1.1 Range of requirements, functional requirements, measurable quality attributes, qualitative requirements

The range of requirements should use JavaScript programming language, Utilization of ES6 features such as destructuring and async/await, Node.js engine for running the API, MongoDB for the database, Mongoose for data modeling and CRUD operations, JWT for authentication, Bcrypt for password encryption, Express package for API development, Jest for unit and integration testing, Option for performance testing and metrics reporting using Artillery, Possibility of building a UI on top of the API using additional frameworks such as Puppeteer.

The functional requirements should include user registration, authentication, access levels, order placement, database management, security, testing, performance evaluation.

* User registration: users should be able to register and create an account.
* Authentication: users should be able to log in to their account using a username and password.
* Access levels: the system should have two types of user access levels - Admin and User, with elevated permissions for the administrator and restrictions for the simple user.
* Order placement: users should be able to place an order for predefined food boxes (Box1 and Box2) with a limited number of choices.
* Database management: the system should be able to store and manage user information and order details in a MongoDB database.
* Security: the system should ensure the security of user data through the use of JWT for authentication and Bcrypt for password encryption.
* Testing: the system should have unit and integration tests in place using Jest.
* Performance evaluation: the system should have the capability to measure performance and report metrics using Artillery.

The measurable quality attributes are:

* Security: the level of protection provided to user data, such as password encryption and authentication through JWT.
* Performance: the responsiveness and efficiency of the API, measured in terms of response time, throughput, and resource utilization.
* Scalability: the ability of the system to handle an increased load and number of users, without affecting the overall performance.
* Reliability: the consistency and dependability of the API in delivering correct results, with a low rate of errors and failures.
* Usability: the ease of use for the users in performing tasks, such as registration, authentication, and order placement.
* Maintainability: the ability of the system to be updated and modified without causing significant disruption, making it easier to fix bugs and add new features.
* Test coverage: the extent to which the system is tested and the quality of the tests, ensuring the reliability and stability of the system.

Some of the qualitative requirements should include:

* User experience: the overall feeling of the users when interacting with the API, such as ease of use, intuitive navigation, and visual appeal.
* Flexibility: the ability of the system to accommodate changes and customization, such as adding new food boxes or updating the ordering process.
* Readability: the clear and concise structure of the code, making it easy for developers to understand and maintain.
* Code quality: the quality of the code and efficient use of resources.
* Error handling: the system's ability to detect and respond to errors in a meaningful way, providing clear and helpful error messages to the users.