

lota

CS 489 Thesis Project

Decisions made so far.

The original plan for lota was to develop exclusively using Node.JS however, as the project expanded this became impractical as Node can become over saturated with dependencies to accomplish simple tasks. For this reason, it was decided that the webapp function of lota shall be developed using simpler frameworks, and more custom developed code, than originally planned. The backend will continue to be developed in Node, as it handles stateless programming styles expertly, and its native JSON and API support makes it backend friendly. MongoDB has been selected as the database, also due to its strong JSON support.

Server Design

In order to be considered a modern web app, lota would need to have a strong API. For this reason, an “API First” programming style was selected. Simply put, this is the method of programming, that focuses on the server and its API layer, before the end user design, or feature implementation. This encourages the developer to create a responsive logic layer, that is not cluttered with code unconcerned with data. It also makes the concept of “stateless” programming easier to execute. The “stateless” programming style allows lota to perform at a significantly higher efficiency than what would be expected. Every transaction with the server is calculated purely off the data provided and without any data being stored or tracked in memory. While this results in more database calls, it means that the applications compute time does not hinge on system memory being consumed by processes tied to each active user.

By taking the above concepts into account, Node.JS was decided as the best application for lota’s logic/API server. By having separate stateless servers for authentication and logic, lota will behave in a high-speed manor that is conducive with current industry standards.

Security.

For lota to be properly utilized by Greek organization, modern security is a must. The stateless nature of the application also creates unique security challenges. Fortunately, stateless authentication is not a new concept. JSON Web Tokens (JWT’s) are an existing technology that has been created specifically for the purpose of high-speed authentication. The concept is that the server provides the client with a signed token that identifies the user. This token is then used with every transaction and is checked each time to be valid. So long as the token retains validity, the user retains access. These tokens are by their nature unforgeable.

Currently, there exists an even more secure (yet more complex) system called OAuth 2.0, that builds on the fundamental ideas of JWT’s. This will be considered as a replacement for JWT’s in this project if time allows.

Front End Design

While many would argue that the front-end is the most essential part of the application as it is user facing, a contrary argument can be easily made. The user experience is one of the most fluid elements of a project. It changes considerably depending on current design trends. The front-end will be the last aspect of this project to be completed. Significant thought is being put into the API so that when it comes time to develop the user experience, it will be straightforward to do so given the functionality of the API.

Current Progress summarized.

Iota is making significant progress, most of which has been in documentation and system design, as the actual implantation is rather straightforward. Existing systems that will be utilized have been selected, and the server software groundwork has been completed. The next step is expanding the API functionality without any authentication. This will allow core functionality to be developed without spreading development between multiple facets simultaneously.