**Technologies used**

**Modeling:**

For the modeling part of the development, StarUML was used. StarUML is an open-source software modeling tool aimed to support agile and concise modeling. Since the tool supports most of the diagram types specified in UML 2.0, it made the modeling phase of the development much smoother.

**IDE:**

We aimed for developing a desktop application, making use of tools developed my Microsoft. One of the first decisions made regarding implementation was about the IDE. We decided to use Visual Studio. This IDE enables easy application development for all platforms, in any language, especially for platforms and languages also developed by Microsoft, like the ones we chose. Also, the extensions provided by Visual Studio saved considerable amounts of time when it came to writing code.

**Programming language, UI:**

In terms of programming languages, C# was chosen, because it is a widely-used general-purpose object-oriented language, commonly used for developing desktop applications. In order to keep consistent with the family of technologies used, the UI was made in the Windows Presentation Foundation framework. WPF is a UI framework used to create desktop client applications, and it is a subset of the .NET framework. From one hand, WPF proved to be a good choice from the chosen programming language’s point of view, from the other hand WPF supports a broad set of application development feature that we made good use of during development.

**ORM:**

As Object-Relational-Mapper, we chose to use Entity Framework 7. This framework provides two approaches to create the context and entities of the application in a consistent manner. We used the database first approach. By doing so, we were able to flexibly modify the entities of the application as well on the code, as well on the database level, which helped in keeping the code clean and transparent.

**Data persistency:**

For our data storage we used a service called GearHost, which provides a variety of hosting services, including web hosting for MSSQL. The main reason that led us to host our database on a remote server was that it provided a way of accessing our data in a consistent manner which proved to be especially useful during our application development. As such, GearHost was a rather straightforward choice as its services were completely free to use and reliable.

**Version Control:**

The choice of a version control system was one of the first decision which had to be made to enable team collaboration and independent development of the project. We have decided at the beginning to use a private GitHub repository for this purpose as it allowed each member to make their contributions accessible to the other team members and it made available a clean history of the changes and additions. The reason we have chosen GitHub is that it had a clean and easy to use interface on both web and desktop platforms, and that it had the possibility of working on a private repository which ensured that only the developers had access to the development process.

**Task management:**

The other important aspect of the team collaboration lied within how the individual tasks of the members had been managed and kept track of. Trello has proven itself as being a very powerful tool for this purpose, as it made it very easy and straightforward to attribute tasks and to manage the existing ones based on their current status, so that at any point all team members were aware of their current responsibilities and deadlines.