USE CASE + ACCEPTANCE CRITERIA

Use case 1	**UI**
Brief description	The user enters the sentence to be analyzed and ticks the box to show the syntactic tree.
actor	User
Special Requirements	The user input must be valid, if it's not-valid (blanks, non-existent words, punctuation) the system requires a new input.
Flow of events	User launches the application → UI opens → User enters the input sentence → check for non-valid input → provide output (syntactic tree, output sentence, toxicity level, Results.txt)
Pre-condition	Valid input
Post-condition	The user sees the results and the syntactic tree

Acceptance Criteria:

- **Given** the user accesses the application
- **When** they type a sentence into the input field
- **Then** the sentence is saved and becomes available for analysis

Use case 2	**Box**
Brief description	The user can tick the box to see the syntactic tree
actor	user
Special Requirements	The UI must be lunched correctly
Flow of events	User ticks the box → UI submits the request of syntactic tree to App → App generates the syntactic tree using Google API → UI shows the syntactic tree
Pre-condition	The UI is lunched
Post-condition	If the box is ticked, UI shows the syntactic tree.

Acceptance Criteria:

- **Given** UI lunched
- **When** the user ticks the box
- **Then** UI shows the syntactic tree

Use case 3	**Analyze syntax**
Brief description	The App access the Google API (Analyzing Syntax) to analyze correspondingly the input sentence and the output sentence
actor	App
Special Requirements	The user must have working Google API credentials
Flow of events	User enters a valid input sentence → Google API extracts phrases from the input sentence → The phrases are sent to App
Pre-condition	The user must have working Google API credentials

Post-condition	The input sentence is split in phrases

Acceptance Criteria:

- **Given** a valid input sentence
- **When** the app sends a request to the appropriate Google API
- **Then** the API returns analysis results, which are processed by the app

Use case 4	**Toxicity**
Brief description	The app sends the output sentence to Google's Moderate Text API to determine the level of toxicity.
Actor	App
Special Requirements	The user must have working Google API's credentials.
Flow of events	App generates an output sentence → Google API calculates the toxicity level (using Moderate Text) → returns value to App → UI shows results
Pre-condition	App generates an output sentence
Post-condition	App receives the toxicity level

Acceptance Criteria:

- **Given** an output sentence
- **When** the sentence is ready
- **Then** the app sends it to Google's Moderate Text API
- **And** receives the toxicity level

Use case 5	**Word List**
Brief description	The application loads lists of nouns, adjectives, verbs, adverbs, articles, and pronouns from their corresponding .txt files located in the resources folder. These lists are then used to generate nonsense sentences
actor	App
Special Requirements	Each file.txt must exist and contain at least one valid word per category. Each file.txt must be correctly loaded in the source code
Flow of events	The app interacts with WordList --> WordList accesses the resources directory -->the App creates a list of Nouns,verbs,... to put in the sentence structure for the output sentence
Pre-condition	
Post-condition	The application holds in memory categorized lists of words ready for use in generating output

Acceptance Criteria:

- **Given** the .txt files contain lists of words (nouns, verbs,...)
- **When** WordList gives the base of Noun,Verb,.. object
- **Then** App creates a list of Nouns,Verbs,...
- **AND** loads it in the sentence structures chosen

Use case 6	**App**
Brief description	It’s the main core of the system, it deals with the basic logic that parses the input sentence and provides an output to UI
actor	App
Special Requirements	The user must insert his correct and valid API Google Cloud’s credentials in the correct file (credentials.json) so that the App can have access to the Google’s API
Flow of events	Input sentence → analyze sentence with google API → randomly select words and sentence structure → create output
Pre-condition	Correct credentials
Post-condition	The UI receives the generated output

Acceptance Criteria:

- **Given** the valid sentence and valid Google API credentials
- **When** sent to the Google Cloud Natural Language API for syntactic analysis
- **Then** provides the output to the UI

Use case 7	**Resources**
Brief description	The app randomly selects: nouns, adjectives, verbs, adverbs, articles, pronouns, sentence structures; from correspondingly: Nouns.txt, Adjectives.txt, Verbs.txt, adverbs.txt, Articles.txt, Pronouns.txt, Sentence Structures.txt.
actor	WordUtil
Special Requirements	The files must be not empty
Pre-condition	The App needs to loads the sentence structure template with the selected list of noun,verbs,…
Post-condition	The output sentence template is loaded with all the words needed

Acceptance Criteria:

- **Given** the app needs to generate new content
- **When** it accesses the word libraries
- **Then** it randomly selects elements from the corresponding files for sentence construction

Use case 9	**User**
Brief description	The user lunches the app and enters an input sentence and if he wants ticks the box for the syntactic tree. Then the system provides the output
actor	user
Special Requirements	The user must be able to lunch the app
Pre-condition	The User sentence is entered and analyzed only if valid
Post-condition	The output is printed

Acceptance Criteria:

- **Given** Valid Input Sentence
- **When** The App is lunched
- **Then** the output is shown to the user