# Evaluation Document

Team:

Members:

Instructions to the Team:

- For each of the rows in the rubrics, select which level corresponds best to the current state of the delivery. Comment on each row.
- Propose changes to the rubric where you see need or possibilities for it.
- Ask up to three questions (at the end of the document) that you want to have especially feedback on.

Instructions to the Evaluator:

- For each of the rows in the rubrics, select which level corresponds best. Comment on possible improvements.
- Provide an overall comment for each section (that means, for each rubric.) These should help to improve overall delivery.
- Answer the team's questions.

**Use Cases**

| | **Excellent** | **Good** | **Sufficient** | **Not rateable** |
|---|---|---|---|---|
| Exceptions and Alternatives | Consistent handling of relevant exceptions and alternatives. | Most relevant exceptions and alterntives described. | Some inconsistencies. | |
| Alignment | Consistent with user requirements, vision and system. | Good alignment to the user requirements. | Minor inconsistencies. | Missing alignment. |
| Descriptions | Concise descriptions. | Use cases have good, descriptive names. | | Major language errors and typos. |
| Value | The use cases provide excellent value for the development by addressing relevant issues in a compact form. | The described use cases form a cohesive, focused whole that are a good basis for planning the upcoming development of the system. | Some value for further development, but inconsistent. | |

Own judgements:

- Exceptions and Alternatives: **Good**
- Alignment: **Excellent**
- Descriptions: **Good**, Use Case No. 1 differs a bit in name from the Hierarchy
- Value: **Good**

Evaluator judgments:

- Exceptions and Alternatives: Good. It is not very clear where and when the alternatives and exceptions diverge from the normal flow in UC-1. I am not entirely sure if reserving another scooter is an alternative flow, since the procedure is the same. Maybe I do not understand it entirely.
- Alignment: Excellent. Consistent with what it is defined in the vision and requirements
- Descriptions: Good. Excellent summaries of the objective and act as descriptions, but the names of the use cases and alternatives could be better redacted by using the present ("Unlocking the scooter" ->"Unlock a scooter") or reducing the name ("A person tries to create an account but has entered bad information" -> "Bad information input")
- Value: Good. Relevant use cases and very explicative that will help the development, but maybe other use cases could have been more relevant, like "Lock a scooter" rather than "Account Management". But both are necessary, so it is very nice anyways.

**Deployment diagram**

| | **Excellent** | **Good** | **Sufficient** | **Not rateable** |
|---|---|---|---|---|
| Layout | Layout follows a strategy that helps to understand the diagram. | Layout is structured. | Layout is structured. | Layout is unstructured and random. |
| Syntax | Correct syntax. | Correct Syntax. | Overall good syntax, with a few minor errors. | Major syntactical flaws. |
| Level of Detail | Consistent and intentional level of detail. | Adequate detailing. | Some inconsistencies, too much focus on some details on the expense of others. | |

Own judgements:

- Layout: **Excellent**
- Syntax: **Good**
- Level of Detail: **Good**

Evaluator judgements:

- Layout: Excellent. Very nicely structured, helps a lot for readibility.
- Syntax: Excellent.
- Level of detail: Good. Very detailed, but as the syntax is very good it is understandable, and it will probably help the development a lot.

**Sequence diagram**

| | Excellent | Good | Sufficient | Not rateable |
|---|---|---|---|---|
| Coverage | All relevant scenarios are handled with an appropriate level of detail and clearness. | All relevant scenarios are handled. | Most relevant scenarios are handled. | Important scenarios are missing. |
| Implied Scenarios | All relevant implied scenarios are handled and clarified. | Some implied scenarios are handled. | No implied scenarios are described. | |
| Combined Fragments | Appropriate use of combined fragments where relevant. | | Occasional inappropriate use, that means used where they are not necessary or lack of use where they would help. | |
| Level of Detail | Consistent and intentional level of detail. | Adequate detailing. | Some inconsistencies, too much focus on some details at the expense of others. | |
| Layout | Layout follows a strategy that helps to understand the diagram. | Layout is structured. | Layout is structured. | Layout is unstructured and random. |
| Syntax | Correct syntax. | Correct Syntax. | Overall good syntax, with a few minor errors. | Major syntactical flaws. |

Own judgements:

- Coverage: **Good**
- Implied Scenarios: **Excellent**
- Combined Fragments: **Excellent**
- Level of Detail: **Good**
- Layout: **Excellent**
- Syntax: **Good**

Evaluator judgements:

- Coverage: Excellent. It covers all relevant scenarios and is coherent with the flow defined in the use cases
- Implied Scenarios:  Excellent. Even though I am still not sure about the alternative flow of renting another scooter in UC-1, all defined scenarios in the use cases are then covered in the sequence diagrams.
- Combined Fragments: Good. They are used a lot, which could be confusing. Instead of this, different use cases could be used, for example, in UC-3, instead of 6 conditions.
- Level of Detail: Excellent. Very detailed and descriptive, which is helpful for the development.
- Layout: Excellent

- Syntax: Sufficient. Even though it is very understandable and readable with common language, it could be more useful to put inputs and outputs or information exchanged in a better way. So for example, every time you talk about scooter information, you could put it like this: scooterinfo(id, status, location) and just put relevant information in each exchange.

### State machines

| | Excellent | Good | Sufficient | Not rateable |
|---|---|---|---|---|
| Control States | Appropriate use of control states that helps to make the machine understandable. | Overall good use of control states. | Overall okay use of control states, with some flaws. | Wrong use of control states. (1) |
| Events | Explicit and understandable handling of events by correct use of transitions or */defer*. | All relevant events are handled in all relevant states. | Most relevant events are handled in most states where they matter. | Major events are not handled. |
| Semantics | Machine is consistent and critical situations explained in comments. | Machine is free of design flaws. | Some few minor issues. | There are major design errors. |
| Implementability | Clear how the machine can be implemented in STMPY, ambiguous cases are commented. | Machine is implementable in STMPY. | Machine is apart from a few isolated issues implementable in STMPY. | Machine contains several constructs that are unclear how to implement. |
| Level of Detail | Consistent and intentional level of detail. | Adequate detailing. | Some inconsistencies, too much focus on some details on the expense of others. | |
| Layout | Layout follows a strategy that helps to understand the diagram. | Layout is structured. | Layout is structured. | Layout is unstructured and random. |
| Syntax | Correct syntax. | Correct Syntax. | Overall good syntax, with a few minor errors. | Major syntactical flaws. |

Own judgements:

- Control States: **Good**
- Events: **Good/Excellent**
- Semantics: **Good**
- Implementability: **Excellent**
- Level of Detail: **Good**

- Layout: **Excellent**
- Syntax: **Good**

Evaluator judgements:

- Control States: **Excellent**
- Events: **Good.** Some events could be better handled. For example, it could be confusing why some events are entries and some are triggers instead of exits, or how some triggers could happen.
- Semantics: **Good**
- Implementability: **Excellent.** It is very clear how the machine could be implemented and would work. It is also very intuitive and simple.
- Level of Detail: **Excellent.**
- Layout: **Excellent**
- Syntax: **Good**. All perfect except for the same problem I wrote in "Events"

**Overall delivery**

| | Excellent | Good | Sufficient | Not rateable |
|---|---|---|---|---|
| Complexity and scope | The chosen scope for the system is appropriate. | | | System is too simple to apply learned material. |
| Consistency | All parts of the delivery are consistent. | Overall good consistency. | Some minor errors with the consistency. | Major inconsistencies between the parts. |
| Report | Show clear signs of organization and consistency, comments where appropriate. | Generally consistent and organized. | Complete report following the rules for deliveries. | Major flaws in layout and structure. |

Own judgements:

- Complexity and Scope: **Excellent**
- Consistency: **Good**
- Report: **Good**

Evaluator judgements:

- Complexity and Scope: **Excellent.** Everything perfect. It is true that maybe the use cases could have been more relevant and managing the account doesn't seem as important as payments or locking scooters, but it is overall still important.
- Consistency: **Good.** Use cases could be a bit more consistent, as it seems the alternative flows in UC-1 are not written as the rest, for example.
- Report: **Excellent.**

## Questions

Team: Ask **up to three** questions about aspects of the delivery you especially want to have feedback for.

Evaluator: Please answer the questions.