

Requirement Extraction and Analysis

Table of contents

1 Requirement Extraction and Analysis	1
---------------------------------------	---

1 Requirement Extraction and Analysis

This lab focuses on understanding the Quackstagram project in greater detail. You will analyze the project specification, source code, and stakeholder conversations to extract and categorize functional and non-functional requirements. This process is important for shaping the project's development trajectory and ensuring that the final product aligns with stakeholder expectations.

Lab Duration: Approximately 2 hours

Learning Objectives:

- Extract functional and non-functional requirements from the project specification, source code, and stakeholder conversations.
- Learn to differentiate between functional and non-functional requirements and understand their impact on the project.
- Begin the process of requirement prioritization to guide future development efforts.

Preparation:

1. Review the Quackstagram project specification ¹.
2. Familiarize yourself with the existing Quackstagram source code ².

Lab Tasks:

¹You can find the project specification on Canvas or on <https://bcs1430.ashish.nl/assignment/>

²You can find the project source code on Canvas or <https://gitlab.maastrichtuniversity.nl/bcs1430/quackstagram>

1. Understanding Project Specification and Source Code:

- Carefully read the project specification to understand the envisioned system and its functionalities.
- Explore the Quackstagram source code provided to get a feel for the existing implementation and identify areas that might need modification or enhancement.

2. Stakeholder Conversations:

- Review the transcripts of conversations with four stakeholders. Identify and note their specific needs, concerns, and expectations. You can find the transcript on Canvas or alternatively by clicking [here](#).
- Understand the different perspectives and requirements each stakeholder brings to the project.

3. Extracting Requirements:

- From the project specification, source code review, and stakeholder conversations, create a comprehensive list of functional and non-functional requirements.
- Functional requirements should detail what the system should do.
- Non-functional requirements should focus on system properties.

4. Requirement Prioritization:

- Discuss and decide on the importance of each requirement with your team. Consider factors like stakeholder importance, implementation complexity, and project timeline.
- Categorize requirements into ‘Must Have’, ‘Should Have’, ‘Could Have’, and ‘Won’t Have’ (MoSCoW method) or any other prioritization technique you prefer.

Post-Lab Reflection:

- Consider how the requirements might influence your approach in the upcoming refactoring and design changes.

End of Lab:

- Compile the finalized list of requirements and their priorities. This will be a part of your project submission.
- Prepare any questions or clarifications you might need from stakeholders, you have access to two of the stakeholders for any clarifications (Spriha and Ashish).

Preparation for Next Week:

- Begin thinking about how the existing system might be refactored to accommodate the identified requirements.
- Consider the classes and their responsibilities and how tweaking them might be useful in the next stages of the project.