

01. How you achieve non functional requirements should be discussed in detail.

- **User Friendliness**

We design user interfaces according to the UX/UI principles. For the easier navigation and using of the system, we give some tips and a “Help” option for the users. And also in every input field users will be given a brief explanation of the task to perform. The website will be responsive and fit into the size of the browser and also different mobile screen sizes.

We also create a mobile application. So customers who cannot login through the website can use it.

The web application will have simple and understandable user interfaces so that interaction between user and system will be successful.

The web application will adapt familiar interfaces (eg: payment interface will look like the payment interface of eBay) so the users will not be confused.

- **Accuracy and Reliability**

The system's reliability and 100% uptime will be guaranteed by using dedicated private servers. There will be two backup servers to ensure the system stays online if one system fails to operate. All of the system data will be synchronized among the main server and backup servers every 30 minutes.

For the data accuracy, admin registers employees, moderators and accountants with the system and they should provide real data. Before adding a delivery person to the system we validate his license details.

In our system customers have a complaint and review option. Then they can rate products.

- **Availability**

The system guarantees its availability for customers for 24 hours every day. If a customer needs to order products he can make an order any time by using the web application or mobile app and it will be delivered within a week.

- **Security**

Each user would have different user privileges and different dashboards for their user accounts. The system will prompt users to enter a strong password for their accounts; otherwise, they will not be able to sign up into the system. For user authentication and authorization we are going to use spring security. We encrypt all the payment details of users and then store them in the database.

02. Why employee handling is out of scope. Discuss with your supervisor more and check whether the scope is enough.

We have added an employee handling component to the system. Then the admin can add/ remove/ update/ delete an employee and approve a leave request and the employee can request a leave, view leave statistics, view his/her salaries. The accountant can handle employee salaries.

03. Spring boot is a microservice based architecture. Therefore, implement the system to reflect micro service principles.

We are planning to implement the system according to the microservice based architecture.

04. Implement reports with good filtering techniques, Generate printer friendly reports.

We are generating reports such as sales reports, salary reports, inventory reports etc, with filtering techniques based on time, product and category based. Admin and accountant can view these reports graphically.

05. Bulk upload and export features would be useful.

When the admin updates the stock, he can import a file with all the products that he wants to add, instead of adding products one by one.
And also we generate printer friendly reports.