Color coding

- Output instructions:
  - format (i.e. length, paragraph, bullet-points, specificity, etc.),
  - content components
- Project context
- Role assignment
- Course context/definitions: Definitions, Required features/components, Project expectations/scope
- Examples

Prompt 1 – Role Assignment and Course Project Scope:

Hi ChatGPT! You are an expert software engineer with extensive experience in the requirements stage of software development, particularly for mobile applications. You will aid in requirements synthesis and elicitation for an Android mobile application. More specifically, you will help in identifying actors, functional requirements, and non-functional requirements from an application idea. Please consider the time, expertise, and technical constraints provided below.

The development of the application will be carried out by a group of 4 students of a software engineering course over the course of 3 months. The project is intended to be a client-server software system with the client (front-end) running on a mobile device and the server (back-end) running in the cloud. It must include the following four components:

1. Use of an external authentication service, e.g., Google or Facebook authentication.
2. Use of at least one additional external service accessible via an API, e.g., Google calendar, Google maps, etc.
3. Real-time updates, e.g., multi-user chats, push notification, etc.
4. Non-trivial logic in either front-end or back-end component (i.e., something that involves an algorithm, not a simple API call or database read/update/delete).

I will provide you with my application idea in the next prompt.

Prompt 2 – Project Context:

The idea for my mobile application is a <FULL TITLE>, called <NAME>.

The target audience and main problem the idea addresses are as follows:
<TARGET AUDIENCE AND MAIN PROBLEM>

The main actors of the application are: <MAIN ACTORS>

The main use cases of the application are:
<MAIN USE CASES>

If the explanation is clear, please respond with a 2-3 sentence description of the project. If not, please identify what is unclear.

Prompt 3 – Actors:
Let's start with the main actors: <ACTORS>.
<ACTORS DEFINITION>
Please write a **description of each actor** in 1-2 sentences.

Prompt 4 – FR names and descriptions:
Let's move on to define the functional requirements.
<FR DEFINITION>

Please produce 6 non-trivial **functional requirements** for my application idea. Keep in mind the four required components the project must satisfy:
1. At least one functional requirement uses an external authentication service
2. At least one functional requirement uses one additional external service
3. At least one functional requirement incorporates real-time updates
4. All functional requirements require non-trivial logic

Provide the name (denoting a use-case and describing an action performed by the user), a 1-sentence description, and the primary actors for each functional requirement.

Prompt 5-26 – FR specification:

- Let's consider one functional requirement at a time.
  <SS AND FS DEFINITIONS>
  For functional requirement <#> (<FR NAME>), please list the **success scenario**(s) and **failure scenario(s)**. Each failure scenario must correspond to a numbered success scenario item. The list of failure scenarios does not have to start at 1.

- Recall that actors are people, devices, other systems, external services or APIs that interact with the system and are external to the system itself.
  For functional requirement <#> (<FR NAME>), what are the **primary actor(s)**?
  If additional actors need to be defined, please explain why. You may not put the "System" as a primary actor. Any external service or API should also be added as a primary actor. Any complex algorithm is internal to the system and is not to be included as a primary actor.

- Let's put it all together for functional requirement <#> (<FR NAME>). Format it like this:
  - The name of the requirement
  - A short description of the requirement (1-2 sentences)
  - Primary actor(s)
  - Success scenario(s)
  - Failure scenario(s)
  Please do not remove any success or failure scenarios and maintain the exact level of detail as provided before. It is sufficient to copy the previous success and failure scenarios.

Prompt 27 – All FRs
Here are the 6 non-trivial functional requirements we arrived at. Please confirm that you understand the functional requirements.
<FRs>

Prompt 28 – Actors updated

- Next, let's reconstruct the list of main actors. The main actors are a list of actors that are primary actors in at least one functional requirement. Do not include actors that do not appear as a primary actor in any of the functional requirements, or "System". Given the updated functional requirements, identify the main actors. Consider each functional requirement one at a time. Then, once you have the main actors, generate a description in 1-2 sentences for each one.

Prompt 29 – Use case diagram

Next, create a UML **use-case diagram** as a LaTeX graph. First, please retrieve and output the title and primary actors of each updated/modified functional requirement. Then follow the steps sequentially:

1. Create a node for each main actor.
2. Create a node for each functional requirement. Please use the title of the functional requirement as its identifier. Remember, the title denotes a use case.
3. For each functional requirement, add a connection between the functional requirement and each of its primary actors. Go step by step.
4. Add any additional use case diagram relationships if necessary.

Prompt 30 – NFRs

Let's consider the non-functional requirements next.

<NON-FUNCTIONAL REQUIREMENTS DEFINITION>

Please provide 3 non-trivial **non-functional requirements** for the project as a list. For each requirement, provide the textual description, an explanation for why this requirement is needed/relevant in 1-2 sentences, and how the requirement will be validated in 1-2 sentences. Complete one non-functional requirement before moving on to the next one. Remember that the development of the application will be carried out by a group of 4 students of a software engineering course over the course of 3 months.