Class and Interface Approach

As we mentioned in the Requirements Analysis document, the four main components in our application are: the crawler, the normalizer, the aggregator, and the anomaly detector. Therefore, we are proposing the following approach, regarding the classes, in order to build an application for the required task.

The Crawler Component

The crawler component will be built in two parts:

NameCrawler

-TBD

DeclarationCrawler

With every politician name collected by the first crawler, we can search for every income declaration

Classes:

- **Parser** - a class designed to give the program a command line approach, with the option of inserting the politician's first name, last name, functions, declaration type, region and number of declarations to be downloaded.
- **Crawler** - a class designed to crawl the information after the given parameters. We use Selenium for this task, dividing it into three functions: one to provide the details given by the input, one to search the politicians by these details and one to save their declarations as we mentioned in the parameters.

The Normalizer Component

The component with the role of transforming the data into a usable format. In order to define this format, we created two models, based on the entities we need to use in the further layers.

Classes:

- **PoliticianModel** - an entity class with the identification properties of a politician(first name, last name etc.) which will be used by the classifiers.

- **SearchModel** - an entity class which will be used between the crawlers. The NameCrawler will communicate with the DeclarationCrawler using SearchModel objects which describe the first and last names of the politician to be crawled.

- **DeclarationType** - an enum used to describe which declaration are we using to crawl/predict with

- **NormalizerInterface** - a class meant to simulate a normalizer interface, which will contain a function to normalize data

- **CSVNormalizer** and **PDFNormalizer** - two classes designed to transform the csv/pdf files into the text data which later on will be converted in aforementioned models.

- **Convertor** - interface-role class, meant to convert data from the text formats into the models.

- **PoliticianConvertor** and **SearchModelConvertor** - classes which inherit the Convertor class, each meant to obtain their assigned model objects

The Aggregator Component

The component meant to transform all the usable data into a specific format used by the Anomaly Detector.

Classes:

- **ContractionFixer** - a class used for removing all the contractions from the given text

- **WordTokenizer -** a class used in the process of segmenting running text into words. In essence, it's the task of cutting a text into pieces called *tokens.*

- **PunctuationRemover -** a class used to get rid of punctuation in order to get a more homogeneous structure of the text.

- **Lemmatizer** - a class used to convert every word to its canonical dictionary form (ex. 'connection' => connect).

- **POSTagger** - a class used to make a dictionary with the words from the data as keys and their parts of speech as the values.

The Anomaly Detector Component

After we gather the data, we will split the dataset into train and test, in order to prepare the model to predict. Because we can make an assumption starting with the comparison between the wage of a politician and his income declaration, initially we came up with two approaches (which will be changed/adapted as we further continue with the implementation).

Classes:

- **KMeansApproach** - a class which is designed for a clustering based approach based on the salaries, we attempt to classify each salary based on value. We could potentially add many more labels and cluster based on them, or we could use the present approach as a guideline for versions of the Anomaly Detector to come

- **KnnApproach** - a class which is designed for a classification problem based approach - based on the salaries and another label - that, in a real scenario, may be represented by a lot of things - if the person has been associated with known corrupted individuals, if he/she was, and many more