

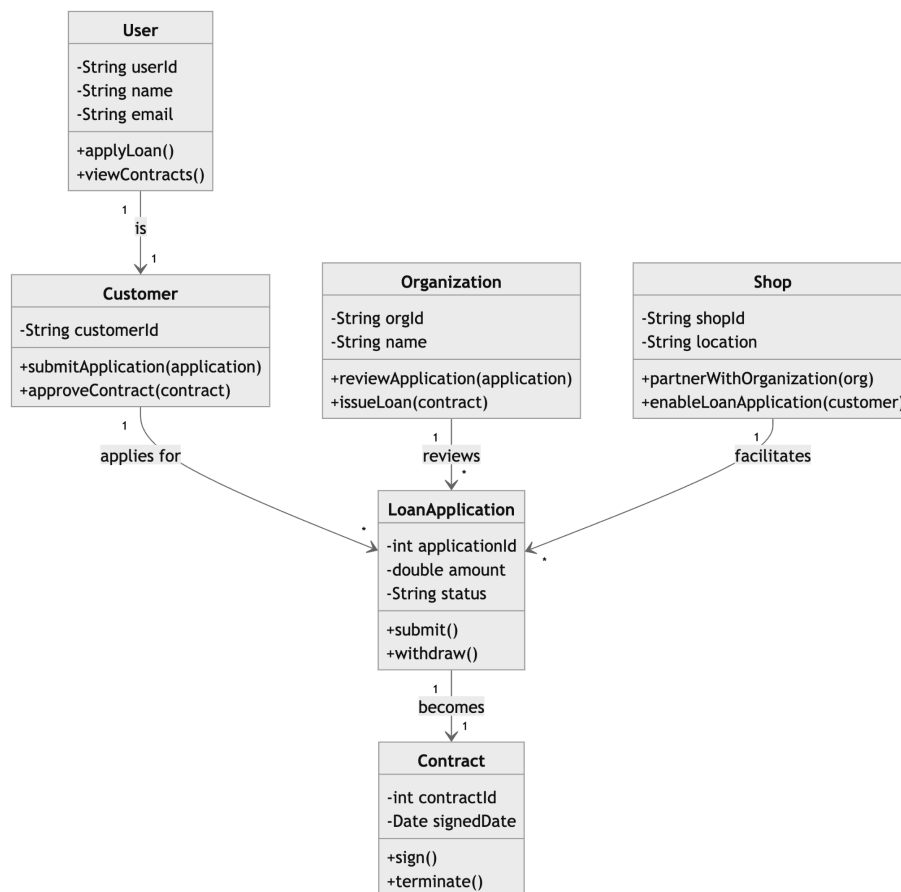
Task 5.

The application we have used for generating the UML class diagram is <https://diagrammingai.com>. It is a GPT-based GenAI tool, which can create various types of diagrams, including but not limited to flowcharts, sequence, entity relationship, class diagrams, given a prompt or an image.

We have chosen DiagrammingAI, for its specialized capabilities in creating the diagrams, and its ability to iteratively refine diagrams through prompts.

The prompt used for generating the diagram is as follows - *The system facilitates a small loan process, allowing users to apply for loans between 500 to 15,000 EUR, sign contracts, and receive funds in their bank accounts. Loan decisions are nearly instantaneous for most applicants. It manages various user types, including customers and organization representatives, to streamline loan processing. Key entities include User, Customer, Loan Application, Contract, Organization (loan provider), and Shop (partner store for loan applications). Customers can apply for multiple loans through this system. A loan application signed both by the Organization and the customer becomes a Contract.*

Below is the output:



Considering that the GenAI tool was provided with the limited description, the result can be counted as satisfactory. Although the number of attributes is insufficient (which can be justified

by the lack of context), the ones present were aligned with the diagram we created. The associations between different classes, and their multiplicities were also chosen correctly. Furthermore, DiagrammingAI identified the dependencies between classes, such as “Customer applies for a Loan Application”, “Organization reviews LoanApplication”.

The generated diagram effectively depicts the intended behaviour of the classes by providing the operations. While the majority of operations DiagrammingAI included, are relevant for our system, a few operations important for the business logic were omitted (e.g choosing identification method for the Customer, or updating the status for the LoanApplication, etc).

All in all, I believe that this tool can be convenient for developing an initial sketch of a class diagram. Its strengths lie in identifying the basic operations, and relationships between classes based on a brief description. However, due to the lack of context, it can miss crucial domain-specific features, which can be considered a weakness. By iterating and constantly refining the prompt with details, the diagram provided by the DiagrammingAI can become increasingly similar to the actual diagram intended for the system.

Although we did not update our model with the details provided by the tool, it was useful for validating that our model aligns with the standard conventions for UML class diagrams.