


Software Design - Area Type Prediction

Created by Oleksii Fedorenko, last modified on Jul 28, 2021

Template Last Major Edit: 3/14/2021 Added instructional text. Minor changes to layout and formatting.



System Level Design
The goal of the System/Sub-System design document is to provide an approved contract of interaction among all the involved modules in the system to enable meeting the requirements specified for this feature.

Technical Leads		
Roles	Names	Approvals
Platform Architect	n/a - Not needed for GUI	
Design Lead(s)	@ Grant Oesterling , @ Saurabh Panthri	Approved
Status		
Document Status	APPROVED	

- Overview / Background
- References
- Functional / Logical Element Decomposition
- Logical Interconnection / Internal Interface Design
- Behavioral Analysis (e.g. Sequence Diagram)
- External Interface Definition
- Physical Design & Deployment
- Test Scenarios:
- Rejected Solutions
- Other Notes / Resources

Overview / Background

When dealer changes the name of the Area, that arbitrary name is going to be used to predict the area type. The area type is going to be predicted using the ML algorithm that runs on the DS endpoint if the computer is online. Otherwise, the fuzzy match algorithm will be utilized. The source of the prediction is stored in the project db under the AreaTypeSource. ML algorithm is accessed through the API gateway.

APITool

Through the APITool, the connection to the remote ML algorithm is going to be made. It is expecting an area name as an input and returns the predicted area type list. The top prediction is stored in the database under AreaTypePredicted. The AreaType columns are identified with integer id's, which are changed into strings at a runtime. All of the columns are populated with Null i.e. undefined at the time of row creation and then populated by triggered events. The only event that will be implemented in phase 1 is on the area name update. In order to avoid the confusion with integer to string mapping. Each version of gui would use a single version of ML algorithm.

From Technical Approach - Room Type Prediction:

ProjectDB Updates

Create a new class AreaTypeDetails -> Not a DomainObject. Will just hold these 3 properties. Benefit is that all AreaType related code can be moved to that class. Area class will hold a reference for that class. Added table tblAreaTypeDetails, without updating the processor database at the moment.

Column Name	Type	Description
AssociatedAreald	Integer	Refers to the Areald AreaTypeDetails is associated with
AreaTypePredicted	Integer	This column will be used exclusively for the AreaType prediction feature, which is different from the existing column names AreaType, which is used in different settings. This stores the Area Type that was predicted, either by the ML algorithm or the fuzzy match algorithm. This is used to determine if the Area Type chosen by the user matches what the algorithm predicted.
AreaTypeSelected	Integer	This column will be used to identify which AreaType was actually selected by the user.
AreaTypeSource	tinyint	Stores the source of area type predicted. Can be fuzzy match or ML algorithm.

The AreaType columns are identified with integer id's, which are changed into strings at a runtime. All of the columns are populated with 0 i.e. undefined at the time of row creation and then populated by triggered events. The only event that will be implemented in phase 1 is on the area name update.

AreaTypes Enum

DomainObjects project will contain an enum to bind the int in AreaTypePredicted and AreaTypeSelected with the string. The enum is local to the gui and is not part of sqlite submodule. The enum for the integer to string mapping would be updated together with Gui versions and as new versions of ML algorithm are available. Since the model returns a string, the integer to string relation (which is stored as a table in the gui) should not change, but it can be expanded in the future gui updates. The list of strings that is used for the ML algorithm is going to be merged with IntelligentProgrammingRoomType list used for caseta. 0 is changed from Unsupported to Unset. The full list can be seen in



APITool

Through the APITool, the connection to the remote ML algorithm is going to be made. It is expecting an area name as an input and returns the predicted area type list. The top prediction is stored in the database under AreaTypePredicted. The AreaType columns are identified with integer id's, which are changed into strings at a runtime. All of the columns are populated with Null i.e. undefined at the time of row creation and then populated by triggered events. The only event that will be implemented in phase 1 is on the area name update. In order to avoid the confusion with integer to string mapping. Each version of gui would use a single version of ML algorithm, which share enum values.

FuzzyMatch

FuzzyMatch was originally implemented in Python. There are two packages for the types of FuzzyMatch, which are fuzzysset and fuzzywuzzy. When tested against the file below (~10,000 area names with known area types). FuzzySet gives ~80% accuracy while FuzzyWuzzy gives ~75%. FuzzyWuzzy is also a bit slower. However, I could not find any implementation for the fuzzysset algorithm on c#. Therefore, the package that I will probably use is [FuzzySharp](#), which is an implementation of FuzzyWuzzy on C#.

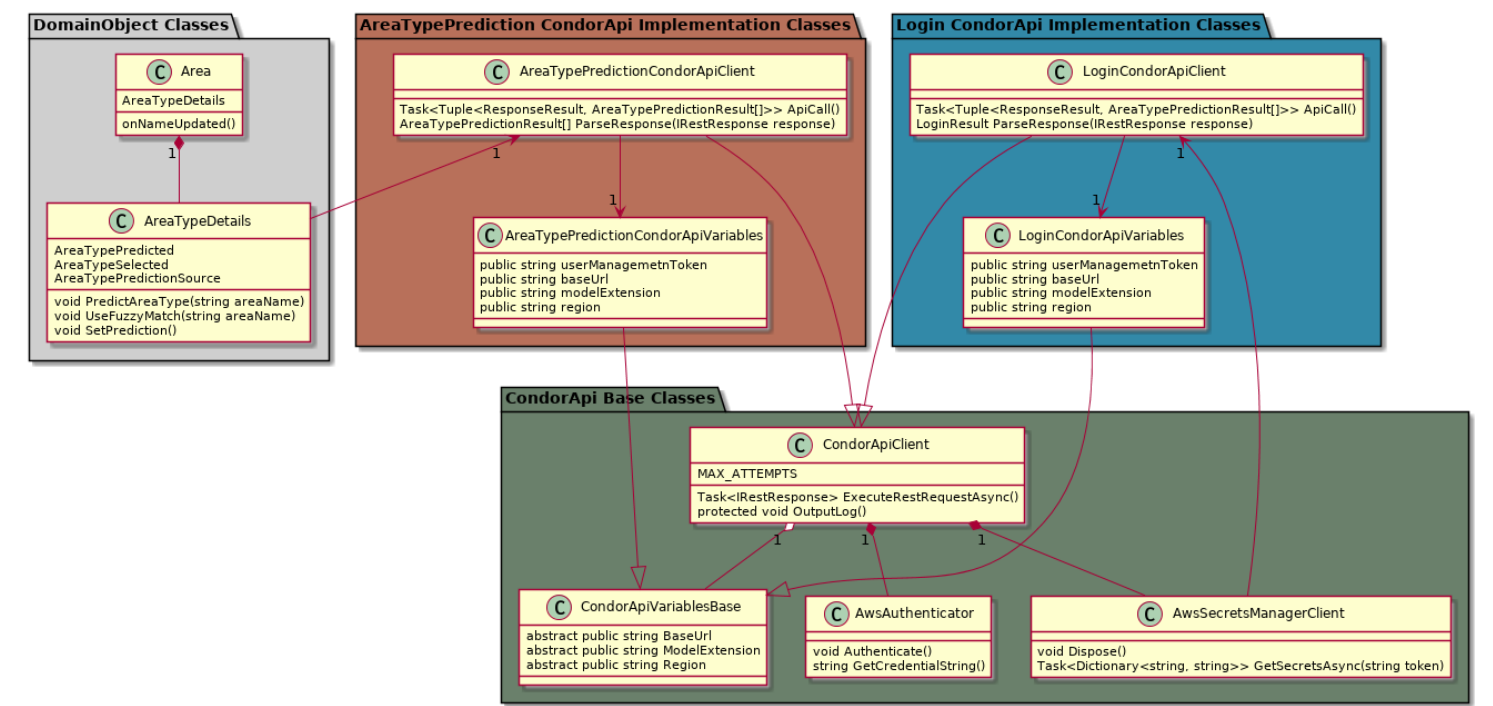
References

- [Requirements - Area Type Prediction](#)
- [System Design - Area Type Prediction](#)

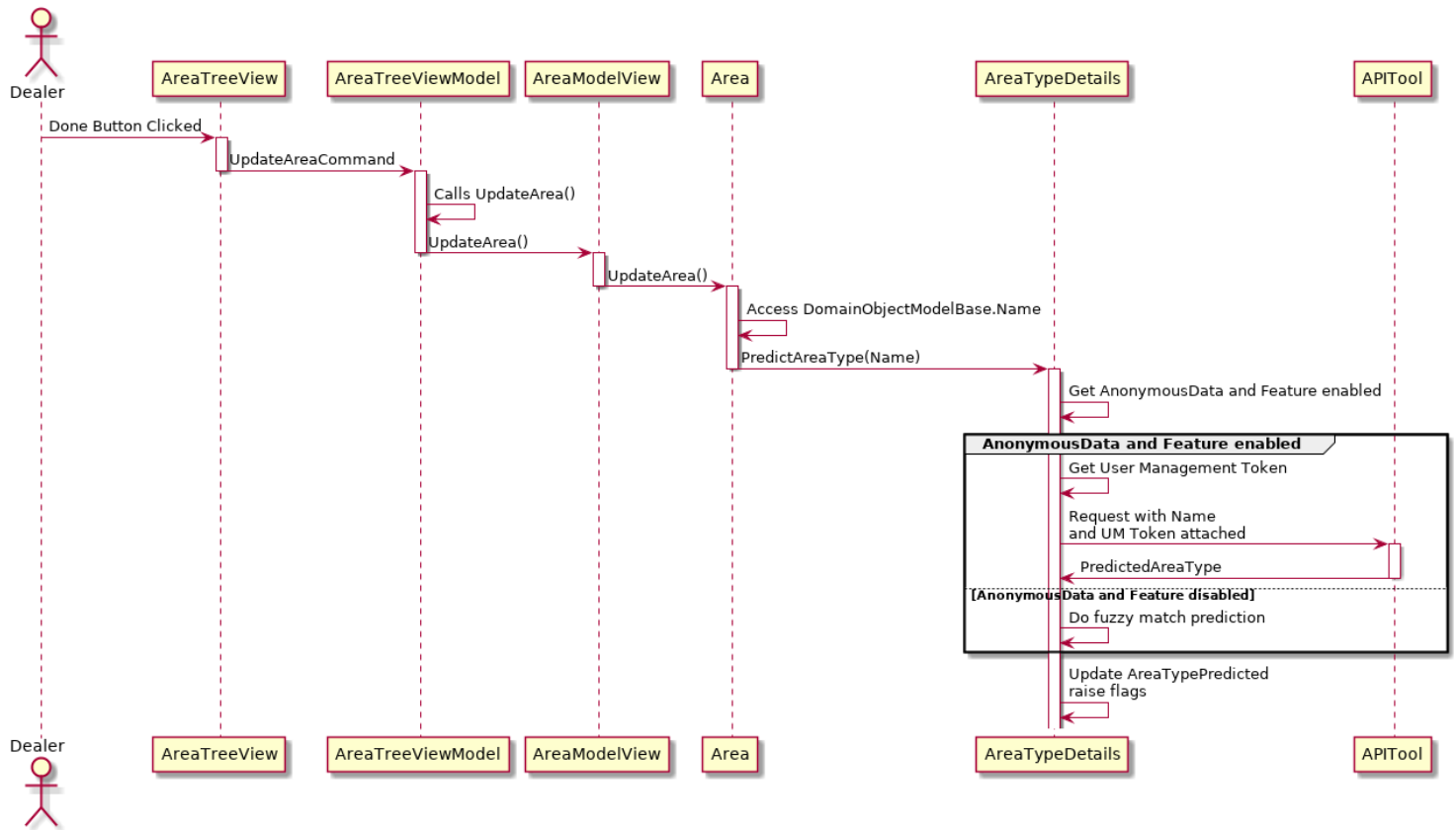
✔ See [2.1.4 Logical Design, Functional Decomposition and Interface Analysis](#) for guidelines and examples.

Functional / Logical Element Decomposition

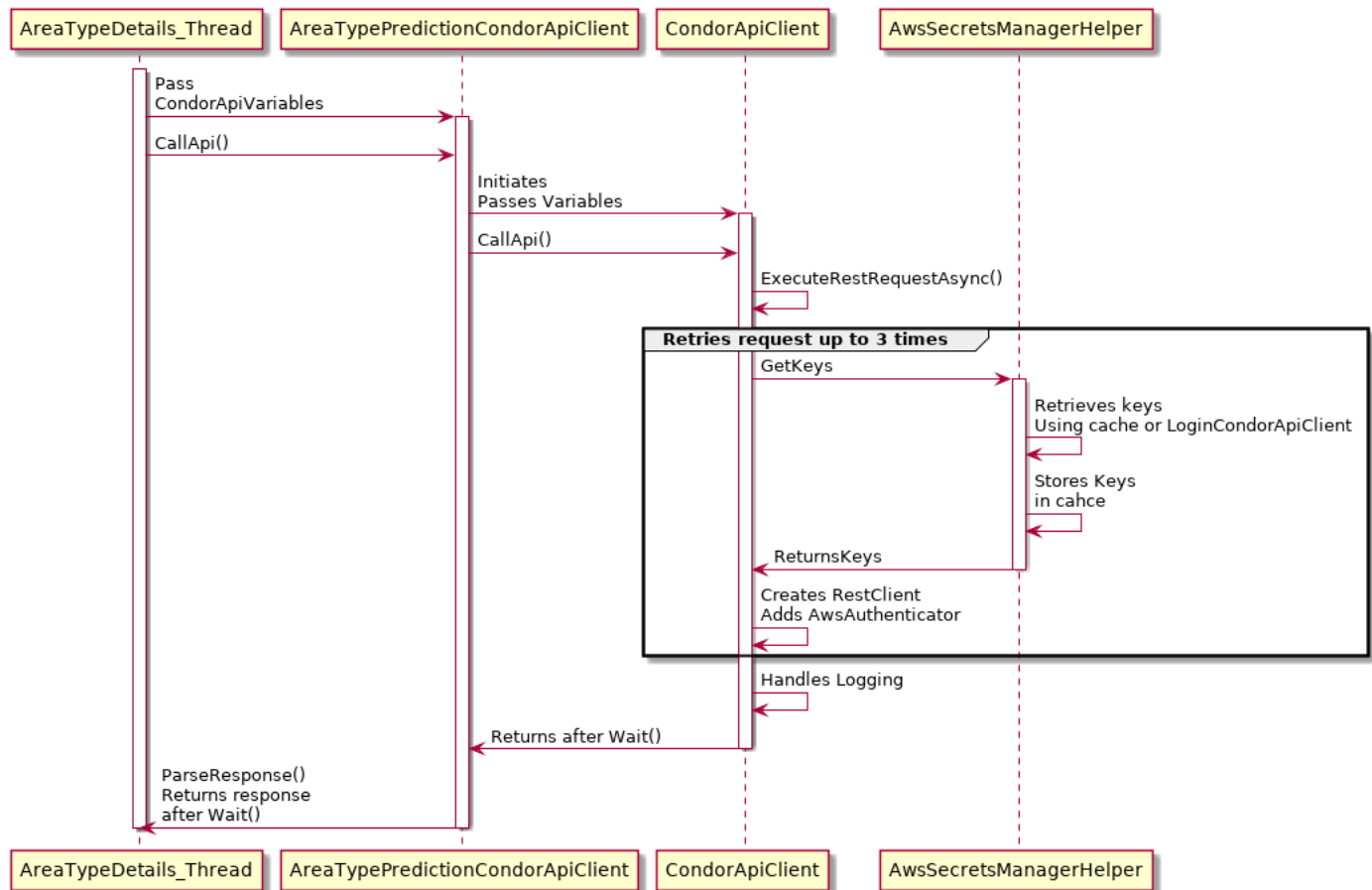
Logical Interconnection / Internal Interface Design



Behavioral Analysis (e.g. Sequence Diagram)



APITool Execution sequence



External Interface Definition

✓ See [2.1.5 Physical Design & Deployment](#) for guidelines and examples.

Physical Design & Deployment

Test Scenarios:

- Updates AreaTypePredicted only when area name is updated.
- Correctly reads AreaTypePredicted and AreaTypePredictionSource from the database.
- If online, executes CondorApi request, updates DomainObject and Database with the correct AreaTypePredicted and AreaTypePredictionSource.
- If offline, executes fuzzymatch, updates DomainObject and Database with the correct AreaTypePredicted and AreaTypePredictionSource.
- If request fails, falls back on fuzzymatch.

Rejected Solutions

Other Notes / Resources

Meeting Minutes

Create New Meeting Minutes

doc3

1 Comment



Oleksii Fedorenko

Where to put the CondorApiIntegration submodule?