

Task #3

Our first choice for the current task of creating a class model was Figma. It is a cloud-based design tool mostly used for UI/UX design, but in our case, it also suits nicely to effectively design class diagrams due to its flexibility. Its interface and familiarity with our team made this tool a relatively easy choice for creating a simple class diagram. Since there are literally no restrictions to designing things in it, it is easy to visually represent class structures and relationships with Figma's extensive library of shapes, and connectors. As it is in the cloud, our team can work simultaneously on one project or diagram in an online workspace, which is a great advantage of using this tool. The ability to export designs in various formats also ensures good integration with other tools during the development process.

Our choice for future diagram modelling will be Visual Paradigm, which is as they say "*the market #1 visual modeling and diagramming platform*" designed to help create various types of diagrams. An important moment for our team in terms of choosing a tool for this matter was the presence of a free trial or community version in the case of Visual Paradigm, which makes it suitable for all kinds of users, especially students. This tool is also an online platform, which provides flexibility in working on projects, if you connect to Visual Paradigm Online, it reveals a lot of new team tools to conveniently manage the work. Your projects and diagrams can be saved to an online workspace. Diagrams specifically can be edited from both online and desktop versions. Additionally, the tool includes a version control system, allowing users to track changes, revert to previous versions, and manage the progress of the diagram. Users can upload work through commits (like in GitHub), however, in Community Edition only 5 commits per day are allowed.

Task #3 (additional point)

Regarding Figma, unfortunately, it doesn't provide the ability to generate code from a class diagram, since it is not specifically a tool to design these structures, but rather generally for UI/UX design.

As for Visual Paradigm, has a very nice feature of generating not only application code from diagrams but also the code for databases. It supports various programming languages, such as Java and C#. This tool has great support for Java persistent code, including static methods,

factory classes, Data Access Objects (DAO), Plain Old Java Objects (POJO), or even just the mapping between objects and database entities.

However, we have not yet tried this feature of Visual Paradigm, since application class mode is yet to be created.