Planning of the DNA Tool application

Main Components

- 1. Local Database to store obtained ancestry results (JSON file)
- 2. Application windows / screens
 - 1. Menu window (with access to all other windows)
 - 2. Database window

Allows to view the database, sort and search there. Provides addition / removal from the database. Provides uploads from other formats

- 3. Ancestry tool window / section
 Provides access to autosomal DNA comparisons and other tools based on local
 database and user input
- 3. Engine / Program to run data on (used by the Ancestry tool window)



Menu window

Database Ancestry

Database Menu window

View Database

Upload Data

Clean Database

Table of Database Contents

Database Window

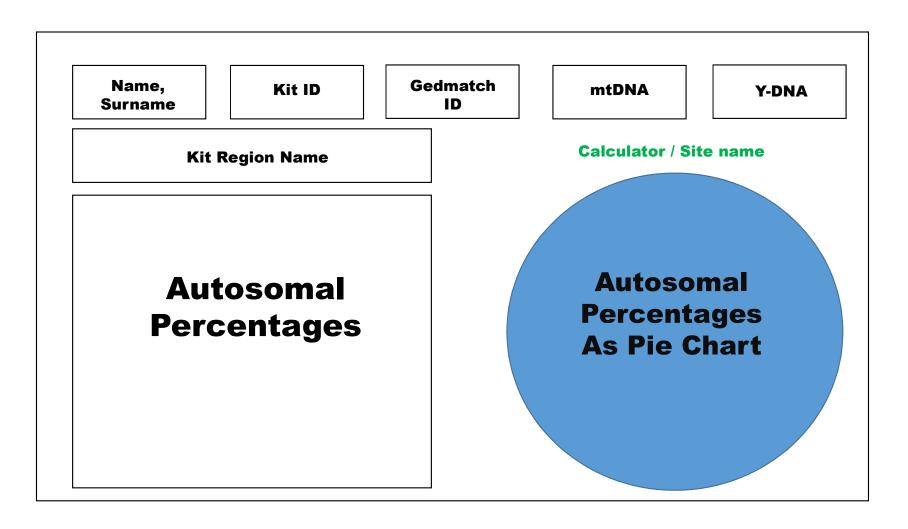
Each row is clickable / expandable



Database Add Window

Name, Surname (Alias)	One sample?
Gedmatch Kit ID	Enter Autosomal
Kit Region Name	Percentages
mtDNA	Not one sample (sample average)
Y-DNA	Select
ncient sample?	samples
Historical Time Period	

Sample Window



Ancestry Window

Autosomal Distance Autosomal Single mtDNA Y-DNA

Autosomal Distance and Autosomal Single Windows

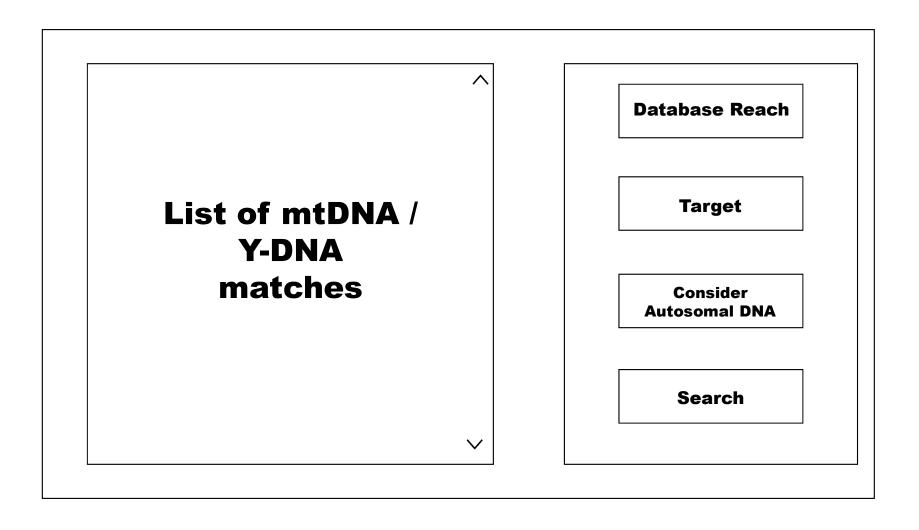
Database Reach List of Distance to Samples / **Target Single Calculator Type Closest Filter Samples on** Map Run

Location / Region of Matching Samples highlighted on the map

Map Window (as a Popup)



mtDNA and Y-DNA Windows



About Engine / Program

The engine /program will be called by Ancestry section mainly

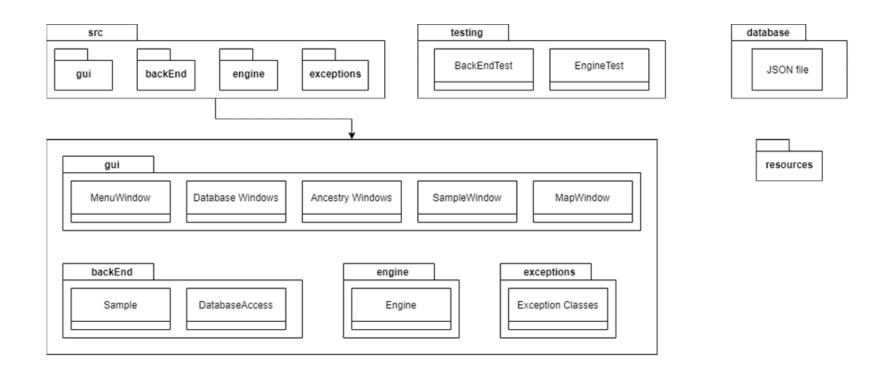
It should be able to provide a list of samples with closest distance to given target based on filters and calculator type

Similar idea should apply when calculating Single

The engine / program would also be used when searching for haplogroups considering the autosomal data

Planned UML diagrams for Java code

UML of packages



UML of back end

Sample SampleAverage name : String name : String id : String id: String gedmatchID : String type : String type : String regionName : String regionName : String components : ArrayList<Sample> percentages : JSONObject mtDNA : String percentages : JSONObject yDNA: String + ALL_AVG_SAMLES : ArrayList<SampleAverage> + ALL_SAMPLES : ArrayList<Sample> + SampleAverage(name : String, regionName : String, <-----Use----components : ArrayList<Sample>) + Sample(name : String, gedmatchID : String, type : String, percentages : JSONObject) generateID(): String + Sample(name : String, gedmatchID : String, + calculatePercentage() : JSONObject type: String, regionName: String, + removeComponent(component : Sample) : void percentages : JSONObject, + addComponent(component : Sample) : void mtDNA: String, yDNA: String) + remove(): void generateID(): String + getter methods + setter methods (except "id", "percentages") + remove() : void + toString(): String + getter methods + setter methods (except "id") + toString(): String Extends SampleAncient timePeriod : String + SampleAncient(base, timePeriod : String) generateID(): String + getTimePeriod() : String + setTimePeriod(timePeriod : String) : void + toString(): String

DatabaseAccess

- + loadDB(): void
- + addToDB(sample : Sample) : void
- + addToDB(sampleAvg : SampleAverage) : void
- + removeFromDB(sample : Sample) : void
- + removeFromDB(sampleAvg : SampleAverage) : void