**FIRST ->**

BACK END:

Do we need to know Java, Python? (probably only if we want to actually build the whole thing/if we want to use a NN in certain ways)

ML aspect:

1. Database <- **FIRST step in build**
2. Cuisine
   1. Pasta
3. Nutritional value

2. NN (for what?) (for turning the data into CB ontological data?) (NN is only interesting for certain functions like for instance COMBINING two domains [re: cuisines] into FUSION/using one domain as a eigenfunction of the other)

1. DL4j **<- SECOND step in build OR SKIP**
   1. How to use????

3. Crystal Ball

1. Graph Algorithms - Neo4j
   1. NEO4j Certifications **<- FIRST prereq**
2. Graph Embeddings - GAMMA integrations
   1. NEO4j Certifications
3. Graph Applications
   1. Graphileon integrations **<- SECOND/THIRD STEP**

**SECOND ->**

FRONT END: **<- THIRD/FOURTH STEP**

Crystal Ball GUI

1. Rhino + Grasshopper scripts?
2. Maybe we just draw it and make a single use case

Mimetic GUI - normal app stuff

\*means not building/just demo purpose

Prototype Options:

1. Do we just draw this out as a use case demo?
   1. Do we build the entire app \*in theory\* with pictures, too?
2. How much of the NN is actually implemented?
3. How much does CB actually work in this example?

Other stuff:

Can we patent the use of the 3D space to navigate Graph Database Algorithms?

Single use:

Cuisine Ontologies but only in the Country of Pasta and then in any Pasta neighborhood

Search parameters

In fridge

Not missing more than X ingredients

Fusion types

Nutritional Value/Dietary restrictions

Ability to make (appliances/skills)

Necessary components:

BACK END:

ML aspect:

1. Database
2. Cuisine
3. Nutritional value (extract food products database; nutritional value database?)
4. NN - DL4j
5. Crystal Ball (Graphs + NN)
6. Graph Algorithms - Neo4j
7. NEO4j Certifications

2. Graph Embeddings - GAMMA

1. NEO4j Certifications

3. Graph Applications

1. Graphileon

FRONT END:

Crystal Ball GUI

1. Rhino + Grasshopper scripts?
2. Maybe we just draw it and make a single use case

Mimetic GUI - normal app stuff

\*API Hooks

\*Recipe Search

\*Meal plans

\*Diet specific meal plans

\*allow people to load their own diet ontologies

**PROTOTYPE**

**BACK END:**

1. Dataset
2. NN - DL4j
3. Graph Algorithms - Neo4j
4. Graph database
   1. Use for importing dataset (ontologies) into visual format
      1. Visual replication in Rhino (with Real num abstracted labeling)

GAMMA

1. Graph embeddings
   1. Take Rhino representations and use them as Maps

Graphileon

1. Graph applications
   1. Additional feature construction

**FRONT END:**

Crystal Ball GUI

1. Rhino + Grasshopper Scripts
2. Objective: Display Informatihedrons and Prove that we can manipulate them in the way Crystal Ball hypothesizes
   1. Stages of transformation
      * 1. Going from general to subjective informatihedron (General informatihedrons = Cuisine Type, Dishes, pasta carbonara but this specific pasta carbonara [within the neighborhood related to this specific pasta carbonara etc. etc.] is a subjective informatihedron [within the neighborhood related to this specific pasta carbonara etc. etc.])

Mimetic GUI - normal app stuff

\*Neighborhood Search

\*Allow people to load, share, patent and publish their own informatihedrons

\*API Hooks

Difference between an informatihedron and a normal graph theoretical object is that an informatihedron contains not only all the properties of X but the spectral ranges of those properties themselves. In other words, all of the data is already connected.