KU MEDICAL CENTER The University of Kansas

TENNESSEE TECH UNIVERSITY

Never-Ending Medical Operative (NEMO)

High Level Design Document
Version: 1.2

Authors:
Cody Moffitt
Jackson Chandler
Andrew Pachuilo
Oliver Payne

Table of Contents

Chapter 1: System Overview

1.1: System Overview

The Never-Ending Medical Learner (NEMO) is being developed for the Kansas University Medical Center as a tool for machine learning research. The primary purpose of the system is to aid domain experts in developing predictions relating to patients. The system uses machine learning techniques in order to make conclusions and predictions from prior data. The system shall operate by taking questions from the domain experts and creating learning algorithms to generate predictions. The system shall make use of a feedback mechanism, where users must give responses to the learners based on the prediction models they develop.

The system's structure is generalized in order to allow the addition of other machine learning techniques. However, the basic framework for the system has been constructed. The NEMO system shall be comprised of a data mart and three major components: the AI controller, the data loader, and the web application. The AI controller will be used to create and manage learning algorithms in response to questions and feedback given by domain experts. The data loader will be responsible for loading test data and learner data into the data mart from KUMC's external database. The data mart will hold all of the test and learner data, as well as all of the users' questions and feedback. Finally, the web application will be the domain experts' interface with the NEMO system, where they will be able to create questions and submit feedback.

The purpose of this document is to provide a high level design perspective for the NEMO system, which includes outlines for the system's architecture as well as how each of the system's components interact with each other. This document also demonstrates how users are expected to interact with core components of the web application. An overview of features available to users and administrators is also included. Finally, the document concludes with a description of how data is stored within the data mart.

1.2: Glossary

NEMO: Never-Ending Medical Operative. The project is based off of the Never-Ending Language Learner (NELL) system. The goal is to create a system that continually learns on real-world data in order to make predictions and answer medical related questions.
Data Mart: A smaller subset of the medical database. Entries here will be partitioned into test data and learner data. The learners will use this data to answer questions and form conclusions.
Database: A collection of a large amount of data. This data is organized into entries of records. Each record is made up of several components.
i2b2: A National Center for Biomedical Computing which developed a biomedical informatics framework funded by the U.S. National Institutes of Health.
AI: Artificial intelligence. It is the process of simulating one or more human characteristics with a computer. In this particular case, the artificial intelligence is machine learning.
Classifier Algorithm: An algorithm that is purposed with evaluating data and new classifications.
AI Node: A particular instance of AI algorithms.
Learner: An algorithm that is purposed with evaluating data and forming conclusions based off of it. A machine learning algorithm.
HIPAA: Health Insurance Portability and Accountability Act.
KUMC: Kansas University Medical Center.
Soft Edit: An edit of a question that creates a new question in the database.
Hard Edit: An edit of a question that changes the original question in the database without creating a new one.
Global Dashboard: This is the area of the web client where users can view other users' questions.
User Dashboard: This is the area of the web client where users can view only their own questions.

Chapter 2: System Architecture

2.1: Architectural Design

The diagram below shows the context model for the Never Ending Medical Operative system. An in-depth description of each component can be found in the Decomposition section.

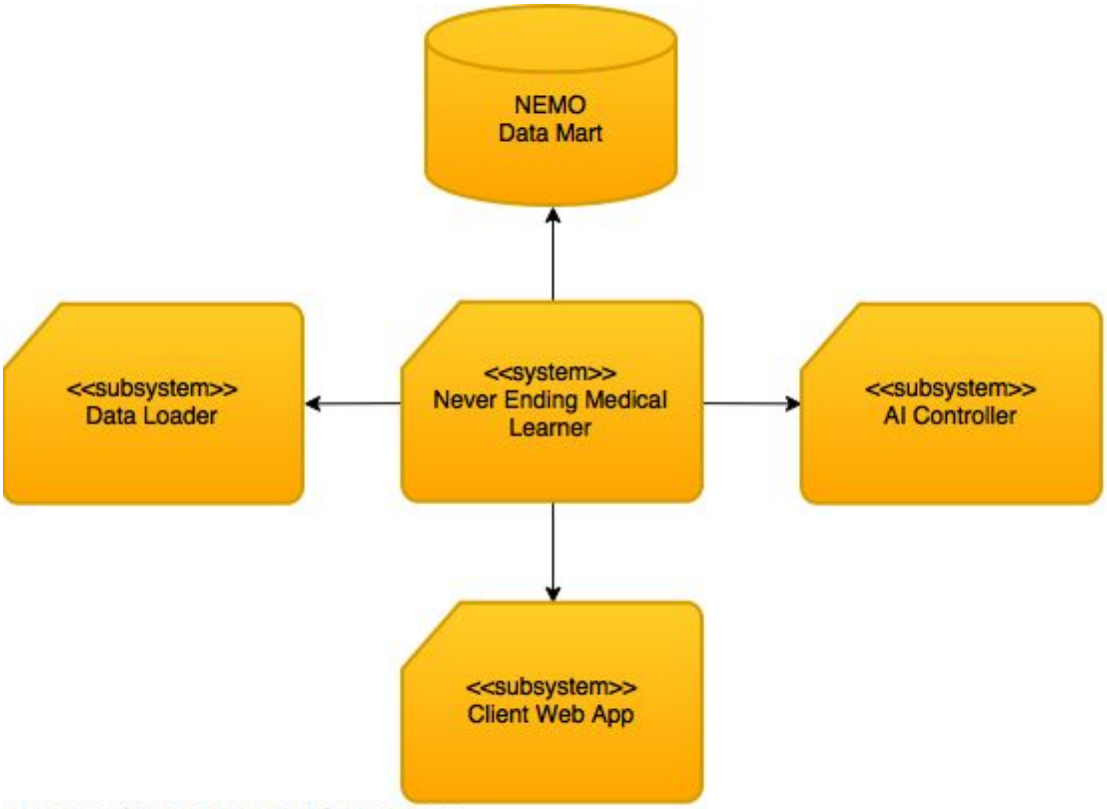

Figure 1: Context Model of Subsystems

The NEMO system is comprised of three major components and a data mart, as can be seen above. The Client Web App is the user's primary interface with the system. The Client Web App is where the domain experts will pose questions and give feedback to the learning algorithms. The AI Controller is the component responsible for handling the creation and training of the learning algorithms. The AI Controller is also tasked with responding to the users' questions and feedback by adjusting the learners. The Data Loader pulls records from the KUMC source database and stores them on the data mart. The data retrieved by the Data Loader will be used by the learners as they form predictions. The data mart holds information used by all components of the system. The data mart is explained and diagrammed in further detail later in this document (Chapter 4).

2.2: Decomposition

The decomposition model breaks down the Architectural Design into smaller components. The tasks and responsibilities of each subsystem can be seen in the model below. In addition to this, the connections and communications between each piece can be seen.

As the figure 2 depicts, all components of the system will be connected to the NEMO Data Mart. The three major components of the system will be able to load and store information in the data mart. The data mart holds a subset of patient data to be used by the learners, the questions that the domain experts will pose, and the feedback that the domain experts submit to the learners. This data mart is created and loaded with a subset of i2b2 medical records via the data loader system. Domain experts will be posing their questions via the web client, which stores them on the data mart. The AI controller sees these questions and creates learners to handle them. These learners use patient data from the data mart.

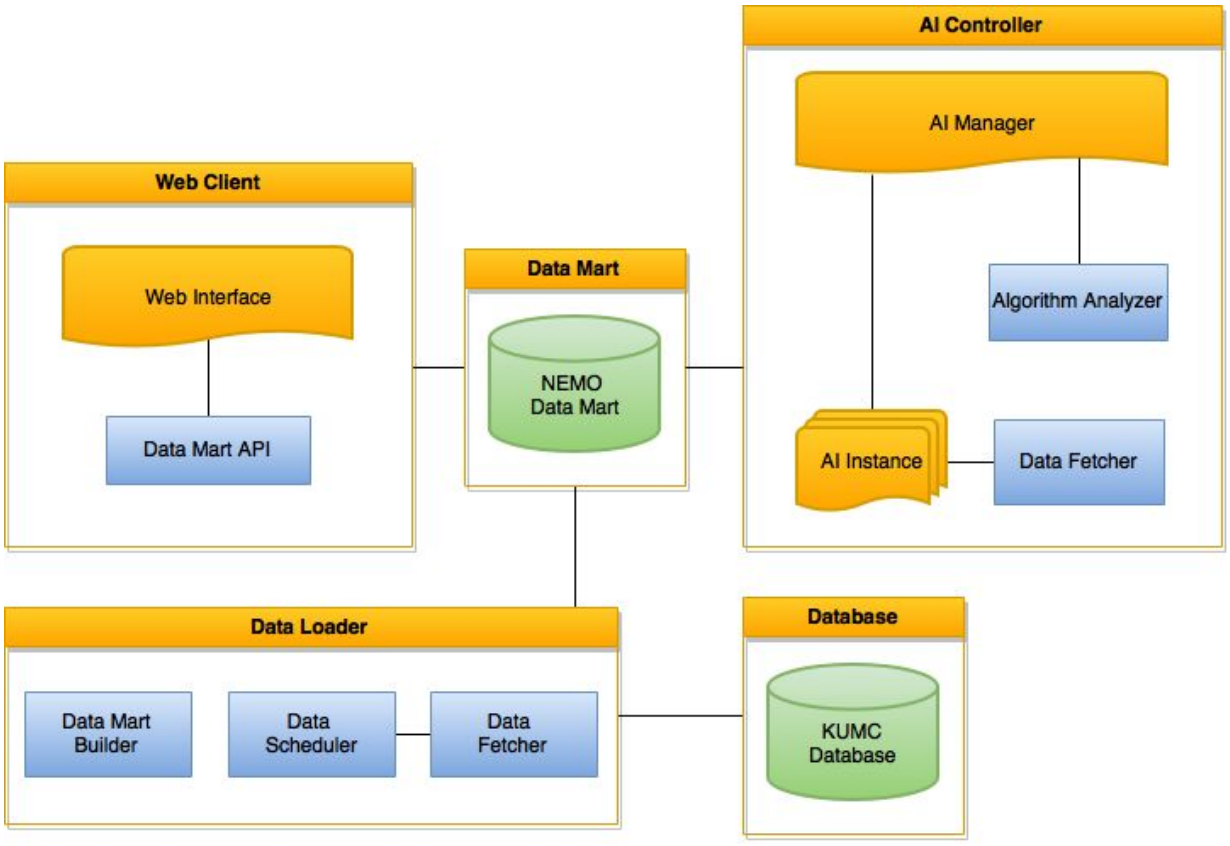

Figure 2: Decomposition of the system's subsystems and associated relationships

2.3: Component Design

2.3.1: Use Case Diagram

The use case diagram below shows possible interactions between client users/administrators and the NEMO system. User interaction begins with the web client and from there they can take the possible actions below. Administrators can interact with the system directly to load and schedule new data for the data mart.

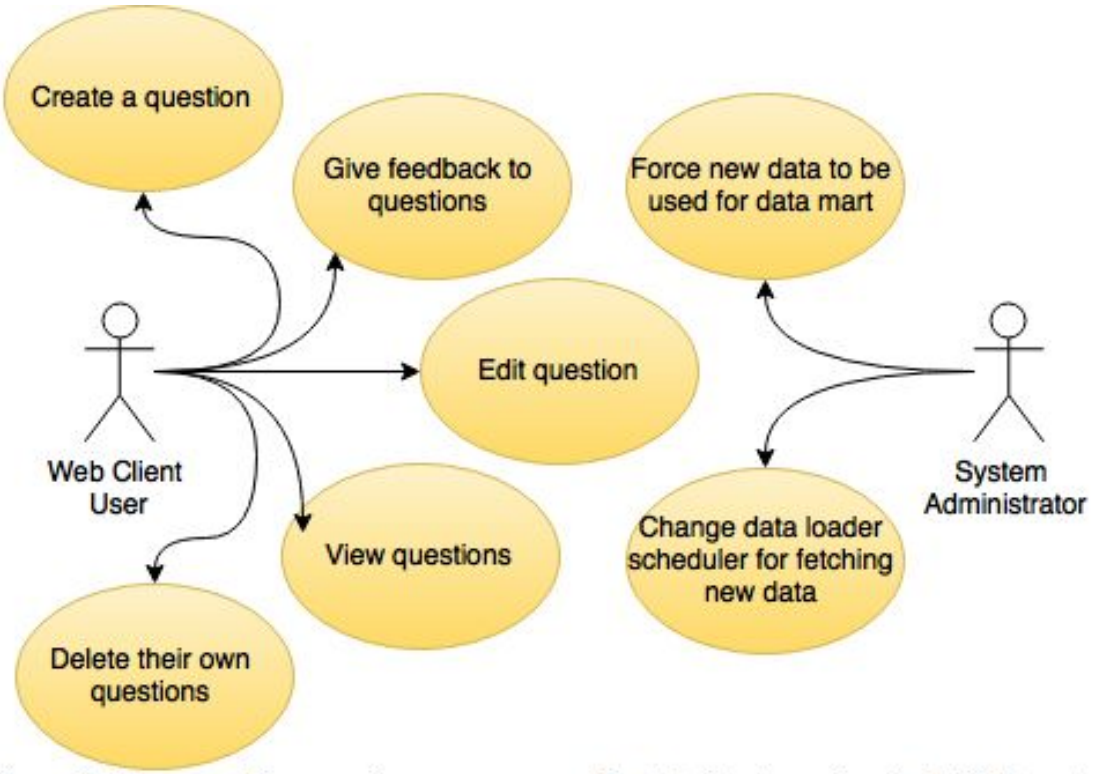

Figure 3: Use case diagram of a super user and administrator using the NEMO system

2.3.2: Sequence Diagram: User Poses a Question via Web Client

This sequence diagram walks through the process of posing a question to the AI. When the user is satisfied with the question, they will click submit and send their question to the database. The database will notify the web client of the details of the submission and the user shall be able to see the status for the question.

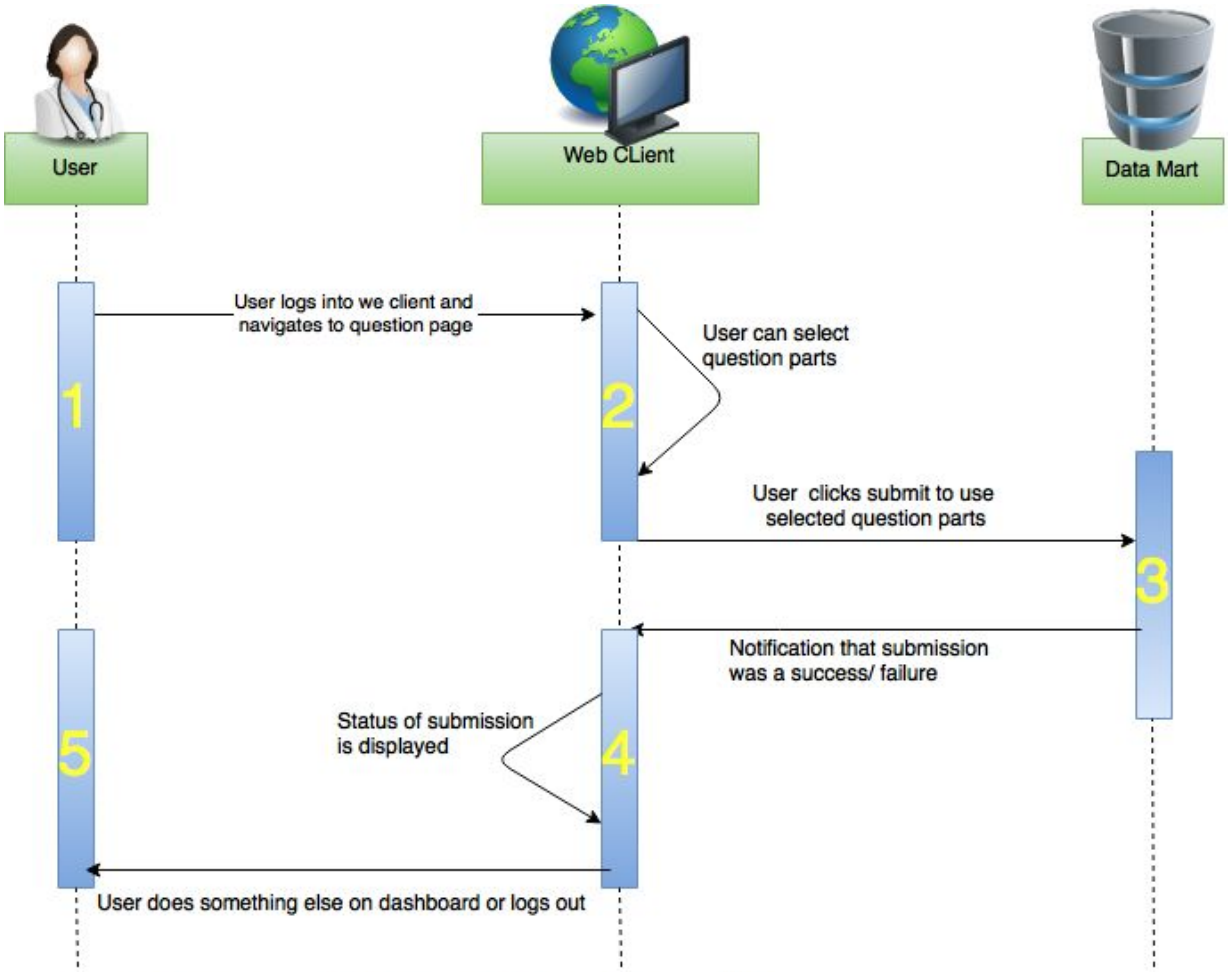

Figure 5: Sequence diagram for creating a question

1. User logs into web client using their credentials and navigates to the questions page of the user dashboard.
2. User then composes the the question they want to ask, along with any additional details
3. The user then clicks the submit button to send the question to the database
4. The database notifies the web client whether or not the submission was a success
5. The web client then notifies the user of the submission details

2.3.3: Sequence Diagram: User Gives Feedback to a Learner Prediction

This sequence diagram walks through the process of delivering feedback to a question that the learner has answered. When a learner finishes with a particular question, the users will be able to give feedback from the web application. When the user formulates his or her feedback, the information is stored in the data mart. This feedback shall be accessed later by the AI Controller. New or pre-existing instances of the learner shall be created based off of this information. Once the feedback has been submitted, the user is informed that the submission was successful, and the status of the question is updated accordingly.

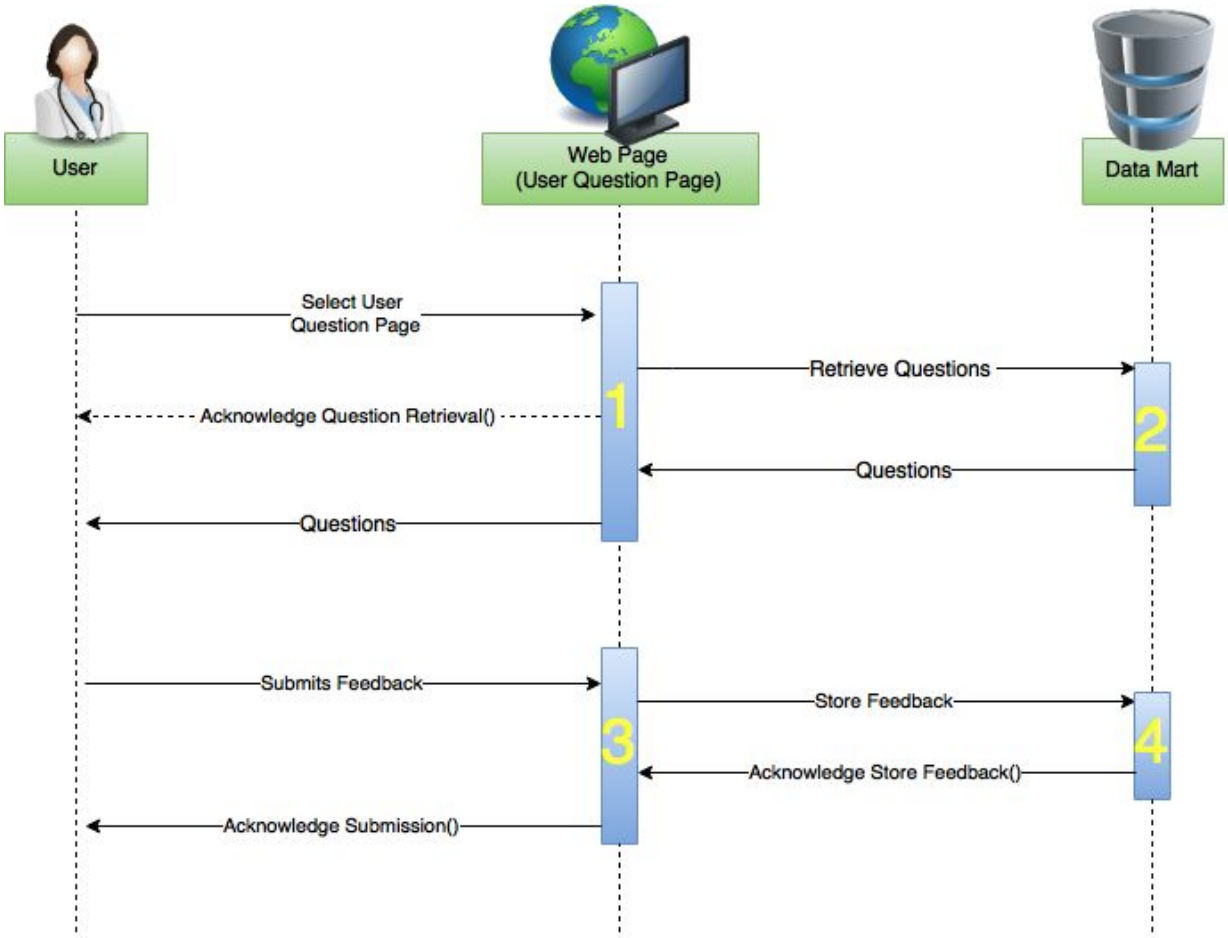

Figure 5: Sequence diagram for submitting feedback

1. User selects the Question Page from the menu screen. This begins loading the page. The web application then needs to populate the user's question table with questions.
2. The web application requests the data mart for the user's questions. Meanwhile, the webpage sends acknowledgement to the user to inform him or her that the application is retrieving questions. The data mart locates all of the questions posed by the user and returns them to the web application, and they are in turn displayed to the user.
3. User chooses a particular question and submits feedback for it. The feedback is then sent to the data mart.
4. The feedback is received by the data mart, which then sends acknowledgement back to the web application. Conformation of the submission is then shown to the user.

Chapter 3: User Interface Design

3.1: Introduction

The following cognitive stories are to help the developers understand how users interact with the system and correspond to the previous sequence diagrams. Each cognitive story gives detailed actions users take during interaction with the web client. The stories help ensure that the NEMO system is designed so that users will be able to know the correct actions to take to affect the system. The stories also help ensure proper provide feedback mechanisms are created so that users understand the actions they've taken.

3.2: Cognitive Story #1 - Creating Questions for use by the AI

The users are expected to have a firm grasp of the questions they are submitting. The users will be domain experts: they will be well experienced in their fields, with the knowledge and understanding needed to pose questions to the learning algorithms. The user is expected to have the discernment needed to notice potential trends resulting from specific medical situations.

The user is assumed to already be logged into a valid session on the web application. The user navigates to the question dashboard on the main menu screen. Once there, on the top of the page there will be a section labeled “Ask a question”. There shall be a drop down list of prefabricated questions to choose from. Once a base question is chosen, a drop down list shall be shown with possible parameters the user can choose. To the right of this drop down list there shall be a ‘+’ symbol which allows the user to add more parameters. Some parameters will have extra fields that need to be filled out. Once the user has fully formed their question, they should notice a green ‘submit’ button that shall be disabled until a user has at least one parameter for their question. After hitting the ‘submit’ button the user shall then see an indication on the screen stating that the question has been successfully added. If an error occurs, the details of the error shall be stated instead.

3.3: Cognitive Story #2 - User Gives Feedback to a Learner Prediction

The users are expected to have a firm grasp on the questions they are submitting. It is assumed that they already have a decent idea of the predictions that the learners are developing. The users will be domain experts: they will be well experienced in their fields, with the discernment needed to evaluate the performance of the learning algorithms. It is also assumed that the users will be delivering feedback for questions that they themselves have submitted, and not for any other user.

The user is assumed to already be logged into a valid session on the web application. The user navigates to the question page on the main menu screen. Once there, the webpage displays a user dashboard, which is populated with questions that the user has already asked. The user checks the status of his or her submitted questions and selects one that displays “Awaiting Feedback”. This reveals the AI’s response to the question. The user selects the “Give Feedback” option that is displayed on the AI entry. The user is first asked the question “Are you satisfied with the performance of the AI?” If the user sees that the accuracy of the learner is too low, he or she selects “No”. When this option is selected, the algorithm shall be ran again, in an attempt to improve its accuracy. After answering the first question, the user is given another. The user is asked “Do you agree with this prediction?” If the user feels that the learner’s prediction model is inaccurate, he or she may select “No”. When this happens, the user shall be asked a third and final question.The user is asked “What would you like to do with this learner?” Note that the user is only asked this question if he or she disagrees with the learner’s prediction. The user is given two options: “Reset learner” or “Delete entire question”. If the first option is selected, the learner the question shall be run again with different parameters or on a new algorithm. If the user selects the second option, the learner will be deleted and the question shall be removed from the data mart. When the user has answered all the feedback questions, he or she shall be prompted to submit the feedback. If the user is finished with the feedback, he or she can select “Submit”. This shall inform the user that the feedback has been successfully submitted. If the user made a mistake or wants to change a response, he or she can select “Undo” to undo the response. If the user selects this option, the question’s status shall remain “Awaiting Feedback”.

Chapter 4: Data Design

4.1: Data Description

The following two graphics are Entity Relationship Diagrams (ERD) describing the Data Mart. An ERD is a data model used to show the relationships between different entities for a specific domain.

The first image shows the NEMO specific section of the Data Mart. This includes any data needed for the web client and the AI controller to function.The question and parameter tables allow the AI to form queries for the I2B2 section of the Data Mart.

The second image shows the I2B2 section of the Data Mart. This contains medical data, which is a smaller subset of the source I2B2 database. This section of the Data Mart follows the genericized I2B2 standard.

The Data Mart is hosted on a SQL database server. The AI controller, AI instances, web client, and Data Loader all write and read from the Data Mart, sometimes simultaneously.

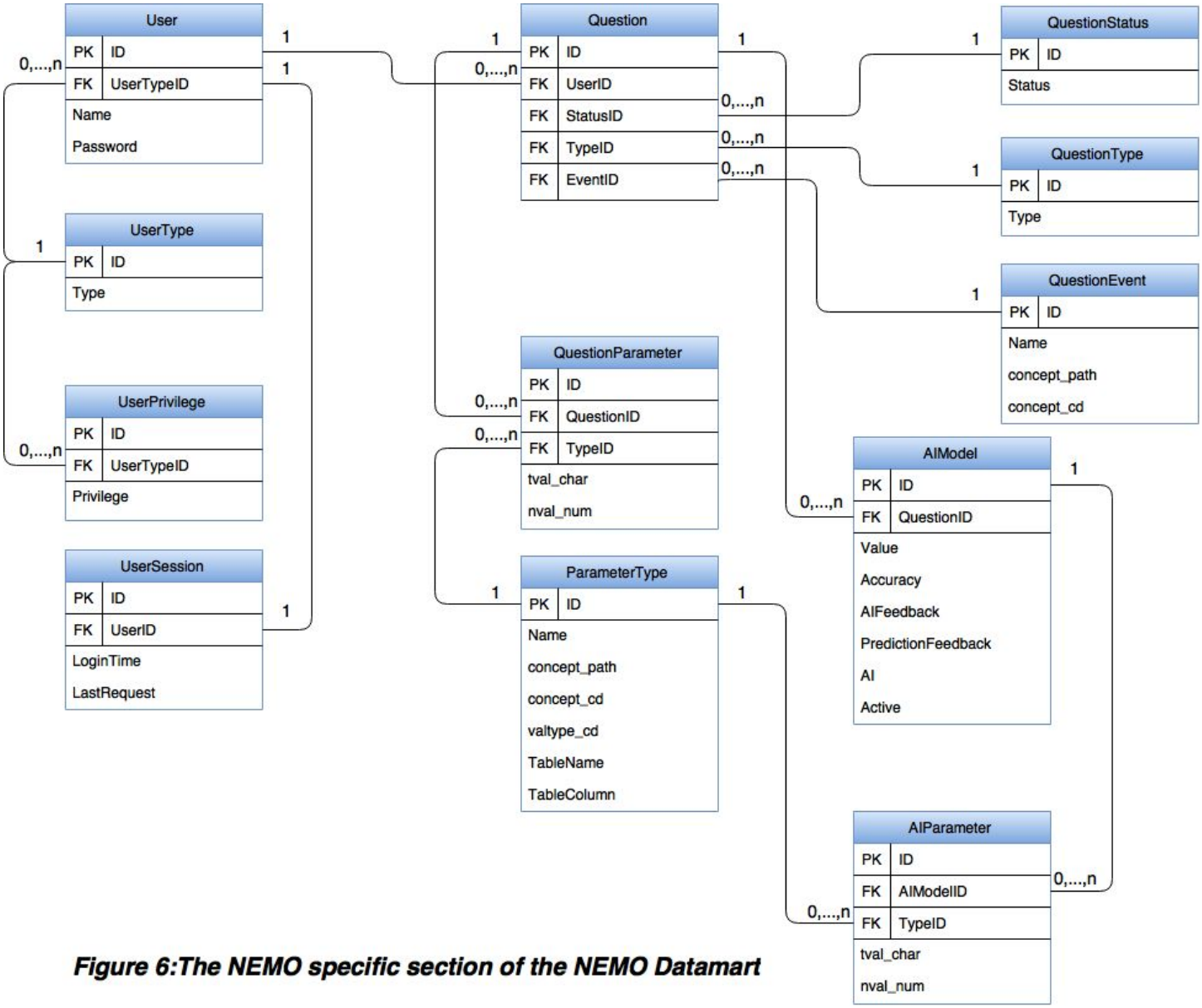

Figure 6:The NEMO specific section of the NEMO Datamart

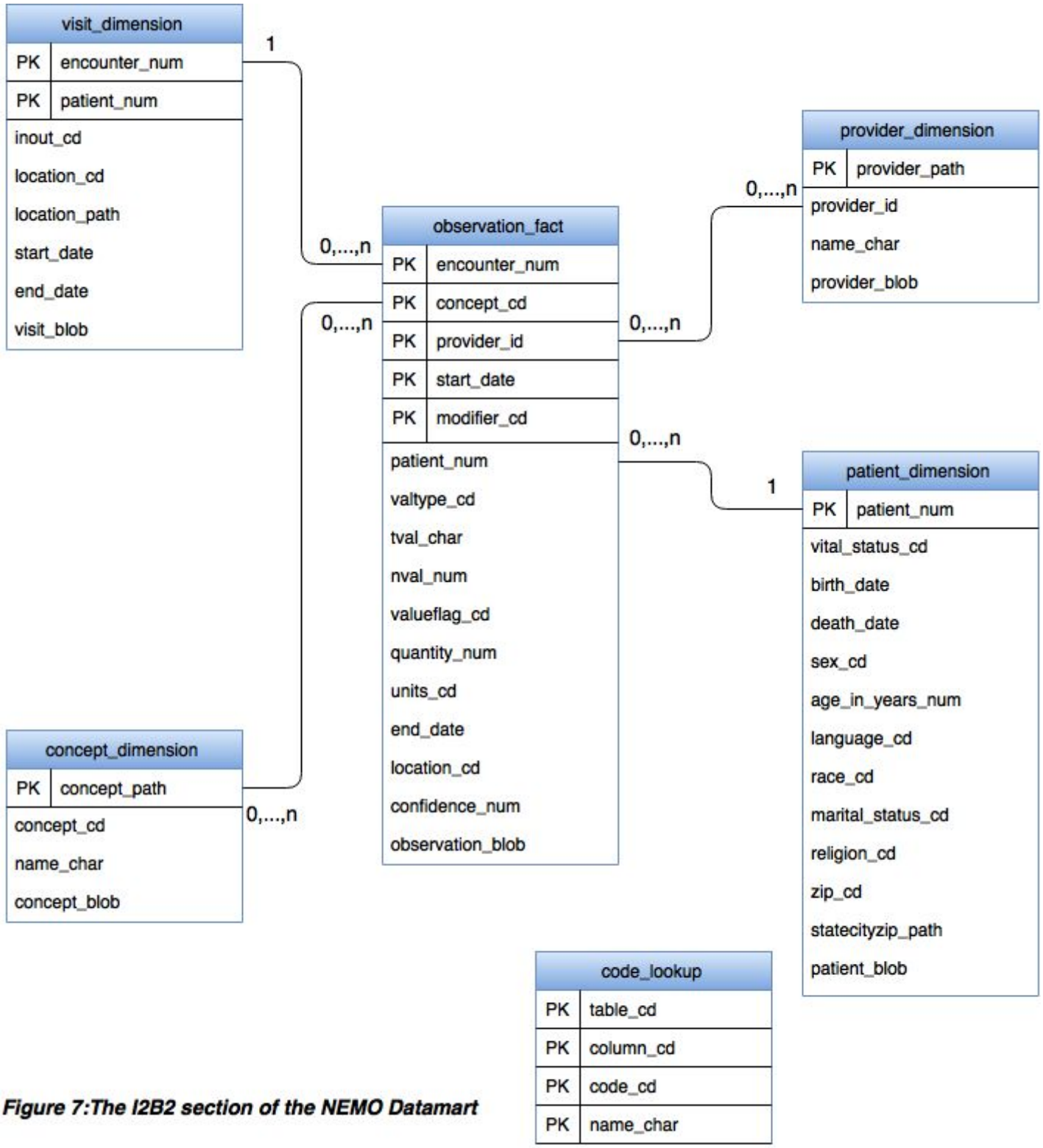

Figure 7:The I2B2 section of the NEMO Datamart

4.2: Data Dictionary

The following tables show a breakdown of each entity in the preceding Entity Relation Diagrams. Each attribute is described in terms of Name, Data Type, Description, Default Value, Uniqueness, and whether the attribute is required or not. These tables should be used as a guide to develop the SQL tables necessary for the NEMO Data Mart. Each row may exist as a column in the actual SQL tables.

4.2.1 NEMO Specific Section of the Data Mart

User					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify a user in the database	null	Yes	Yes
UserTypeID	GUID	The type of user, a foreign key to the user type table. This determines the user privileges available to the given user.	null	No	Yes
Email	VARCHAR	The email address of the user	null	Yes	Yes
Hash	VARCHAR	An encrypted password for the given user	null	No	Yes
First	VARCHAR	User's first name	null	No	Yes

Last	VARCHAR	User's last name	null	No	Yes
Affiliation	VARCHAR	Affiliation of the user	null	No	Yes
Confirmed	TINYINT	0 or null indicates the user is not yet confirmed. Non zero means the user is confirmed	null	No	Yes
ConfirmationHash	VARCHAR	Encrypted value to match against data sent by user during confirmation	null	No	No

UserType					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify a given user type in the database	null	Yes	Yes
Type	VARCHAR	The type of user. Certain types of users will have different privileges	null	Yes	Yes

MaxQuestions	INT	The max number of questions this user can ask	null	Yes	Yes

Question					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given Question	null	Yes	Yes
UserID	GUID	A foreign key relating this question to the user who owns it	null	No	Yes
StatusID	GUID	This foreign key describes the status of the Question	null	No	Yes
TypeID	GUID	A foreign key that describes what type of question this is	null	No	Yes
EventID	GUID	A foreign key that describes the event this question is trying to find, which could be a diagnoses, or hospital readmittance	null	No	Yes
DateModified	TIMESTAMP	The last time this row was edited or added	CURRENT TIMESTAMP	No	Yes
PatientJSON	VARCHAR	A JSON string defining a prediction patient	null	No	No
Classifier	VARCHAR	The user's choice for next classifier	null	No	No
Optimizer	VARCHAR	The user's choice for next optimizer	null	No	No
Prediction	VARCHAR	The prediction given by the classifier for the given patient	null	No	No

		json			
MakePrediction	TINYINT	0 specifies the AI Controller should not make a prediction for this question and the Patient JSON, 1 indicates it should	null	No	No

QuestionStatus					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given Question status	null	Yes	Yes
Status	VARCHAR	The name of the given status. It could be Queued, Awaiting Feedback, and so on...	null	Yes	Yes

QuestionType					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given Question type	null	Yes	Yes
Type	VARCHAR	The name of the given type. It could be “Chance of…”	null	Yes	Yes

QuestionEvent					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given Question event	null	Yes	Yes
Name	VARCHAR	User friendly description of the event for the web client	null	Yes	Yes
concept_path	VARCHAR	The path of the given concept in the I2B2 section of the data mart. For example, this could be 'Neurologic Disorders (320-389)\(346) Migraine\' to help find patients with migraines using the concept dimension	null	Yes	No
concept_cd	VARCHAR	The code describing the desired concept, if known. This can be used instead of concept_path to narrow patient searches		Yes	No

QuestionParameter					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given Question parameter	null	Yes	Yes
QuestionID	GUID	A foreign key that relates a parameter to a question	null	No	Yes
tval_char	VARCHAR	The value of the parameter, if the parameter type indicates text	null	No	No
nval_num	DECIMAL	The value of the parameter, if the type indicates it is numerical	null	No	No
concept_path	VARCHAR	The concept path for this parameter (such as in concept_dimension)	null	No	No
concept_cd	VARCHAR	The concept code for this parameter (as in concept_dimension)	null	No	No
valtype_cd	VARCHAR	Type of value, generally not used	null	No	No
TableName	VARCHAR	Used to specify if this parameter relates to a table besides observation_fact	null	No	No
TableColumn	VARCHAR	Used to specify which column this parameter relates to, if not in observation_fact	null	No	No

min	DECIMAL	Used for bounded parameters (such as age_in_years_num or LOINC parameters). The minimum bound.	null	No	No
max	DECIMAL	Used for bounded parameters (such as age_in_years_num or LOINC parameters). The maximum bound.	null	No	No

AIModel					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given AI Model	null	Yes	Yes
QuestionID	GUID	A foreign key that relates an AI Model to a question	null	Yes	Yes
Value	VARCHAR	The prediction or answer that the AI model has determined for it's question	null	No	No
Accuracy	DECIMAL	The accuracy rating of the AI model	null	No	No
AI Feedback	BIT	User feedback indicating if the AI is performing satisfactorily	null	No	No
Prediction Feedback	DECIMAL	User indication of whether the prediction was satisfactory	null	No	No
AI	BLOB	The serialized binary containing the AI model instance	null	No	No
Active	Bit	Determines if this model is still in use or deactivated	null	No	Yes
DateModified	TIMESTAMP	Tells the last time this row was added/updated	null	No	Yes
Algorithm	VARCHAR	The algorithm used for this AI model	null	No	No
Optimizer	VARCHAR	The optimizer used for this AI model	null	No	No

ConfusionMatrix	VARCHAR	The JSON string defining the confusion matrix created by this AI Model	null	No	No

AIModelParams					
Name	Type	Description	Default Value	Unique	Mandatory
ID	GUID	The primary key used to identify the given AI Model parameter. This is for the AI's suggested parameters	null	Yes	Yes
AIModel	GUID	A foreign key that relates a parameter to an AI Model	null	Yes	Yes
Param	VARCHAR	Defines the WEKA parameter used with this AI Model	null	No	Yes
Value	VARCHAR	The value of the parameter, if needed	null	No	No
param_use	VARCHAR	The optimizer used with this parameter	null	No	Yes

LearnerPatients					
Name	Type	Description	Default Value	Unique	Mandatory
patient_num	INT	The patient number of this patient used for AI learners	null	Yes	Yes

TestPatients					
Name	Type	Description	Default Value	Unique	Mandatory
patient_num	INT	The patient number of this patient used for AI testing	null	Yes	Yes

PatientReadmittance					
Name	Type	Description	Default Value	Unique	Mandatory
patient_num	INT	The patient number of this patient	null	Yes	Yes
readmitted	BIT	Whether this patient has been readmitted or not	null	No	Yes

4.2.2 I2B2 Specific Section of the Data Mart

observation_fact					
Name	Type	Description	Default Value	Unique	Mandatory

encounter_num	INT	Encoded i2b2 encounter number	null	No	Yes
concept_cd	VARCHAR	ID number for observation of interest (i.e. diagnoses, procedures, medications, lab test)	null	No	Yes
provider_id	VARCHAR	Practitioner id or provider id.	null	No	Yes
start_date	DATETIME	Starting date-time of observation	null	No	Yes
modifier_cd	CHAR	Ranking of Modifiers	null	No	Yes
patient_num	INT	Encoded i2b2 patient number	null	No	Yes
valtype_cd	CHAR	Format of the concept. N = Numeric, T = Text	null	No	No
tval_char	VARCHAR	The operator used when valtype_cd = ‘N´ E = Equals L = Less Than G = Greater Than	null	No	No
nval_num	DECIMAL	Stores a numerical value	null	No	No

valueflag_cd	CHAR	Used to flag certain outlying or abnormal values valueFlag_cd H = High L=Low A = Abnormal	null	No	No
quantity_num	DECIMAL	Quantity of nval	null	No	No
units_cd	VARCHAR	Units of measurement of nval	null	No	No
end_Date	DATETIME	The ending date-time for the observation	null	No	No
location_cd	TEXT	A location code, such as for a clinic	null	No	No
confidence_num	VARCHAR	Assessment of accuracy of data	null	No	No
observation_blob	TEXT	Holds any extra data that exists, often encrypted PHI	null	No	No

patient_dimension					
Name	Type	Description	Default Value	Unique	Mandatory
patient_num	INT	Encoded i2b2 patient number.	null	No	Yes
vital_status_cd	VARCHAR	Vital status code N = Living Y = Deceased, death date accurate to day M = Deceased, death date accurate to month X = Deceased, death date accurate to year	null	No	Yes
birth_date	DATETIME	Birth date of patient	null	No	No
death_date	DATETIME	Death date of patient	null	No	No
sex_cd	CHAR	Code for sex of patient	null	No	No
age_in_years_num	INT	Age of the patient, in years	null	No	No
language_cd	VARCHAR	Code for the primary language of the patient	null	No	No
race_cd	VARCHAR	Code for the race of the patient	null	No	No
marital_status_cd	VARCHAR	Code for the marital status of the patient	null	No	No
religion_cd	VARCHAR	Code for the religion of the patient	null	No	No
zip_cd	VARCHAR	Zip code of the patient	null	No	No
statecityzip_path	VARCHAR	Path specifying the state, city, and zip code of the patient's residence. Example: MA\BOSTON\02114	null	No	No
patient_blob	TEXT	Holds any extra data that exists, like encrypted PHI	null	No	No

visit_dimension					
Name	Type	Description	Default Value	Unique	Mandatory
encounter_num	INT	Encoded i2b2 encounter number.	null	No	Yes
patient_num	INT	Encoded i2b2 patient number.	null	No	Yes
inout _cd	VARCHAR	Code determining if this was an inpatient or outpatient visit	null	No	No
location_cd	VARCHAR	A location code, such as for a clinic	null	No	No
location_path	VARCHAR	The path that delineates location of the visit	null	No	No
start_date	DATETIME	Start time of the visit	null	No	No
end_date	DATETIME	End time of the visit	null	No	No
visit_blob	TEXT	Holds any extra data that exists, often encrypted PHI	null	No	No

concept_dimension					
Name	Type	Description	Default Value	Unique	Mandatory
concept_path	VARCHAR	The hierarchical path of the given concept	null	Yes	Yes
concept_cd	VARCHAR	The code that represents the diagnosis, procedure, or any other coded concept value.	null	Yes	Yes
name_char	VARCHAR	The name of the concept	null	Yes	Yes
concept_blob	TEXT	Holds any extra data that exists, often encrypted PHI	null	No	No

provider_dimension					
Name	Type	Description	Default Value	Unique	Mandatory
provider_path	VARCHAR	The path that describes the how the provider fits into the institutional hierarchy. Institution, department, provider name and a code may be included in the path	null	Yes	Yes
provider_id	VARCHAR	Practitioner id or provider id	null	Yes	Yes
name_char	VARCHAR	The name of the provider	null	Yes	Yes
provider_blob	TEXT	Holds any extra data that exists, often encrypted PHI	null	No	No

code_lookup					
Name	Type	Description	Default Value	Unique	Mandatory
table_cd	VARCHAR	The table for which this code applies. Example: VISIT_DIMENSION	null	No	Yes
column_cd	VARCHAR	The column where this code applies Example: LOCATION_CD	null	No	Yes
code_cd	VARCHAR	The given code Example: FH	null	No	Yes
name_char	VARCHAR	The actual name Example: Faulkner Hospital	null	No	Yes

Chapter 5: Notes, Issues, Assumptions

- When delivering feedback to the learners, the feedback will be stored within the data mart. The feedback will not be handled immediately by the AI controller. The controller will reference the feedback when spawning a new algorithm to handle the previous questions.
- It is assumed the Data Mart will be able to store serialized AI models as BLOBs, if this is not the case, another solution will need to be found. This solution could involve storing a file path to the serialized AI model on the server file system.
- Hospital readmittance may need to be defined as a concept or column in the I2B2 section of the NEMO Data Mart in order to use it as part of a possible query. The data loader may be able to calculate this if it is not given as part of the test data set.
- It is assumed the test data from KUMC follows I2B2 standards.
- The use case diagram (2.3.1) shows the possible actions a user with the highest privileges could do. These privileges could be different for other users e.g. someone who is learning in the field wouldn't be allowed to provide feedback for their own questions and someone else would be designated for this.