enrolled in classes and other information.

4 Development

4.1 MySQL Database



4.1.1 Implementation

Implementation of the database was challenging but we managed to establish a connection using “Connector /j 5.1.48” which we needed to import into our library. Once imported, we were able to successfully connect to our local database. After we were able to do this, we were able to create our own database schema that consists of 6 tables which are students, courses, classes, userlogin, departments, and advisors. Each of these tables are filled using mySQL commands in the registration we will be creating. The names, emails, usernames, passwords, and such will be some of the attributes for our tables as shown above. The studentID will be primary key for the STUDENTS table that will be related to CLASSES table and act as one of the foreign keys. Just like the STUDENTS table, the COURSES will also be sending its primary key to CLASSEs to make connection from one another. Once the registration is set up, the login window will be searching through these tables to check if the required attribute are present at its respective Entity. These relations are essential to our project because it gives the users all of the required information when pulling out the folders.

4.1.2 Challenges

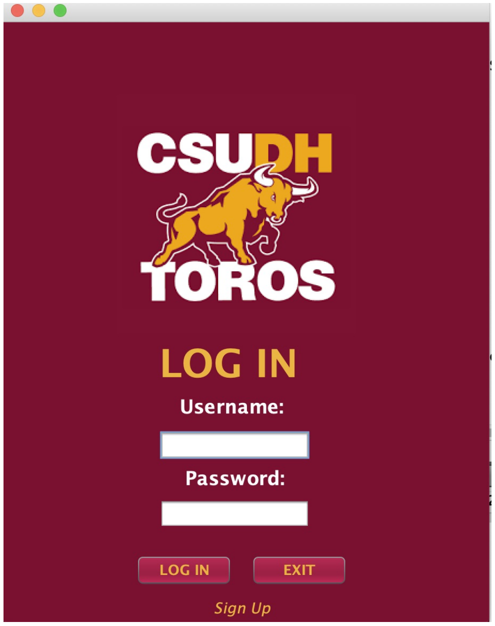
The main challenges in creating the database was learning the sql statements and understanding the relationships between the tables for accessing the information. Another equally challenging obstacle was establishing a connection to the database. One of us were able to connect to the database while the other could not, due to pathing issues. However, this problem has become apparent after some research. We found out that the problem was the project type we created from the beginning. Apparently, creating a project from apache netbeans does not allow us to direct our work to the desired path easily. Therefore, we created different type of project where we were able to direct the "Connector /j 5.1.48" easily into our work.

4.2 GUI

The graphical user interface(GUI) that was used for this project is Java Swing. We chose Swing because we did not have background on using GUI with the passed projects during our college years. However, we did some activities using Swing back in java intro programming where we had to do it manually and thus, how it came about. The actual RGB colors of the school are also used for all of the jFrames, matching the school logo we have used in the log in screen.

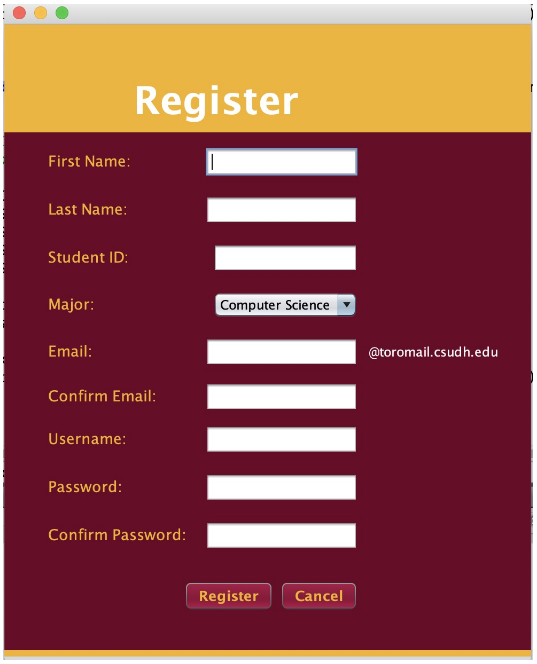
4.2.1 User Login

For implementing our user login screen we had to make sure during registration of either student/advisor that a valid CSUDH provided email was used as well as the schools provided student IDs are required to register. Depending on who logs in (Student/Advisor), it would have different output.

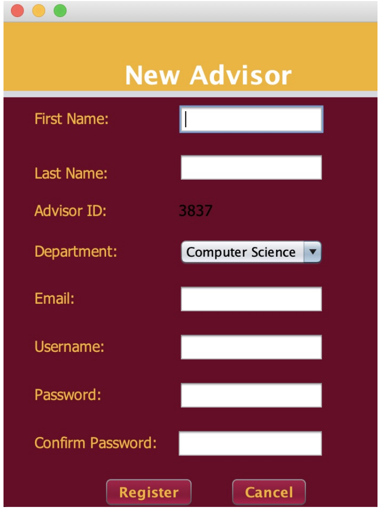


4.2.2 Registration

Registration will be required in order to login and view advising information for both advisors and students. The students can easily access the registration form from the log in screen.As for the advisor, however; an existing advisor has to register the new advisor.



Major: Computer Science



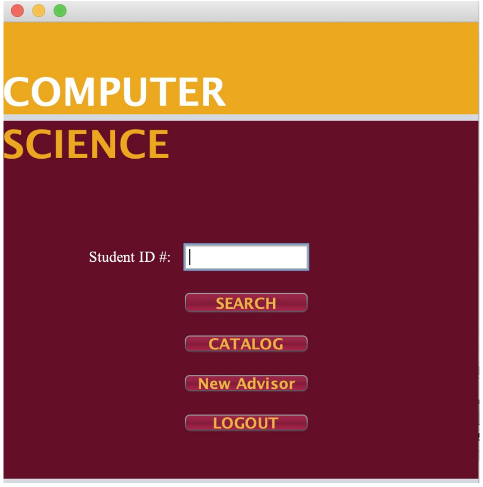
Advisor ID: 3837

Department: Computer Science

Since the registration form for an advisor can only be accessed by an existing advisor, the form was created inside the advisor home page.

4.2.3 Advisor Home Page

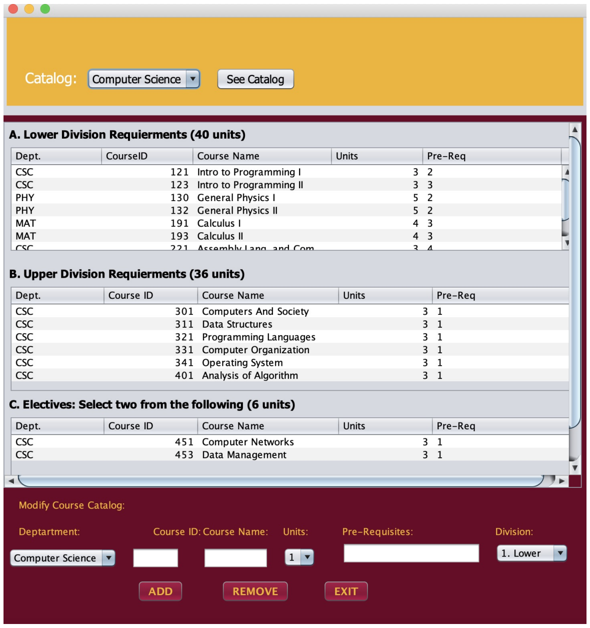
Logging in as an advisor will bring the advisors home page. From here, an advisor will be able to search students by their student ids, register new advisors, view and modify course on catalogs.



If the entered student ID exist in the database, it would pull out students' folder containing all the associated information.

4.2.4 Catalog

This JFrame is designed after the catalog that students would usually see before registering to a class. It shows all the classes required based on the students' major. This catalog however is a little different because it is only accessible to advisors. From there, they can add classes into the system as well as remove these classes if the school decided to reduce or increase the requirements for csc majors.


Catalog: Computer Science

A. Lower Division Requierments (40 units)

Dept.	CourseID	Course Name	Units	Pre-Req
CSC	121	Intro to Programming I	3	2
CSC	123	Intro to Programming II	3	3
PHY	130	General Physics I	5	2
PHY	132	General Physics II	5	2
MAT	191	Calculus I	4	3
MAT	193	Calculus II	4	3
CSC	221	Assembly Lang. and Com	3	4

B. Upper Division Requierments (36 units)

Dept.	Course ID	Course Name	Units	Pre-Req
CSC	301	Computers And Society	3	1
CSC	311	Data Structures	3	1
CSC	321	Programming Languages	3	1
CSC	331	Computer Organization	3	1
CSC	341	Operating System	3	1
CSC	401	Analysis of Algorithm	3	1

C. Electives: Select two from the following (6 units)

Dept.	Course ID	Course Name	Units	Pre-Req
CSC	451	Computer Networks	3	1
CSC	453	Data Management	3	1

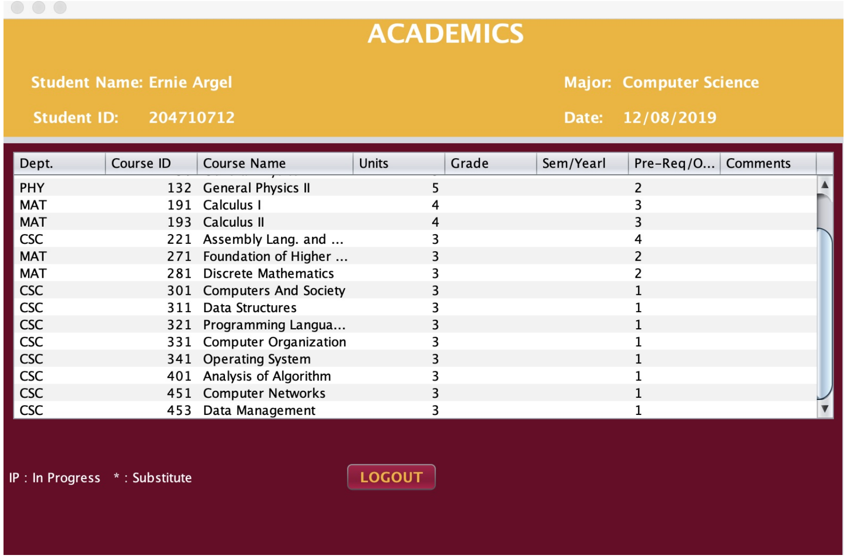
Modify Course Catalog:

Deptartment: Computer Science
Course ID:
Course Name:
Units: 1
Pre-Requisites:
Division: 1. Lower

The form below this catalog allows advisor to add and remove classes. Once it is filled, both the advisor's form as well as the student form will be updated.

4.2.5 Student Page/ Folder

Student page will display the students information and all available courses for their major (CS) as well as view all the courses they have taken, grades and comments received. After a student registers, he or she will be able to access this page. The design of this page was based on actual students folder that the advisor uses during the advising.



ACADEMICS

Student Name: Ernie Argel

Major: Computer Science

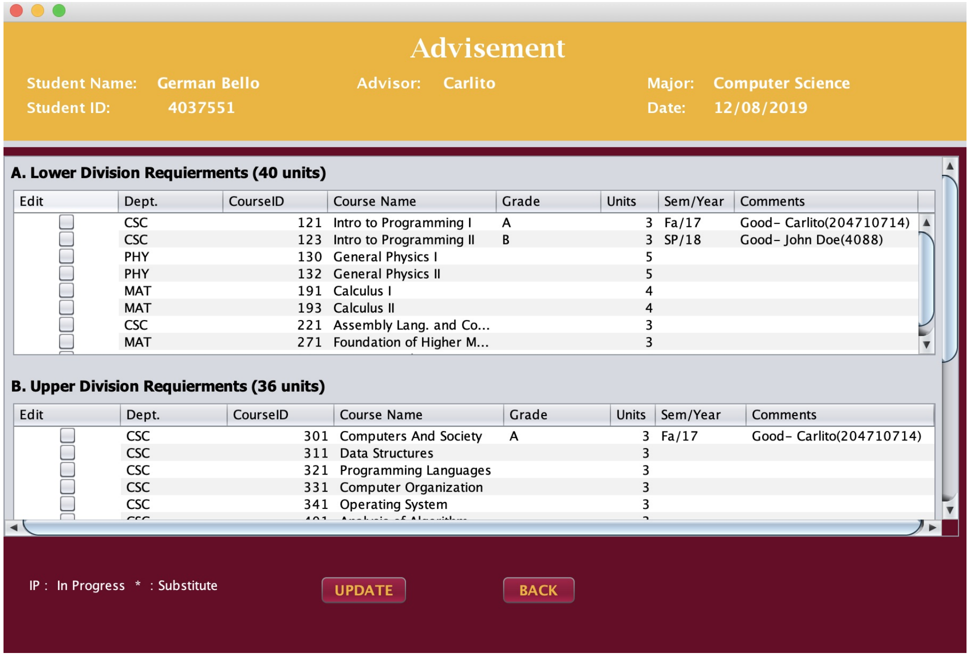
Student ID: 204710712

Date: 12/08/2019

Dept.	Course ID	Course Name	Units	Grade	Sem/YearI	Pre-Req/O...	Comments
PHY	132	General Physics II	5			2	
MAT	191	Calculus I	4			3	
MAT	193	Calculus II	4			3	
CSC	221	Assembly Lang. and ...	3			4	
MAT	271	Foundation of Higher ...	3			2	
MAT	281	Discrete Mathematics	3			2	
CSC	301	Computers And Society	3			1	
CSC	311	Data Structures	3			1	
CSC	321	Programming Langua...	3			1	
CSC	331	Computer Organization	3			1	
CSC	341	Operating System	3			1	
CSC	401	Analysis of Algorithm	3			1	
CSC	451	Computer Networks	3			1	
CSC	453	Data Management	3			1	

IP : In Progress * : Substitute

This folder is the student form that is accessible to students. With this, they can only view their folders.



Advisement

Student Name: German Bello

Advisor: Carlito

Major: Computer Science

Student ID: 4037551

Date: 12/08/2019

A. Lower Division Requierments (40 units)

Edit	Dept.	CourseID	Course Name	Grade	Units	Sem/Year	Comments
☐	CSC	121	Intro to Programming I	A	3	Fa/17	Good- Carlito(204710714)
☐	CSC	123	Intro to Programming II	B	3	SP/18	Good- John Doe(4088)
☐	PHY	130	General Physics I		5		
☐	PHY	132	General Physics II		5		
☐	MAT	191	Calculus I		4		
☐	MAT	193	Calculus II		4		
☐	CSC	221	Assembly Lang. and Co...		3		
☐	MAT	271	Foundation of Higher M...		3		

B. Upper Division Requierments (36 units)

Edit	Dept.	CourseID	Course Name	Grade	Units	Sem/Year	Comments
☐	CSC	301	Computers And Society	A	3	Fa/17	Good- Carlito(204710714)
☐	CSC	311	Data Structures		3		
☐	CSC	321	Programming Languages		3		
☐	CSC	331	Computer Organization		3		
☐	CSC	341	Operating System		3		
☐	CSC	401	Analysis of Algorithm		3		

IP : In Progress * : Substitute

This is the Advisor Form that is accessible from the Advisor Home Page(4.2.3). From here, the advisor can update students folder by inputing grades, year taken and comments.

4.2.6 Challenges

Implementing Java Swing into this project was not really difficult but it did have some learning curve due to lack of experience. At first, we started working on GUI in which we thought would be difficult because we were taught the hard way. The hard way meaning creating it from the scratch. Funny enough, we created our first login screen this way and it was tedious and difficult at the same time because we had to worry about all the alignment every single time. But after some research, we were able to learn the easier way to work around it which is the use of window builder. From there, we were able to easily plan out what we wanted to do with the project.

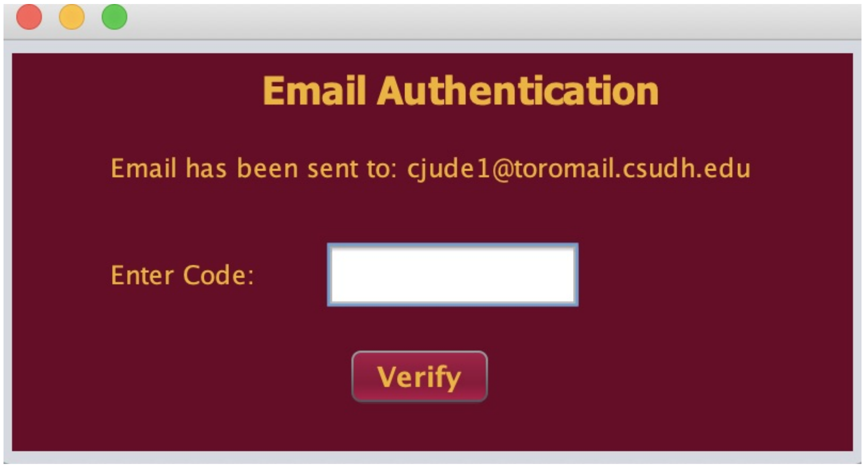
4.3 Security

4.3.1 Password

In order to protect and securely store sensitive information such as usernames and passwords we implemented MySQL's two-way encryption. One of which takes a string such as the password, hashes it and stores it in the database. Therefore, the actual password is never stored.

4.3.2 Email Authentication

Another feature for extra security was implementation of email authentication. When a new student/advisor registers they should register using the CSUDH provided email such as: " *@toromail.csudh.edu". We were able to send an email confirmation number/code which is required to be entered in the next window in order to finalize the registration.



Email Authentication

Email has been sent to: cjude1@toromail.csudh.edu

Enter Code:

This email authentication would pop out for finalizing the students acount creation. The idea for this is to make sure that the email used is real.

4.3.3 Challenges

The difficulty we came upon this section was the fact it was our first time working on these. But since we have already figured out how to direct files to our library, we really did not have the same problem directing the ”javax.mail.jar” into our project.

5 Conclusion & Future Work

Working on this project gave us insight on how to create a fully build application. This experience gave us a chance to experience and learn what cannot gained during lectures or activities in class. The implementation of database was really an eye opening because we really had no experience at all. To think that three months ago, we were clueless about most of these but now we felt knowledgeable. We thought that this is something to be proud about.

Currently we have implemented and tested all of the features required to facilitate the advisement process. Advisors will be able to search students folders with the ability to modify a students courses, as well as changing the course catalog for all majors and register new advisors. Students will be able to login and view their academics and view all necessary courses for their majors.

As for the future work, we can expand our program to other majors as well as creating a server for it. We felt that if we were given more time, we would have been able to add more majors related to computers since some of the courses required are already in our database. We also thought that if we really want our program to be accessible for everyone, we would have to make a server for it.

6 References

References

mySQL connector package -

https://stackoverflow.com/questions/2839321/connect-java-to-a-mysql-database

Java mail package -

https://stackoverflow.com/questions/6606529/package-javax-mail-and-ja vax-mail-internet-do-not-exist

Retrieve data from database -

https://www.tutorialspoint.com/jdbc/jdbc-select-records.htm

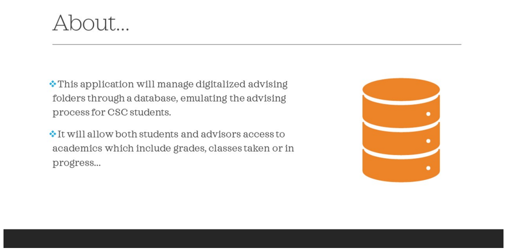
7 Appendix

7.1 Final Presentation

Advising Database

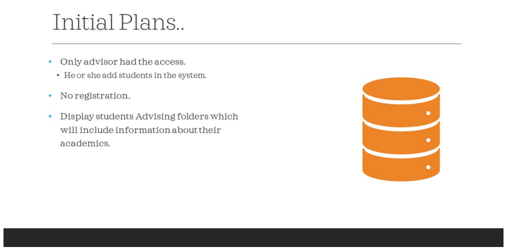
BY: GERMAN BELLO

& CARLITO JUDE



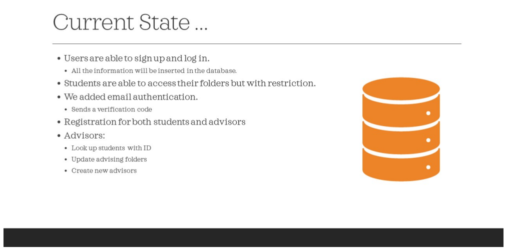
About…

- This application will manage digitalized advising folders through a database, emulating the advising process for CSC students.
- It will allow both students and advisors access to academics which include grades, classes taken or in progress…



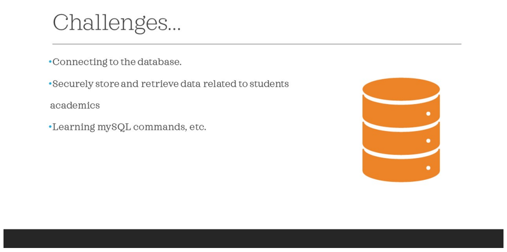
Initial Plans..

- Only advisor had the access.
  - He or she add students in the system.
- No registration.
- Display students Advising folders which will include information about their academics.



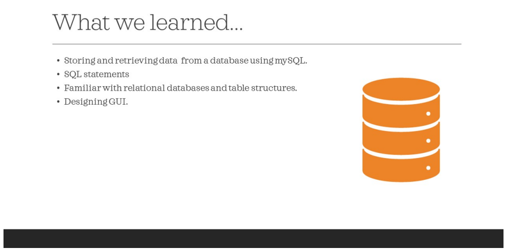
Current State …

- Users are able to sign up and log in.
  - All the information will be inserted in the database.
- Students are able to access their folders but with restriction.
- We added email authentication.
  - Sends a verification code
- Registration for both students and advisors
- Advisors:
  - Look up students with ID
  - Update advising folders
  - Create new advisors



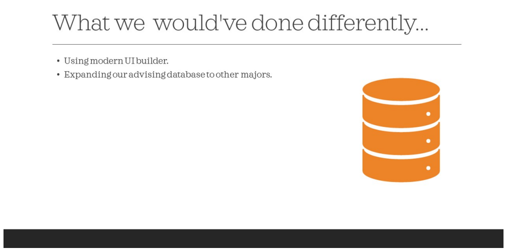
Challenges…

- Connecting to the database.
- Securely store and retrieve data related to students academics
- Learning mySQL commands, etc.



What we learned…

- Storing and retrieving data from a database using mySQL.
- SQL statements
- Familiar with relational databases and table structures.
- Designing GUI.



What we would've done differently…

- Using modern UI builder.
- Expanding our advising database to other majors.