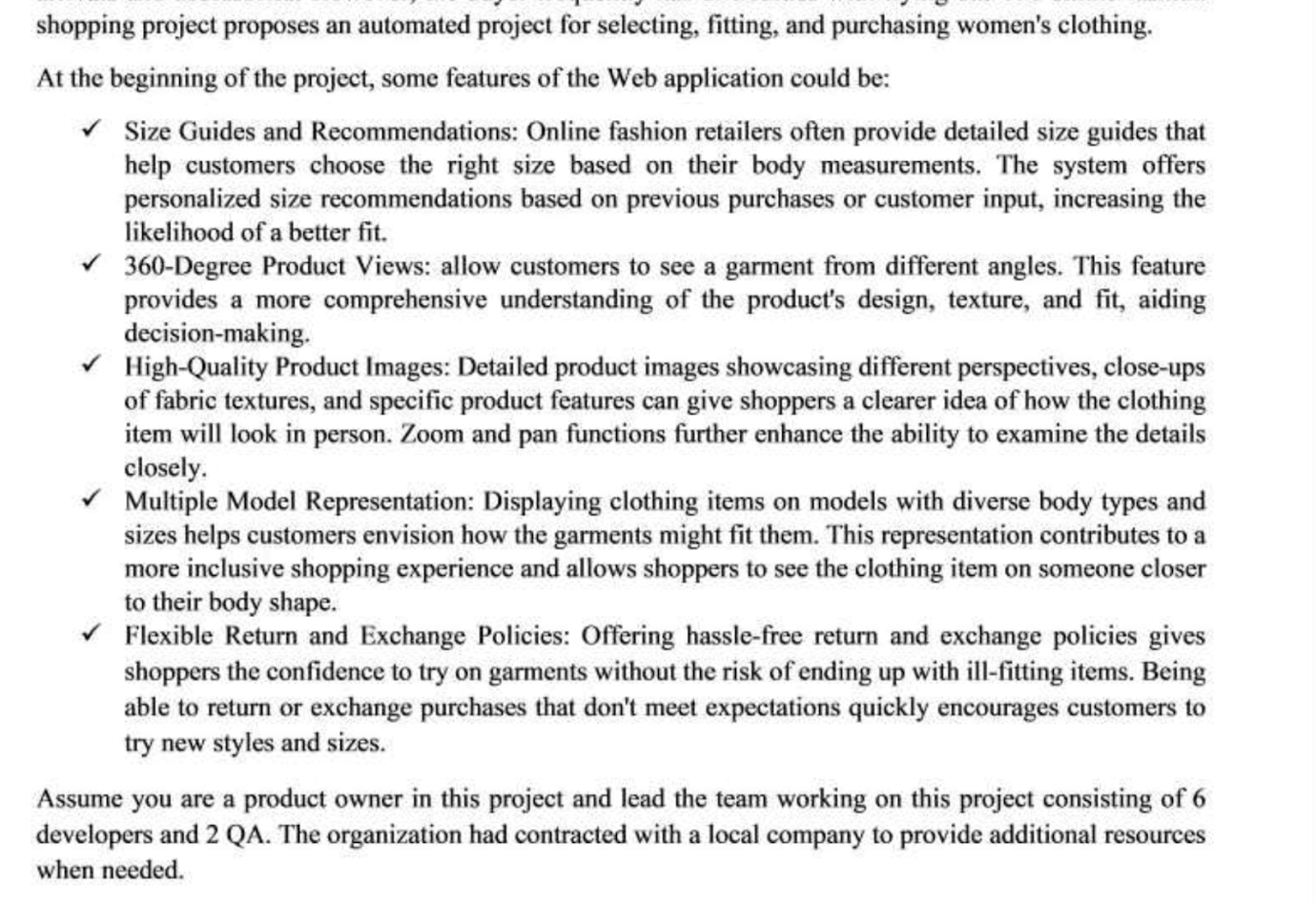
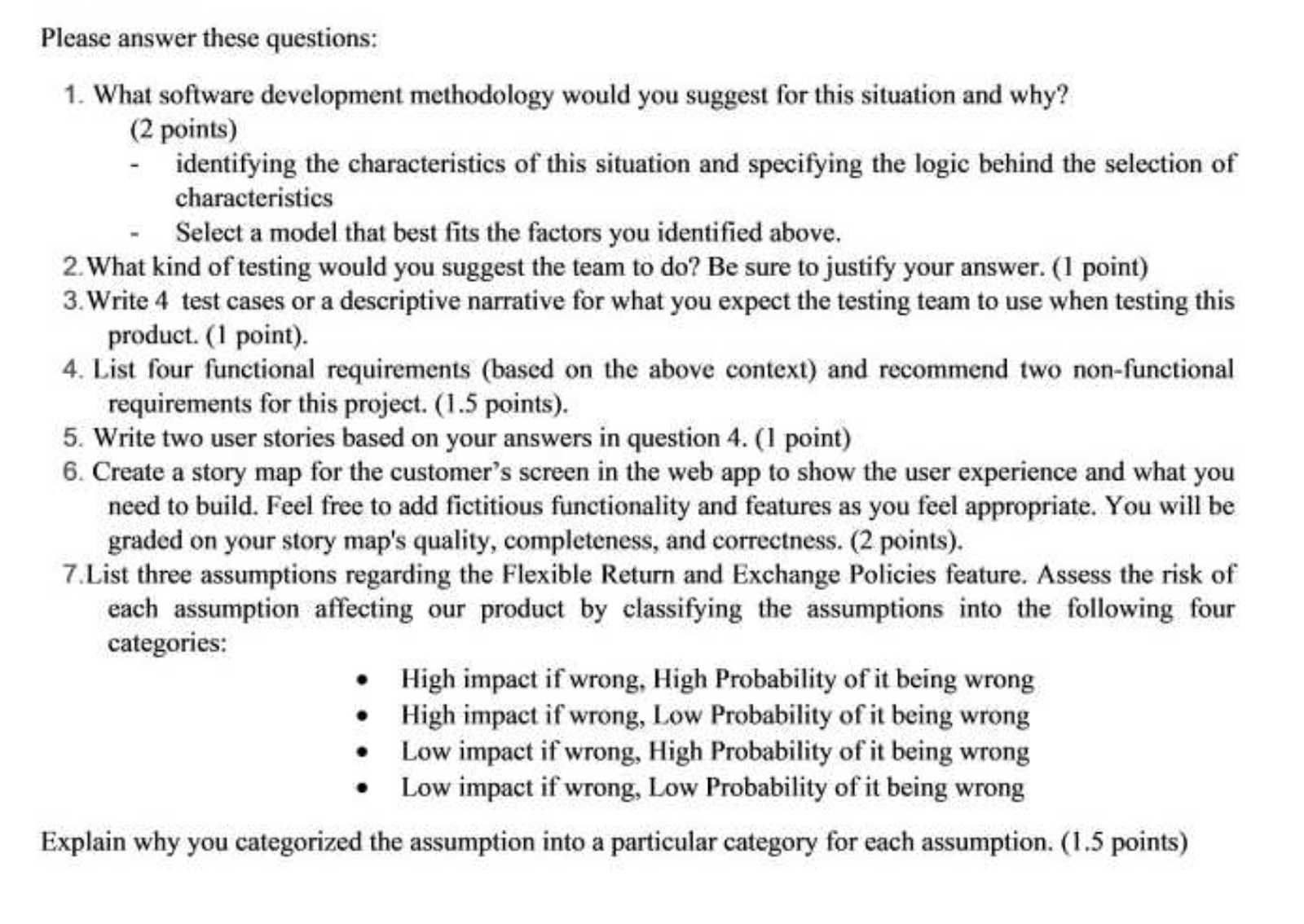
****

****

**Question 1:**

Considering the specific situation and my role as a product owner, I propose the adoption of the **AGILE** development methodology due to the following reasons:

***a) The first aspect relates to the nature of the requirements:***

**+ Flexibility:** The initial requirements are not precisely defined and may evolve or undergo slight modifications during the app development process to stay aligned with market trends and user preferences.

**+ Adaptability:** Given the indistinct nature of the requirements, they might experience changes or minor adjustments throughout development to accommodate evolving customer preferences.

**+ Covering both functional and non-functional aspects:** The project involves a multitude of functional and non-functional requirements, as evidenced by the description of 7 distinct features within the context. This complexity necessitates early detection of any faults for prompt rectification, avoiding costly rework, and preventing disruption to interconnected modules.

- Four Functional:  + Size Guides

                    + 360-Degree Product Views

                    + Multiple Model Representation

                                 + Recommendations

***b) The second aspect pertains to the Development team:***

**+ Team composition:** The team comprises 6 developers and 2 QA members, rendering it well-suited for a project employing the Scrum development methodology.

**+ Effective collaboration and communication skills:** The team members share a common language, facilitating seamless communication and productive teamwork. This advantage significantly contributes to timely task completion.

**+ Proficiency in designated roles and responsibilities**: Each team member possesses a high level of expertise and experience in their assigned roles and responsibilities.

**+ Adaptability to changes:** The team demonstrates a robust ability to adapt to changing circumstances.

***c) The final aspect concerns user involvement in the project:***

**+ Project size:** Medium

**+ Extent of user engagement:** Considered large due to the project's focus on internal company use, primarily targeting employees. This significant user involvement stems from the project's scope.

In conclusion, based on the afore mentioned characteristics, I reiterate my recommendation for **the AGILE METHOD** as the most suitable approach for this project. This choice is grounded in the project's medium scale, intricate array of requirements, the critical need for early fault detection, and the methodology's applicability to projects characterized by initially unclear customer requirements...

Based on the description I provided earlier, **Scrum** appears to be the most suitable model. Here's how I implement it:

+ First, I gather all necessary requirements from team members to create a product backlog.

+ Next, my team and I hold a sprint planning meeting to define the goals and scope of the upcoming sprint. We prioritize items from the backlog and establish a detailed plan for the development activities during the sprint.

+  I plan to divide the project into 4 sprints, each lasting one to two weeks.

* **Sprint 1:** *Size Guides and Recommendations*
* **Sprint 2:** 360-Degree Product Views
* **Sprint 3:** High-Quality Product Images, Multiple Model Representation
* **Sprint 4:** Flexible Return and Exchange Polices

\* During each Sprint, my team will handle both development and testing simultaneously. This entails six developers dedicated to development tasks, while two quality assurance (QA) experts take charge of the testing process. Our front-end development is carried out using React, while backend development is accomplished using Node.js.

\*I have chosen to utilize Jira as the tool for project management and githud to manage the code.

**Question 2:**

Regarding testing, the team might contemplate performing the subsequent varieties of tests:

Unit testing: To guarantee proper functioning of each segment of the application.

Integration testing: To confirm the smooth collaboration of distinct elements within the application.

Acceptance testing: To evaluate the application's operations from the user's viewpoint, ensuring alignment with defined criteria and provision of a favorable user encounter.

Regression testing: To verify that modifications to the application do not disrupt pre-existing functions.

**Question 3:**

* Four test case:

Test case 1:

+ Description : test performance of the system

+ Input : 10000 user buy same products in this time.

+ Expected: All customers can place orders

+ Output: All customers can place orders

+ Exception: System inform error message “API Connection Fail"

Test case 2:

+ Description : Test layout for each device

+ Input : Open the screen of system in mobile device

+ Expected: Display successfully different layout for each device

+ Output: Display successfully different layout for each device

+ Exception: The UI does not meet the requirements

Test case 3 :

+ Description : Test Upload Image Functional

+ Input: Upload image have the size larger than 1000 mb

+ Expected: The system informs error message “Size image exceeds size allow"

+ Output: The system informs error message “Size image exceeds size allow"

+ Exception: The system informs error message “Size image exceeds size allow"

—----------------------------------------

+ Input : Testing buy products and pay it using banking, when it still loading and broken internet.

* Output : Undo money when product payment failed.

+ Input : Testing in non-cookie

* Output : Program still work correctly.

**Question 4:**

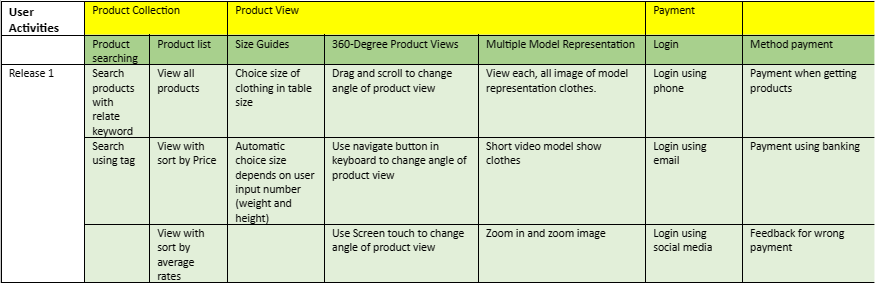
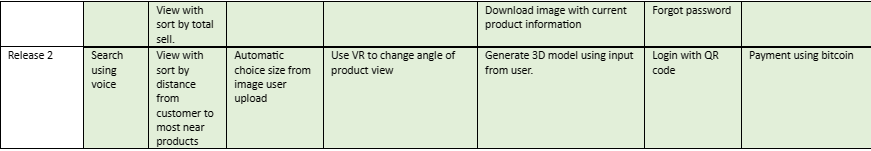
* Four functional requirements
  + Size Guides: allow customers to input their weight, height and through calculations to determine a right size based on their body. Or they don’t want automatic, they can see the table size and choice with number they want.
  + Recommendations: allow customers input keyword of their product they want and use intelligent algorithms to find all the product like most with keyword.
  + 360-Degree Product Views: customers can see their product they want with different angles and this future provides a more comprehensive know the products design, texture, and fit, aiding decision-making, through they can what they want.
  + Multiple Model Representation: User can see a clothing items on models with diverse body types and size help customers envision how the garments might fit them.
* Two non-functional requirements
  + **Browser Compatibility:**The system should be compatible with the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari).
  + **User Interface Consistency:** The user interface should follow the company's branding guidelines and maintain a consistent design.

**Question 5:**

+ As a customer, I want to Size Guides, so that I can be easy to find right size of clothing items I want with my body.

+ As a product owner, I want to Browser Compatibility, so that I want to more customers can join. Because the number of people use Chrome, Firefox, Edge … is really hight.

**Question 6:**



**Question 7:**

**-Low Impact If Wrong , Low Probability being wrong**

***Situation:*** *Intermittent User Interface Glitches*

**Cause:** Infrequent incidents of minor glitches within the user interface, such as temporary visual discrepancies or graphical anomalies, with a low likelihood of occurrence. The effect is minor as these glitches are short-lived and do not impede users' capacity to complete tasks. Users can smoothly continue utilizing the software without substantial interruptions.

**- High Impact, Low Probability**

***Scenario:*** *Unexpected Breakdown in Integration*

**Reason:** Infrequent cases of integration breakdowns between software components, usually triggered by factors such as version disparities, with a slim chance of occurrence. The notable consequence results from interrupted data exchange and inter-system communication, potentially leading to delays and inaccuracies.

**-High Impact, High Probability**

***Scenario:*** *Sluggish System Performance During Busiest Periods*

**Reason:** Regular occurrences of system slowdowns and delayed responses encountered by users during peak usage hours, resulting in a strong likelihood of this situation happening. The notable impact arises from reduced efficiency and user discontent caused by extended waiting periods.