Proposal Feedback:

This is quite a unique and ambitious project. I believe you have already spent a lot of time defining the functionality for this project and you are already showing what can be achieved with an active three-person group. Please start with the database design for the student/university schema.
Please answer the following questions in the progress report.

Have any of you studied natural language processing?
- None of us have experience with NLP, but Anirudh has some experience with more general machine learning concepts and has worked with neural nets and other kinds of classifiers. The IBM Learning Lab (https://www.ibm.com/us-en/marketplace/learning-lab) has numerous courses and guides that can help developers inexperienced with AI and NLP integrate these technologies into their solutions.
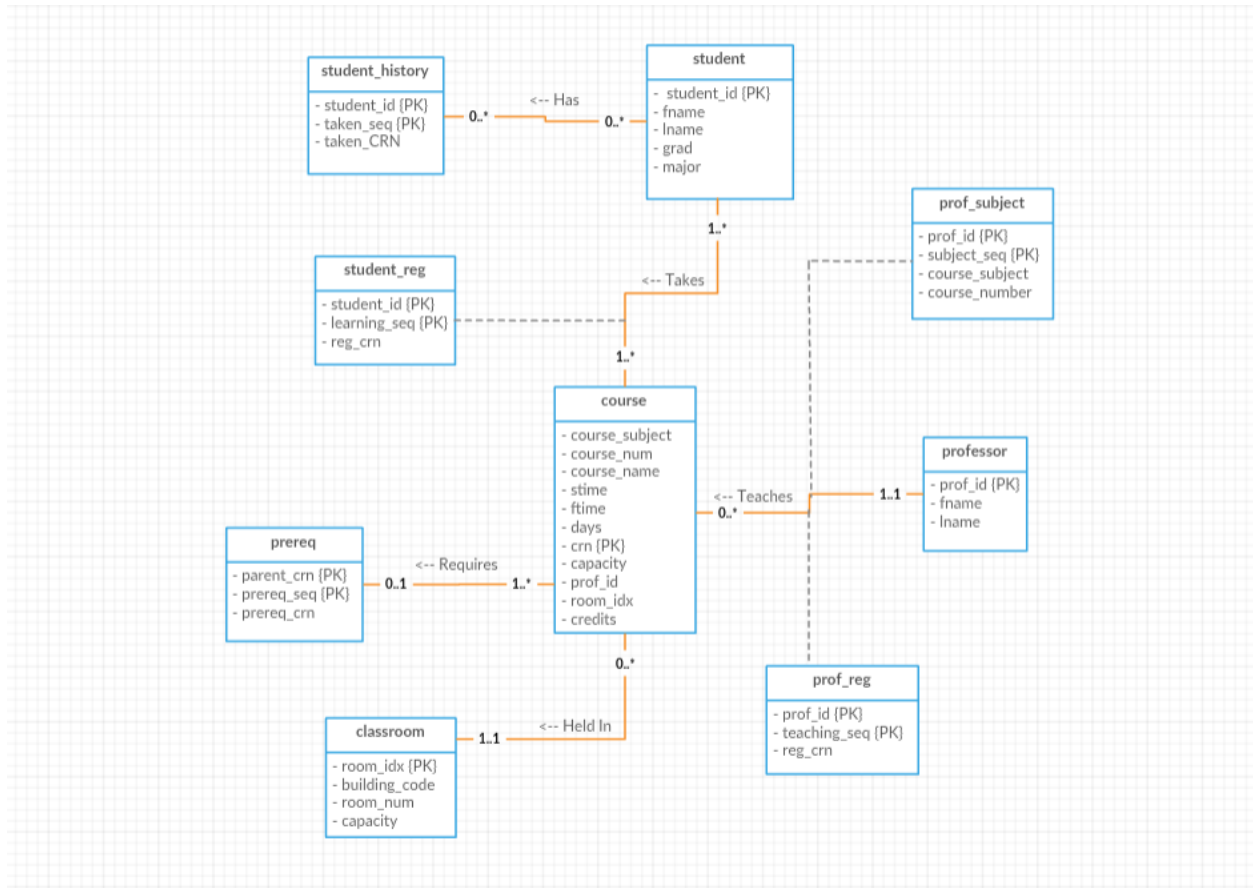
Do you have experience with Watson?
- It is important to mention that the development of the Watson interface is the last hurdle we hope to overcome in this project. None of us have experience working with Watson or Bluemix, but we hope to learn a lot through this project. However, our first goal is to develop a simplistic interface that establishes to the user what functionality they have in interacting with the database. Once we develop such a platform, we will attempt to grow it using the Watson NLP technologies.

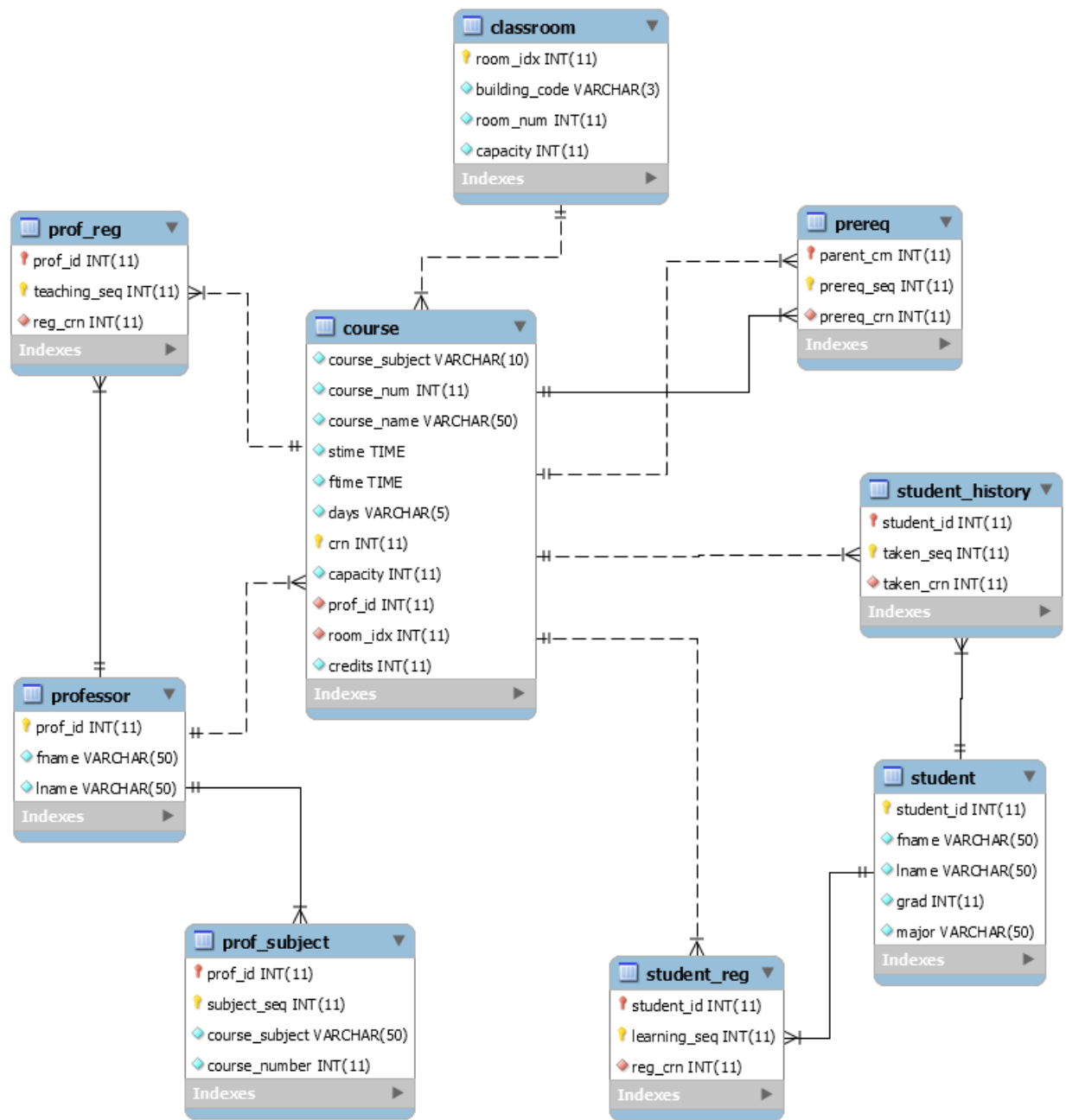*How do you plan to represent the user interface/natural language representation?*
- As part of Bluemix developer cloud toolkit, the Conversation and Discovery app (http://conversation-enhanced.mybluemix.net/) provides basic interface and chatbot functionality for developers. Again, before we aim to develop the tools for this representation, our basic interface will be a simple web client where users can select from a set of actions/commands and provide their parameters. Our first goal is to have a simple website with just a drop-down list of functions the user can use to engage with the database (simple CRUD operations). We hope to evolve this eventually to a chatbot-style interface with the integrated NLP technologies.

Will you use MySQL to represent the user's limited English?
- The amount of flexibility that we allow in terms of the user's grasp of the English language is determined by our training of our NLP classifier. We will have considered this project a success if we can set up a classifier to be trained by training data, despite not having actually trained it. With the state of AI technology and the extent of our background in the subject, I do not think an expectation of proper functionality with users who do not know complete English is realistic.

student_history
- student_id {PK}
- taken_seq {PK}
- taken_CRN

<-- Has   0..*   0..*

student
- student_id {PK}
- fname
- lname
- grad
- major

prof_subject
- prof_id {PK}
- subject_seq {PK}
- course_subject
- course_number

1..*

student_reg
- student_id {PK}
- learning_seq {PK}
- reg_crn

<-- Takes

1..*

course
- course_subject
- course_num
- course_name
- stime
- ftime
- days
- crn {PK}
- capacity
- prof_id
- room_idx
- credits

<-- Teaches   1..1   0..*

professor
- prof_id {PK}
- fname
- lname

prereq
- parent_crn {PK}
- prereq_seq {PK}
- prereq_crn

<-- Requires   0..1   1..*

0..*

prof_reg
- prof_id {PK}
- teaching_seq {PK}
- reg_crn

classroom
- room_idx {PK}
- building_code
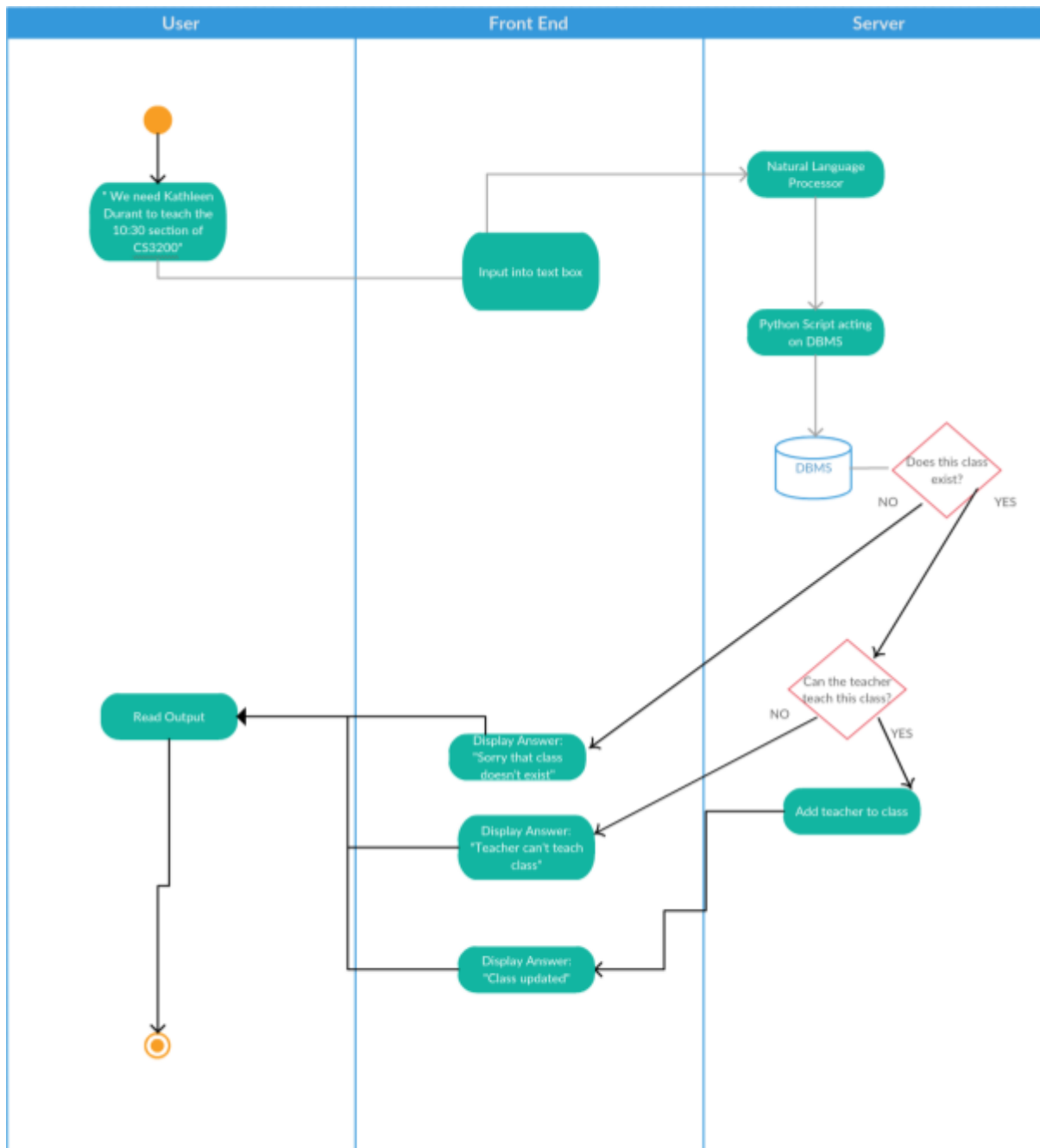- room_num
- capacity

<-- Held In   1..1

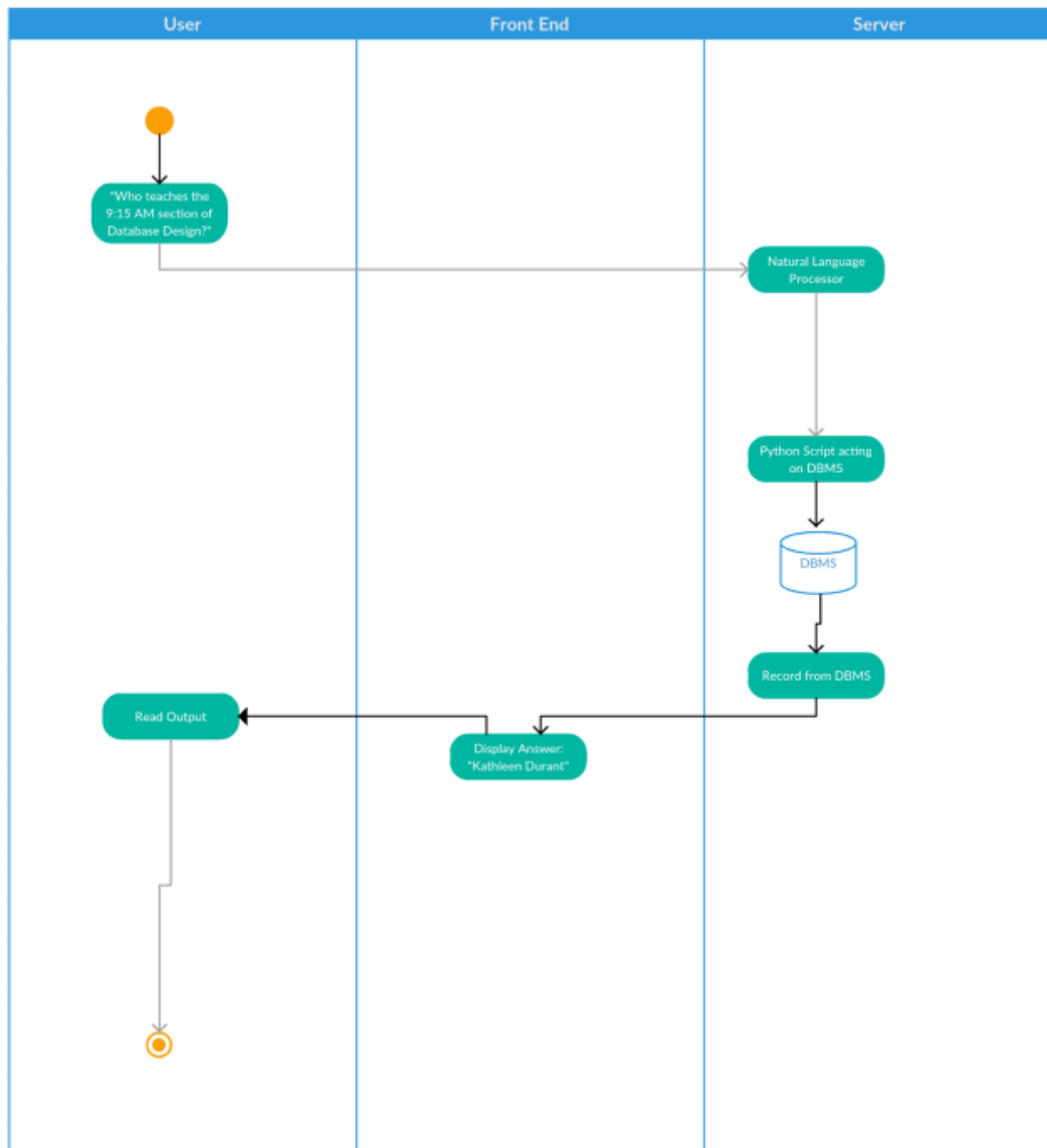*UML Diagram structuring the contents of our database.*

EER Diagram establishing the structure of our database.

These flowcharts outline some of the basic CRUD operations our database application will allow.

User	Front End	Server

Natural Language Processor

"We need Kathleen Durant to teach the 10:30 section of CS3200"

Input into text box

Python Script acting on DBMS

DBMS

Does this class exist? — NO / YES

Can the teacher teach this class? — NO / YES

Read Output

Display Answer: "Sorry that class doesn't exist"

Display Answer: "Teacher can't teach class"

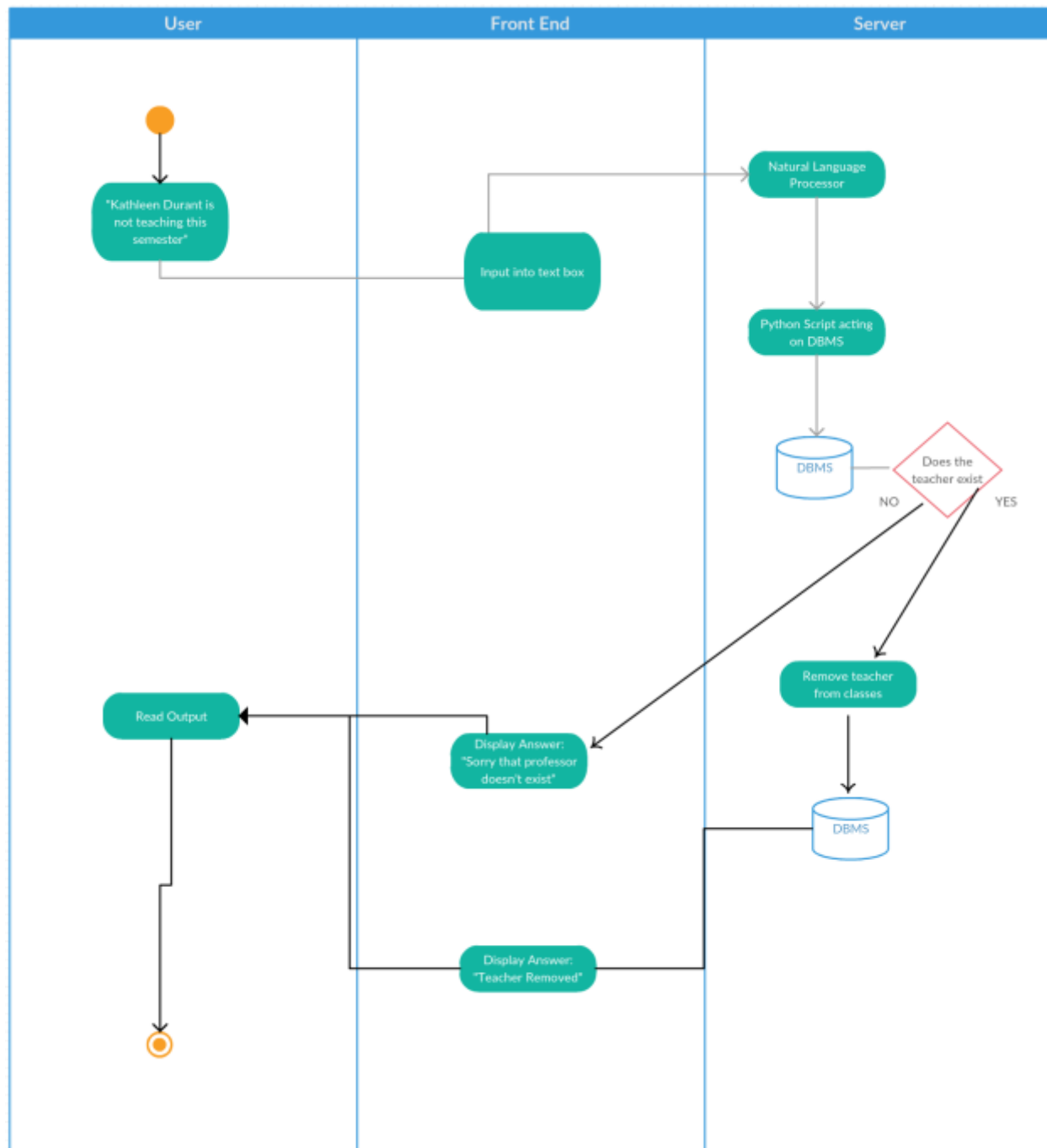Add teacher to class

Display Answer: "Class updated"

*This flowchart describes an **update** operation on the database. The user wishes to update a specific Course by adding a Professor to that Course. The user inputs the text, which is sent to the Natural Language Processor. This parses the relevant information from the English language request and translates it into a MySQL command. The Python script runs this command on the database. To ensure data integrity, triggers are put in place to check if the update command can occur given the constraints of the data. If it cannot occur, the user receives feedback regarding the error. Otherwise, the update occurs and the user is notified.*

User	Front End	Server

User:
- "Who teaches the 9:15 AM section of Database Design?"
- Read Output

Server:
- Natural Language Processor
- Python Script acting on DBMS
- DBMS
- Record from DBMS

Front End:
- Display Answer: "Kathleen Durant"

*This flowchart describes a **reading** operation on the database. The user asks a question regarding the Professor of a specific Course. The Natural Language Processor parses the relevant information from this English language query and translates it into a MySQL query. The Python script then runs the query on the database, producing the desired Teacher's name as output. The output is then displayed to the user.*

User	Front End	Server

- "I want to take CS3200 at 9:15 AM"
- Input into text box
- Natural Language Processor
- Python Script acting on DBMS
- DBMS
- Does the class exist — NO / YES
- Does the class have seats open? — NO / YES
- Do the user have conflicting classes? — YES / NO
- Display Answer: "Sorry that class doesn't exist"
- Display Answer: "That class is full"
- Display Answer: "You have another class at that time"
- Display Answer: "Schedule Updated"
- Read Output
- DBMS
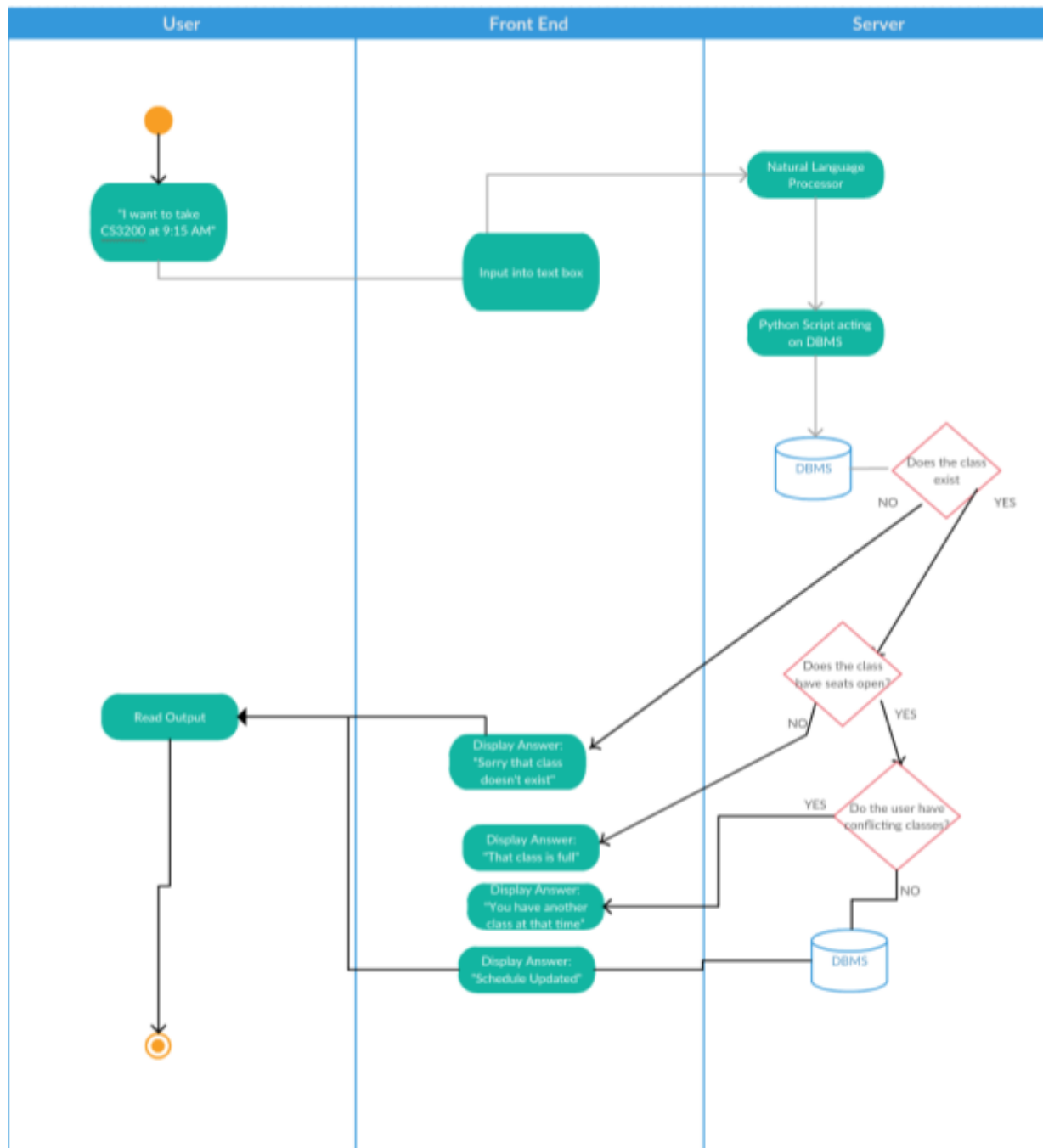
*This flowchart describes an **update** operation on the database. The user wishes to update their Schedule by adding a specific Course. The specified input is added appropriately into some text fields on the web client. The fields are then sent to the server, where a Python script calls the appropriate stored routine to update the database. To ensure the integrity of the data in the database, numerous triggers are put in place to check that the update operation does not violate any restrictions on the data. If the operation is invalid, the appropriate error is sent back to the user. Otherwise, a confirmation that the operation occurred is sent.*

User	Front End	Server

- "Kathleen Durant is not teaching this semester"
- Input into text box
- Natural Language Processor
- Python Script acting on DBMS
- DBMS
- Does the teacher exist — NO / YES
- Remove teacher from classes
- DBMS
- Display Answer: "Sorry that professor doesn't exist"
- Display Answer: "Teacher Removed"
- Read Output

*This flowchart describes a **deletion** operation on the database. The user wishes to remove a specific Professor from the database. The input is parsed by a Natural Language Processor and it determines that a deletion operation has been requested. The processor produces a MySQL deletion command and sends it to a Python script, which acts on the database. A trigger checks if the specified Professor exists. If so, the Professor is removed from the database and the user is notified. Otherwise, the user is told that the Professor does not exist.*

Technical Specifications

The database will be constructed and manipulated in MySQL.

The front end will be written in HTML and JavaScript to provide the behavior of the webpage. Once we complete the basic UI, we will integrate the Bluemix chatbot-style interactions. We have to do a bit more research into this library, but we know that it can be integrated using JavaScript.

The information from the webpage will be sent to a Python script using Flask. The Python script will then manipulate the database using MySQL scripts written through the MySQLConnector library. (we potentially will be using PHP for this level instead).

The results of the queries will be sent back to the UI using a PHP script.

Here is a simple diagram of what we aim to accomplish: