Loomera Quiz Report

Emin Buğra Aksoy 20210601004

Introduction and Meeting Summary

This report contains the requirements analysis for a mobile application to be developed for Umut Can Erduran, a customer operating in the field of wholesale textile sales, as part of the SE 305/321 course. The customer's basic requirements were gathered during a meeting. The client wants an application that can display 5,000 products and accept customer payments via bank transfer, with manual payment approval. The client has also requested features such as in-app live chat, returns management and profit tracking. For this analysis, the Gemini artificial intelligence tool was used to analyse and classify these requirements.

Methodology

Transcription: “An audio recording of the customer meeting was made, and the key requests were transcribed manually to create a summary text.”

Requirements analysis: This is the most important part. You can write it as follows: 'Google's Gemini artificial intelligence model was used for requirements analysis.

Prompt Engineering: First, the detailed summary extracted from the meeting was presented to Gemini as a 'prompt'.

Analysis: Gemini was asked to distinguish between the system's functional and non-functional requirements based on this summary, and to categorise these requirements into the subcategories seen in the slides (usability, performance, etc.).

Validating the AI's output: The lists generated by Gemini (e.g. classifying the 'easy interface' request as 'usability' (NF)) were validated by comparing them with the theoretical definitions in the course materials. Additionally, Gemini was asked to provide further analysis of the potential risks (e.g. delays and scalability issues) that customer requests such as 'manual payment approval' could pose to the system.

Analysis and Findings

Customer (Buyer) Functions:

1)Functional Requirements

1.1) The system shall allow customers to create a new account ("register" button) and log in to their existing accounts.

1.2) The system shall require customers to go through a "verify" (verification) process upon registration.

1.3) The system shall allow customers to view products (sweatshirts, shirts) in a categorized manner via "banners" on the home page.

1.4) The system shall allow a customer to see all models (5000 product types) within that category after clicking a product category.

1.5) The system shall allow a customer to "add to cart" a product they like, specifying the quantity.

1.6) The system shall allow a customer to proceed to the "advance" (prepayment) step by confirming the cart.

1.7) The system shall allow a customer to complete the payment process by selecting the "reference" (wire transfer/EFT) option.

1.8) The system shall allow a customer to view the status of their order (e.g., "delivered," "returned") through the application.

1.9) The system shall allow a customer to initiate a return process using the "I want to return it" button after the order is delivered.

1.10) The system shall allow a customer to ask questions to the Admin (Customer Representative) via the in-app messaging area (live chat).

Admin (Umutcan - Seller) Functions:

[FR-A1] The system shall allow the Admin (Umutcan) to log in to the system.

[FR-A2] The system shall allow the Admin to view the list of customers who have placed an order using the "reference" (wire transfer) option.

[FR-A3] The system shall require the Admin to **manually** check if the payment has been received in their own mobile banking and subsequently approve the order in the system.

[FR-A4] The system shall allow the Admin to **manually** enter the stock quantities for 5000 product types (including returned products).

[FR-A5] The system shall allow the Admin to receive and respond to live chat messages from customers.

- [FR-A6] The system shall allow the Admin to view "profit margin" / "profit share" calculations on a monthly basis by entering costs (product cost, cargo cost) and sales price.

2)Non-Functional Requirements

- **Usability:**
- [NFR-U1] The admin panel must have a "classic panel... simple panel" (classic and simple) interface.
- [NFR-U2] The system must use the client's "own brand logo" and "various phrases."
- **Scalability:**
- [NFR-S1] The system's database and infrastructure must support customer growth starting from the current 12 customers, expanding to 20 this year and 50 by the next year ("progressing process").
- **Performance:**
- [NFR-P1] The system must not slow down under the load of 5000 product types and increasing customer traffic (especially in catalog viewing).
- **Availability & Maintainability:**
- [NFR-AM1] Mandatory system maintenance must be performed during the users' most passive hours, "between 12 AM and 6 AM."
- **Legal:**
- [NFR-L1] The return process must comply with the "base of Turkey... returned within 14 days" (the legal 14-day rule in Turkey).

Discussion and Conclusion

Evaluation of AI Tool: The use of the Gemini AI tool in this experiment demonstrated significant advantages and some limitations in transforming raw transcript data into a structured requirements list.

- **Advantages:**
  - **Speed and Efficiency:** The analysis of a long meeting transcript (Appendix A) was completed in seconds, and all key demands were extracted.
  - **Expert Classification:** The AI was able to instantly assign non-functional requirements to their correct sub-categories, such as the "Easy interface"

demand to "Usability" (NFR-U1) or the "Increasing customer count" expectation to "Scalability" (NFR-S1).
    - **Risk Detection:** In the initial stages of analysis, the AI proactively identified potential business risks associated with the client's "manual payment approval" (FR-A3) demand, such as "Delay Risk," "Error Risk (Reliability)," and "Scalability Risk."
- **Limitations:**
    - **Need for Human Oversight:** The AI could not access the audio recording itself (due to technical constraints) and based its analysis entirely on the *transcript* provided by the user. If the transcript had been faulty or incomplete, the entire analysis would have been flawed.
    - **Lack of Context:** The AI cannot understand the client's tone of voice, which demand was emphasized more (prioritization), or the social dynamics of the meeting. This critical information remains the responsibility of the human analyst (the student) supervising the AI.

Differences: When comparing the requirements elicited with AI in this experiment (Umut Can) against the previously prepared requirements file for our "Loomera" project, it was observed that these two lists represent two different maturity levels of the requirements engineering process. The "Loomera" list is, from an engineering perspective, a much more mature and "better" requirements document.

- **Format and Language:** The "Loomera" list uses standard (IEEE) and legally binding language such as "The system **shall**...", "The system **must**..."; whereas the "Umut Can" list contains the client's raw demands, such as "I want...".
- **Testability & Measurability (The Most Important Difference):**
    - **Performance:** The "Umut Can" list includes an untestable demand [NFR-P1] like "must not slow down," while the "Loomera" list (#13) provides a clear, measurable, and testable metric: "The system must be able to support **at least 100 users**... page load times... should be **under 3 seconds**."
    - **Compatibility:** The "Umut Can" list does not mention which platforms to support, whereas the "Loomera" list (#16) sets a specific target: "**iOS 16+ and Android 13+**."
- **Scope and Omissions:** The "Loomera" list includes critical functions that were never mentioned by the client (Umut Can) but are vital for the system, such as a "**reset password**" feature (#2) and "**search functionality**" (#3), which is critical for Umut Can's 5000 products.

Conclusion: This experiment demonstrates that AI (Gemini) tools are extremely powerful for categorizing and rapidly analyzing raw data from client meetings. However, these tools do not replace the human analyst who initiates the process; rather, they act as an assistant that boosts the analyst's efficiency. Taking the raw list produced by the AI and transforming it into a measurable, complete, and testable final SRS document, like the "Loomera" list, is a critical step that still requires the expertise and diligence of a software engineer.