Technical Documentation

**Languages:**

1. C#
2. HTML5
3. JavaScript
4. MySQL
5. CSS

**Reused Programs and algorithms:**

1. -Store Project from Matthew’s previous work, now rewritten for C# and Discord.
2. -Discord API from Discord
3. -C# Wrapper from Discord – DsharpPlus by Naamaloos
4. -testing components for commands from Dsharpplus.Test
5. -NadekoBot - Examples in layering services
6. -DSharpPlus.Test for examples in using Dependency Injections
7. -MusicBot by Chrisradio and Nadeko.MusicPlayer for music module
8. Authentication uses passport-discord based on Facebook OAuth2
9. Database sync changes are based on repository pattern from examples from NadekoBot

Software Tools and Environments:

1. Visual Studios: used in the core engine. It’s an instanced class we can call from.
2. .Net 4.7: used with the wrapper to make it work. .Net 4.7 is used to get access to AspCore and EntityFrameworkCore frameworks and have access to newly developed and updated libraries that are only available on .Net 4.7
3. Visual Studio 2017 Community used for compiling and writing majority of the bot. Usage of Intellisense saves a lot of time when it comes to implementing interfaces and searching for available methods/variables within a scope.
4. Reactjs and Nodejs used to build a single page website. Utilizes angular to route all those pages so a single page view is seamless
5. Microsoft Azure Portal – for hosting the MySQL server and web server