1. What would you say are your greatest strengths?

I have noticed that some of my greatest strengths come from my communication skills and ability to work in a team. For example during my most recent project, I took inative and appointed myself as our scurm leader. As scrum leader I made sure everyone was on the same page about how the project was to go about. I also gave tasks to each member of the group and set up daily stand-up meetyingd to… This resulted in my team finishing the project is record time allowing us to put more detail into the application.

1. What are your greatest weaknesses?

One thing I have noticed about myself is my bad time management skills. I’m am alos ocncerened about implementing every single dtail of a project that I lose track of time and end up spending more time to finisha project than I initially thought I would. To beat this, I employed the use of work composer, a time tracking software which made me more time consious and allowed me to never miss a deadline

1. What are exceptions in java?

Exceptions are any abnormal event that occur during the execution of a program. They disrupt the normal flow of a program’s executions.

1. What are exceptions used for?  
   Exceptions are represented as exception obejcts which encapsulate information about an exception such as the type of error, the location where it occurred and any additional details that might be helpful in debugging and handling the error
2. In java

It is possinly to have a try block eithout a catch blovk. However, omitting the catch block means you must include the finally block. This allows you to specify code that would run regardless of whether or not an exception occurred

1. What is the difference between a checked and an unchecked exception?

Checked exceptions are exceptions that the compiler requires you to catch or handle before execution. These exceptions are anticipated hence the compiler requires them to be handled.

Unchecked exceptions are exceptions that do not need to be explicitrly caught because the complier does not require any handling requirements for these exceptions

1. S

Throw is used to throw an exception explicitely whether checked or uncheckld whiles throws is used ona method to declare on or more exceptions that might occure whenrunning thr method hence informing developers about the exceptions and their types and enabling developers to handle such exceptions

If your method might throw an exception, surround it with a try catch block when calling it.

1. User requirements for banking system

* Functional requirements
* User should be able to perform transactions
* Users should be able to create account
* Non functional requiements
* Use angular for the front-end
* Use java for the backend
* Questions
* Any authentication and authorization
* Multiple account types
* Notifications
* Currency
* What kind of trancsactions
* When is account created
* How users deposit and withfraw
* Create multiple account or just one

1. Will the banking system support multiple currencies, or will it be limited to a specific currency?

- This question helps determine if the system needs to handle currency conversion and exchange rates.

2. Should the banking system support different types of accounts, such as savings accounts, checking accounts, or investment accounts?

- Understanding the account types allows for proper design and functionality implementation.

3. What security measures should be in place to ensure the safety of user accounts and transactions?

- This question addresses the need for authentication, encryption, and other security measures to protect sensitive information.

4. Will there be any transaction limits or restrictions imposed on user accounts?

- Knowing if there are any specific limits or restrictions helps define the system's behavior and ensure compliance with regulations.

5. Should the banking system provide account statements or transaction history for users?

- Understanding the need for transaction history assists in designing appropriate features and data storage requirements.

6. Will there be any fees or charges associated with transactions or account maintenance?

- This question determines whether the system needs to calculate and handle fees and charges for various banking activities.

7. Are there any specific regulatory or compliance requirements that need to be considered?

- Knowing if the system needs to adhere to specific regulations or compliance standards helps ensure legal and regulatory compliance.

8. Should the banking system provide real-time balance updates to users?

- Understanding if the system needs to update account balances in real-time can impact the architecture and performance considerations.

9. Are there any specific integration requirements with external systems or services, such as payment gateways or fraud detection systems?

- This question identifies the need for integrating with external systems or services to enhance functionality or security.

10. What level of scalability is expected for the banking system?

- Knowing the expected number of users and transactions helps determine the system's scalability requirements.

11. Should the banking system support notifications or alerts for specific account activities, such as low balance or suspicious transactions?

- Understanding the need for notifications helps design appropriate alert mechanisms and communication channels.

12. Are there any specific accessibility requirements, such as support for visually impaired users or multi-language support?

- This question addresses the need for accessibility features and localization options.

13. What are the desired response times for different banking operations?

- Understanding performance expectations helps in designing and optimizing the system to meet response time requirements.

14. Are there any existing banking systems or legacy systems that need to be integrated or migrated?

- Knowing about existing systems assists in planning the integration or migration process effectively.

15. What are the legal and regulatory requirements for data storage, retention, and privacy?

- Understanding legal and regulatory obligations ensures compliance with data protection and privacy laws.

16. Are there any specific reporting or auditing requirements for the banking system?

- This question helps identify the need for generating reports or maintaining audit logs for regulatory purposes.

17. Should the banking system provide customer support or helpdesk functionality?

- Understanding if customer support features are required helps in designing appropriate support channels and workflows.

18. Will the banking system support mobile devices or have a dedicated mobile application?

- This question determines if the system needs to be mobile-friendly or have separate mobile application development.

19. Are there any specific performance or throughput requirements for the banking system?

- Knowing the desired performance metrics helps in designing a system that meets the required performance criteria.

20. Will the banking system have multi-factor authentication capabilities or other advanced security features?

- This question explores the need for additional security measures beyond basic authentication for user accounts.

1. Requirements for fruit juice

* Requirements
* Update inventory management system to accomodat the new producvt
* Track stock levels of new product
* Support ordering and deliverying processes for new product
* Questions
* Constraints regarding price and percentage

Key Requirements for Updating the Inventory Management System:

1. Add support for the new fruit juice product.

2. Track stock levels of all products, including the fruit juice.

3. Manage the ordering and delivery processes for the fruit juice.

4. Capture specific information about the fruit juice, such as the type of fruit used and the percentage of real fruit.

5. Incorporate the catchy name and attractive packaging design for the fruit juice.

6. Consider the market research findings regarding the price point and target audience for the fruit juice.

Based on these requirements, the following class and its properties can be created:

Class: FruitJuice

- Properties:

- name: stores the catchy name of the fruit juice (string)

- packagingDesign: stores the information about the attractive packaging design (string)

- fruitType: stores the type of fruit used in the juice (string)

- realFruitPercentage: stores the percentage of real fruit in the juice (float)

In the Inventory Management System, the following methods can be added or updated to support the fruit juice:

1. AddFruitJuice(): Adds a new fruit juice product to the inventory.

2. UpdateFruitJuiceStock(): Updates the stock level of the fruit juice product.

3. OrderFruitJuice(): Places an order for the fruit juice based on the desired quantity and target audience.

4. GetFruitJuiceDetails(): Retrieves and displays the specific information about the fruit juice, such as the fruit type, real fruit percentage, catchy name, and packaging design.

Missing Information:

To complete the task, the following missing information would be helpful:

1. What are the specific data requirements for tracking stock levels, ordering, and delivery processes? (e.g., database schema, APIs, integration points)

2. Are there any specific validation or business rules related to the fruit juice product that need to be considered? (e.g., minimum stock threshold, pricing rules)

3. Are there any specific integration requirements with other systems or modules within the Inventory Management System? (e.g., invoicing, reporting)

4. How will the fruit juice be packaged and stored in the inventory? Are there any specific storage considerations?

5. Are there any specific user roles or permissions that need to be implemented for managing the fruit juice product in the system?

By obtaining answers to these questions, the developer can better understand the implementation details and tailor the solution to meet the specific requirements of the company.