### **The analysis and testing of the received mobile application requirements.**

### **The list of improvements to the requirements can be found at the end of the document.**

**Author:** O. Solodka

**Date:** 19.09.2024

### 

| **Requirement** | **Detailed Requirement**  **description** | **Analysis** | **Score** | **Comments** | **Check list** |
| --- | --- | --- | --- | --- | --- |
| **1. Registration and Authorization** | ***1.1 The user can create an account using an email address and password*** | This requirement is clear and understandable. It is important to verify email and password validation, and to ensure there is a mechanism for password recovery | **5** | **It is necessary to clarify:**Are there other methods of registration and authentication? (Google, Apple, Facebook)?The requirements for password complexity and email validation. (The minimum password length, the inclusion of uppercase letters, numbers, and other symbols).Security protection mechanisms (blocking after multiple failed attempts, data encryption).Access recovery procedures (for example, forgotten password).Edge case handling (such as incorrect data entry or duplicate accounts). | 1. Verify that the user can enter a valid email address.  2. Check if the system only accepts valid email formats (includes "@" symbol and domain).  3. Ensure the system rejects invalid email addresses.  4. Verify that the password meets minimum requirements (length, inclusion of uppercase letters, numbers, special characters).  5. Verify that the user can confirm the password (re-entering the password).  6. Check if the system shows an error message when the passwords do not match.  7. Verify that the system alerts the user if the email is already registered.  8. Check if the system sends a confirmation email for registration.  9. Verify that a new account is created after the registration confirmation.  10. Verify that there are limitations on the number of failed login attempts (account locking).  11. Ensure that data is transmitted over a secure connection (SSL/TLS) during registration and login. |
|  | ***1.2 The user can log in to the app using their email and password.*** | Need to test successful login with correct credentials and unsuccessful login with incorrect credentials. Also, check if there is protection against attacks such as brute force |  | **It is necessary to clarify:**Provide clarity on the user experience (feedback messages, "Remember Me" feature) | 1. Verify that the user can log in using the registered email and password.  2. Ensure the system does not allow login with invalid email or password.  3. Check if the system provides an error message for incorrect password or email.  4. Verify that the system allows password recovery via "Forgot Password".  5. Check if the system blocks the account after several failed login attempts.  6. Verify that the user can receive a password reset email to the registered email address.  7. Verify that the system accepts a new password after it is reset.  8. Verify that the email and password input fields are displayed correctly.  9. Check if the system provides real-time error messages (invalid email format, short password).  10. Ensure that the user can toggle password visibility during input.  11. Check if there is an option to remember login credentials  12. Verify if two-factor authentication (2FA) can be enabled.  13. Check if the user can log in using alternative methods (social media,Facebook Google, Apple), if supported. |
|  |  |  |  |  |  |
| **2. Product Browsing** | ***2.1 The user can view the product catalog, which includes product name, image, description, price, and availability*** | This requirement for viewing products involves displaying detailed product information, providing robust filtering and search functionalities, and ensuring that these features work together seamlessly. Each component must be logically consistent and user-friendly, addressing potential edge cases and ensuring accurate, real-time data | **5** | **It is necessary to clarify:**Are there any restrictions or specific rules for formatting names (maximum length, special characters)?What are the requirements for image size and quality? How many images can be associated with each product? What file formats are supported?What elements should be included in the description (technical specifications, usage instructions)? What should be the structure of the description?How frequently should price information be updated? What types of prices should be displayed (discounted prices, before and after discount prices)?How often should availability information be updated? What availability statuses should be displayed ("in stock," "out of stock," "limited quantity")? | 1. Verify that the product description includes all necessary details.  2. Ensure the description is accurate and free from spelling and grammatical errors.  3. Check that the description is well-formatted and easy to read.  4. Verify that the product price is displayed correctly.  5. Ensure the currency and price format adhere to standards.  6. Check that any discounts or promotional prices are displayed correctly.  7. Verify that the availability status of the product is accurate.  8. Test how the system behaves when there are no products available in the catalog.  9. Ensure products with incomplete or missing data are displayed correctly |
|  | ***2.2 The user can filter products by categories, brands, price range, etc*** | Ensure that filters work correctly and that filtered results are displayed accurately |  | **It is necessary to clarify:**What categories should be included? How frequently should categories be updated?What brands should be included in the filter? Should there be the ability to add new brands?What should be the standard price ranges for filtering? Is there an option for users to set their own price range?What additional filtering criteria are needed (rating, color)? Which ones are mandatory?What messages should be displayed when no results are found? How should the system behave when no products match the criteria? | 1. Verify that all relevant categories are available for selection.  2. Test whether selecting multiple categories works as expected.  3. Verify that all relevant brands are listed.  4. Check that selecting a brand filters products correctly.  5. Verify that the price range filter displays appropriate ranges.  6. Ensure the price range can be set using sliders, input fields, or other provided methods.  7. Test combinations of different filters to ensure they work together correctly.  8. Check that applying filters does not result in excessive loading times.  9. Ensure filtering works smoothly on different devices and screen sizes.  10. Verify that users receive appropriate feedback when filters are applied or reset.  11. Test how the system handles cases where no products match the applied filters.  12. Ensure the system handles incorrect filter inputs gracefully. |
|  | ***2.3 The user can search for products by name or keywords*** | Test the search functionality to ensure it works correctly and efficiently. Verify that the search algorithm returns relevant results |  | **It is necessary to clarify:**Specify the formats or types of input allowed (e.g., single words, phrases, special characters) Clarify if the search supports partial matches (e.g. searching "bike" should return results for "kids bike") | 1. Verify that different input formats are supported (e.g., single words, phrases, special characters)  2. Verify that search results match the input criteria accurately  3. Ensure that fuzzy matching handles typos or close variations of keywords.  4. Check how the system handles cases where no products match the search criteria (e.g., displays a "No results found" message)  5. Ensure that the search functionality works with multiple languages and non-English characters.  6. Ensure that the search functionality is protected against potential security threats such as injection attacks  7. Verify Product Catalog Pagination |
|  |  |  |  |  |  |
| **3. Product Details** | ***3.1 The user can view detailed information about a product, including specifications, description, rating, reviews, and product images*** | This requirement involves checking all aspects related to the display of detailed product information. This includes accuracy and completeness of information, ease of viewing, responsiveness, and usability. Verify that all functional elements work correctly, providing users with a clear and comprehensive understanding of the product. Also verify that all product details are displayed correctly and updated if data changes. Test interactions with reviews and ratings as well. | **5** | **It is necessary to clarify:**What specific characteristics should be included? (e.g., dimensions, weight, materials, technical specifications)How should these characteristics be presented? (e.g., table, list, text blocks)What length is optimal for the description? (e.g., several paragraphs or a brief summary)Where and how should the rating be displayed? (e.g., stars, numeric value)What format should the reviews take? (e.g., text reviews, star ratings, comments)How many images should be available? (e.g., main image plus additional ones)What format and quality should the images be? (e.g., file size, resolution)How should the product details be displayed on different devices (mobile phones, tablets, computers)?Are there mechanisms for verifying the validity of the product information? | 1. Verify how product technical specifications are displayed.  2. Verify that the product description is detailed and clear.  3. Verify how the product rating is displayed.  4. Verify that user reviews are displayed correctly and are understandable.  5. Verify the quality of product images.  6. Verify the ability to view images in an enlarged format or as a slideshow.  7. Verify how product details are displayed on different devices (mobile phones, tablets).  8. Verify Image Zoom Functionality |
|  |  |  |  |  |  |
| **4. Adding Products to Cart** | ***4.1 The user can add products to the shopping cart*** | This requirement involved not only the functionality of adding products but also ensuring proper interaction with the cart, including quantity adjustments, item removal, and interface consistency. Additionally, responsiveness across various devices should be verified to ensure a seamless user experience. Also check if products are successfully added to the cart and displayed correctly. | **5** | **It is necessary to clarify:**Are there limitations on the number of products that can be added to the cart? What information about the product is displayed in the cart (name, image, price, quantity)?  How does the system handle situations when a product is unavailable or out of stock?  How does the functionality for adding products to the cart work across different devices (mobile phones, tablets, computers)? | 1. Verify that clicking the "Add to Cart" button or equivalent adds the product to the cart.  2. Ensure the user receives a confirmation message or visual indication (e.g., a pop-up or badge update).  3. Check that the product appears in the cart with correct details (name, image, price, quantity).  4. Ensure that the total amount and item count are updated accurately.  5. Check how the system handles errors such as trying to add an out-of-stock item to the cart |
|  | ***4.2 The user can change the quantity of items in the cart or remove products*** | Verify that the quantity can be adjusted and that changes are reflected accurately in the cart. Also, test the functionality for removing items |  | **It is necessary to clarify:**How can users change the quantity? (e.g., through input fields, plus/minus buttons, sliders)Are there limits on the minimum and maximum quantity that can be set for a product? How does the system handle invalid quantity inputs? (e.g., non-numeric values, negative numbers)  How can users remove products from the cart? (e.g., delete buttons, checkboxes for selection)  Can users remove multiple items at once? If yes, how is this handled?  What feedback does the user receive after changing the quantity or removing an item? (e.g., success messages, error messages) | 1. Verify that users can increase or decrease the quantity of a product in the cart.  2. Ensure the cart updates the total amount and quantity correctly after changes.  3. Check for maximum and minimum quantity limits  4. Verify that users can remove a product from the cart.  5. Ensure the cart updates correctly by reflecting the removal of the product.  6. Check if there is a confirmation required for removal.  7. Verify that items remain in the cart when the user navigates away from the cart page or logs out and logs back in  8. Ensure that all relevant product information (name, image, description, price) is displayed correctly in the cart. |
|  |  |  |  |  |  |
| **5. Checkout** | ***5.1 The user can review and modify the contents of their cart before placing an order*** | “Checkout" requirement is critical for e-commerce and must be highly accurate and user-friendly. The main functional aspects such as reviewing and modifying the cart, providing a delivery address, selecting payment methods, order summary, and confirmation must work seamlessly. Ensure that users can edit cart contents before finalizing the order and that all changes are saved correctly | **5** | **It is necessary to clarify:**What specific details should be visible in the cart (e.g., item names, quantities, individual prices, total cost)? How should the cart update in real time as items are added or removed? Is there a need for confirmation or validation before changes are finalized?  How is the order confirmation delivered to the user (e.g., email, SMS, on-screen confirmation)?  How does the checkout process integrate with backend systems such as inventory management, order processing, and shipping?  Are there any third-party services involved (e.g., payment gateways, shipping providers) that need to be integrated? | 1. Check that after reviewing and modifying the cart, the user can smoothly proceed to the checkout process.  2. Ensure that any changes made to the cart are reflected in the checkout summary.  3. Ensure that the cart’s subtotal, taxes, and total cost are recalculated correctly whenever an item is added, removed, or its quantity is changed. |
|  | ***5.2 The user can provide a shipping address and contact information*** | Test the functionality for entering and saving shipping addresses and contact information. Also, check data validation |  | **It is necessary to clarify:**What fields are required for the delivery address (e.g., street address, city, postal code, country)? Are there any special formatting requirements? What details are required for contact information (e.g., phone number, email)? Are there any formatting rules or validation checks?  How is the validity of the delivery address checked? Is there integration with address verification services?  Can users select from multiple saved addresses or add a new one? | 1. Ensure that the form includes fields for all necessary components of the shipping address (e.g., street address, city, state/province, postal code, country).  2. Verify that fields are labeled clearly and that any required fields are marked appropriately.  3. Check if the system validates the format of the address fields (e.g., postal code, phone number).  4. Ensure that the system provides error messages for invalid or incomplete address inputs.  5. Confirm that the form includes fields for contact information, such as phone number and email address.  6. Verify that the phone number and email address fields accept valid formats and reject invalid entries.  7. Check if the system offers auto-fill suggestions or address lookups to streamline the address entry process.  8. Ensure that users have the option to edit the address and contact information before finalizing the order.  9. Test how the system handles errors in address or contact information (e.g., missing fields, invalid formats).  10. Verify that appropriate error messages or prompts are provided to guide the user in correcting the information.  11. Check that users can review and confirm their shipping address and contact information before placing the order.  12. Verify that the address and contact information form is functional and user-friendly across different devices and screen sizes (e.g., desktop, tablet, mobile).  13. Confirm that the shipping address and contact information are securely handled and not exposed to unauthorized access. |
|  |  |  |  |  |  |
| **6. Payment** | ***6.1 The user can choose a payment method such as credit card, payment system, etc*** | This requirement includes several key aspects: it is essential to check the support for various payment methods such as credit cards and electronic wallets, as well as integration with payment gateways. Ensuring the security of payment data through encryption and compliance with standards like PCI-DSS is critical. Validation of payment data, error handling, confirmation of successful payments, and informing users of issues are also crucial. The interface should be user-friendly and clear, with support for different currencies and languages, and the transaction history must be accessible for review. Comprehensive testing of all scenarios and adherence to legal and regulatory requirements are also important.Verify that all payment methods work correctly and that transactions are processed without errors. Also, check the security of the payment process | **5** | **It is necessary to clarify:**What payment methods should be available (e.g., credit card, debit cards, electronic wallets, bank transfers, PayPal) What validation rules apply to payment data (e.g., card numbers, expiration dates)?  How do users receive confirmation of successful transactions?  Can users view, download, or print payment confirmations?  What legal and regulatory requirements must be followed for processing payments? | 1. Verify if the payment methods (e.g., credit cards, debit cards, electronic wallets) are correctly integrated.  2. Confirm that all specified payment gateways are functioning properly.  3. Ensure that payment data is encrypted during transmission and storage.  4. Check compliance with PCI-DSS or other relevant security standards.  5. Validate that payment details (e.g., card numbers, expiration dates) are correctly formatted and validated.  6. Test how the system handles invalid payment data and errors.  7. Confirm that users receive clear confirmation of successful payments. 8. Verify that the system correctly notifies users in case of payment failures or issues.  9. Check if the payment interface is user-friendly and easy to navigate.  10. Ensure there are clear instructions and prompts for users to complete the payment process.  11. Verify if multiple currencies are supported and properly handled. 12. Check if the payment interface is available in multiple languages if required.  12. Ensure that users can access their transaction history.  13. Confirm that users can view, download, or print payment confirmations.  14. Check if the payment process adheres to legal and regulatory requirements relevant to the region. |
|  |  |  |  |  |  |
| **7. Order Tracking** | ***7.1 The user can receive updates about the status of their order, including order confirmation, payment, shipping, and delivery*** | This requirement stipulates that the system must provide users with up-to-date information on the status of their orders, including order confirmation, payment status, shipping status, and delivery status. The system should allow users to access this information through the user interface, send automatic notifications about status changes, and account for international aspects such as languages and time zones. It is crucial to ensure data confidentiality with proper encryption and verify the accuracy and synchronization of status updates with logistics systems. Ensure that users receive accurate and timely updates about their order status | **5** | **It is necessary to clarify:**What specific systems or data sources are used to obtain order status information (e.g., internal databases, partner systems, courier services)?What specific information should be provided to the user (e.g., tracking number, expected delivery date, status of each processing stage)?How frequently is the order status information updated, and how quickly are these updates reflected to the user?How will users access their order status (through a personal account, email, SMS)?How does the system support different languages and time zones for international orders?What security measures are in place to protect users' personal information and payment data?How is accuracy and synchronization of order statuses with external logistics systems ensured? | 1. Verify that order status information (e.g., confirmation, payment status, shipping status, delivery status) is accurate and up-to-date.  2.Ensure that status updates are correctly reflected from source systems or external partners.  3. Check if order status information is updated in real-time or according to specified intervals.  4. Confirm that updates are promptly visible to the user.  5. Verify that users can access order status information through intended methods (personal account, email, SMS).  6. Ensure that users can view complete and detailed order status through their account interface.  7. Confirm that order status information is available in different languages if required.  8. Check if the system correctly handles different time zones for international orders.  9. Verify that user data, including order status and personal information, is protected with appropriate security measures.  10. Ensure that data transmission is encrypted and that access is restricted to authorized users.  11. Check that the system integrates correctly with external logistics and courier systems.  12. Confirm that order status updates from logistics partners are accurately reflected in the system.  13. Test how the system handles errors or discrepancies in order tracking data.  14. Verify that users are notified of issues and that there are procedures in place for resolving them. |
|  |  |  |  |  |  |
| **8. Reviews and Ratings** | ***8.1 The user can leave reviews and rate products*** | This requirement states that users can leave reviews and provide ratings for products in the mobile application. While it adds value by increasing trust in products, it needs to be carefully analyzed from both a logical and functional perspective. Also verify that users can submit reviews and ratings and that these are displayed correctly on the product page | **5** | **It is necessary to clarify:**Whether reviews will be text - only or if additional media (e.g., photos, videos) can be included? Also important to define the rating format—will it be a numerical scale (e.g., 5 stars) or another type of system?Mechanisms for filtering or moderating reviews to prevent spam, inappropriate language, or misleading feedback.How reviews will be displayed—will they be sorted by date, rating, or other filters (e.g., best/worst, with photos/without photos)? Additionally, it’s crucial to determine how ratings will be displayed (average rating, total number of reviews, etc.)?Whether only registered users can leave reviews or if guest users can also provide feedback? This could affect the credibility of the reviews.Will users be able to edit or delete their reviews after posting? This is essential for improving the overall user experience.If the app supports multiple languages, it's important to consider whether users can submit reviews in different languages?Will there be reports available for administrators to monitor the overall satisfaction level with products or address recurring issues or complaints? | 1. Verify that a user can submit a review after purchasing a product.  2. Check if the review includes a text field for feedback.  3. Verify if the review includes an option for a star or numerical rating (e.g., 1-5 stars).  4. Ensure that a user can edit their review after submission.  5. Verify that a user can delete their review if needed.  6. Check if submitted reviews are displayed correctly on the product page.  7. Verify that reviews show the user's name, rating, and date of submission. 8. Ensure that the average rating of the product is updated based on submitted reviews.  9. Verify if users can sort reviews by rating (e.g., highest to lowest, most recent).  10. Check if users can filter reviews to show only those with photos or specific ratings (e.g., only 5-star reviews).  11. Verify that the rating system allows users to select between 1 and 5 stars. 12. Verify that inappropriate content (e.g., offensive language) in reviews is flagged or filtered out.  13. Check if the system automatically blocks multiple reviews from the same user on a single product.  14. Ensure that only registered users are allowed to submit reviews.  15. Check if guest users can view, but not leave, reviews.  16. Verify that reviews submitted in different languages are displayed correctly.  17. Ensure that users can filter or translate reviews by language.  18. Verify that users can upload images or videos with their reviews. 19. Ensure that images/videos are displayed correctly alongside text reviews. |
|  |  |  |  |  |  |
| **9. User Interface** | ***9.1 The UI of the mobile app should match the UI of the web portal for better brand recognition. The UI elements of the mobile app should also match the elements of the web portal*** | This requirement focuses on the visual and functional aspects of the interface, as well as its alignment with the user experience (UX) and the design of the web portal. It is important to ensure consistency between the mobile app interface and the web version to enhance brand recognition and provide a seamless user experience. Additionally, the placement of interface elements, their clarity, ease of use, colors, fonts and responsiveness across different screen sizes and devices must be evaluated. It is also essential to consider the specific design requirements for different mobile platforms (iOS, Android) and ensure accessibility for users with disabilities. | **5** | **It is necessary to clarify:**Are there specific design guidelines that need to be followed (e.g., alignment with brand colors, fonts, and styles from the web version)?Should the design comply with platform-specific guidelines like Material Design for Android or Human Interface Guidelines for iOS?Are there any requirements for how the interface should adapt to different screen sizes and orientations (portrait/landscape)?Does the interface need to adhere to accessibility standards (e.g., color contrast, screen reader support, touch target sizes)?Should the UI be able to support multiple languages and regional settings?Are there performance criteria for how fast UI elements should load or respond to user input? | 1. Verify that the colors, fonts, and other visual elements align with the corporate branding guidelines.  2. Ensure the design follows platform-specific guidelines (Material Design for Android, Human Interface Guidelines for iOS).  3. Check how the interface adapts to different screen sizes (small, medium, large).  4.Test correct display in both portrait and landscape orientations.  5. Verify sufficient contrast between background and text for better accessibility.  6. Check that interactive elements (buttons, links) are of appropriate size for easy interaction.  7. Test for visual or tactile feedback when interacting with buttons or other interactive elements (color change, animations, haptic feedback).  8. Ensure quick loading of interface elements when navigating between screens.  9. Verify that all images and icons display correctly without distortion.  10. Check for proper text display (no truncation or overcrowding).  11. Ensure support for multiple languages and check that UI elements display correctly for each language. 12. Test the functionality of navigation buttons, menus, and other UI elements to ensure proper transitions between sections.  13. Check for crashes or freezing of the interface during user interaction (switching tabs, filling out forms). |

### 

| One of the important questions to clarify is whether the app should be native or hybrid. Native apps offer higher performance and full access to device features, but they require separate development for each platform (iOS and Android). Hybrid apps allow for a single codebase across different platforms, but they may have limitations in performance and integration with device features. |
| --- |

### **Attention should be given to the absence of the following requirements:**

**1. Handling Network Conditions**

**Requirement Overview:** The requirement is that the mobile application must function effectively under different network conditions, including various network types (**Wi-Fi, mobile data - 3G, 4G, 5G**) and network coverage changes. This includes testing the app's performance under different internet speeds, handling network interruptions, and adapting to changes in network quality.

**2. Battery Usage Testing for Mobile Application**

**Requirement Overview:** The mobile application should be tested for battery efficiency. This includes measuring the overall energy consumption of the app during various usage scenarios, such as active use (e.g., browsing content or interacting with features) and passive use (e.g., background processes). Particular attention should be paid to the app's impact on battery life while running in the background, monitoring energy consumption during background tasks, and transitions between active and inactive states. Testing should cover different device models and operating systems to ensure consistent results. It is also important to assess battery consumption under maximum load scenarios, including extreme usage cases, to ensure optimal performance and battery life for all users.

### **3. Error Handling**

### **Requirement Overview:** The requirement specifies that the application should handle and display errors in a user-friendly manner. This includes showing clear, informative messages when errors occur, such as invalid input or network issues, and providing options for users to recover from these errors.

### **4. Handling Incoming Calls and Messages**

**Requirement overview:** The mobile application must handle incoming calls and messages gracefully. This requirement ensures that the app maintains a good user experience and functionality even when interruptions occur due to calls or messages.

**5. Cross-Device Compatibility**

**Requirement overview:** The cross-device compatibility requirement stipulates that the mobile application must function correctly across various devices with different characteristics, such as screen sizes **(including smartphones, tablets, and potentially large screens like those on devices with extended or curved displays**), resolutions (**ranging from high resolutions like HD or 4K to standard resolutions like 720p**), and operating systems (**e.g., iOS, Android**). This ensures that users have a consistent interaction experience with the app, regardless of the device they are using.

**6. Cross-browser testing**

**Requirement overview:** If the mobile application is hybrid, cross-browser testing should be included as a requirement. This involves verifying that the app functions correctly across different web browsers (e.g., **Chrome, Safari, Firefox, Edge)** to ensure consistent display and functionality. It's important to test various browser versions and operating systems to ensure the hybrid app provides a uniform user experience regardless of the environment.