**Diagrama de clase conceptuală**

Proiectul este modelat in așa fel incât să respecte regulile SOLID ale design-ului OOP. Pentru această secțiune cele mai importante reguli le constitue:

1. Single responsibility – fiecare clasă trebuie să aibă o singura responsabilitate. Acest principiu poate fi extins să includă si metodele fiecărei clase.
2. Open closed – o aplicație trebuie să fie deschisă spre a fi extinsă insâ inchisă schimbării. Cu alte cuvinte, pe baza codului deja existent in aplicație, noi funcționalități pot fi adăugate cu ușurință.
3. Dependency Inversion – clasele să depindă de abstracții.

Pentru a respecta aceste principii am ales ca proiectul să fie multi-modular. Fiecare modul in parte are o singură responsabilitate bine definită, este inchis schimbării si deschis extinderii și depinde de abstracții. Ca și tehnologie am ales ca proiectul să fie de tip Maven. Maven este un software de management și ințelegere al proiectului . Este bazat pe conceptul de proiect văzut ca și un model de obiect (POM). Maven poate administra build-ul unui proiect, genera diferite rapoarte ale sale precum si documentația dintr-o singură locație de unde iși ia informația.

De asemenea, Maven suportă și un proiect multi-modular. Printre modulele proiectului se enumeră:

1. Core

Modulul core conține, după cum îi și zice și numele, nucleul aplicației și anume algoritmii fundamentali care constau in manipularea textului. Toți algoritmii specifici operatiilor aplicabile unui text sunt implementați in acest modul.

1. Model

Modulul model conține toate clasele ce modelează lumea naturală. Spre exemplu, acest modul conține o clasă User ce modelează un utilizator al acestei aplicații. Datorită naturii acestui modul, toate celelalte parți componente ale aplicației depind de el.

1. Database

Modulul database conține logica de conectare si manipulare a bazei de date. Deși aplicația in sine nu folosește baza de date in mod intensiv, am decis ca existența acestui modul este necesară tocmai pentru a respecta regulile de bază a design-ului OOP.

1. Service

Modulul service reprezintă legatura dintre modulul UI si modulele de core, database si live coding. Acesta inglobează algoritmii implementați de modulul core, conectarea la baza de date și live coding-ul in clase de servicii ușor de folosit.

1. UI

Modulul de UI reprezintă interfața grafică a aplicației. Ca și tehnologie este folosit Javafx, o librărie grafică foarte puternică și flexibilă.

1. Live coding

Modulul de live coding oferă posibilitatea utilizatorului de a coda in timp real cu unul sau mai mulți prieteni. Pentru a realiza acest lucru, modulul conține un server propriu la care se pot conecta clienții, iar pentru păstrarea corectitudinii operațiilor ce se pot aplica asupra unui text au fost implementați algoritmi de tip transformări operaționale.

Diagrama modulelor si dependințele acestora este următoarea: