Group 7:
Review report of requirements specification of Group 3

Introduction:

Before we had a look at the document, we created a general checklist by considering the contents of the lecture and the course book.

The first three main sections we took from the template to cover/evaluate the overall aspects of the document. The other two sections were inspired by the last lecture (e.g. CRUD check) and the book. However, our checklist differs from the template in the book to better adapt to the nature of the project.

Every member of the group filled his own checklist and afterwards we discussed and merged the results. The finalized table is on the pages below. Every observation is prioritized using different colors to give an overview and a distinction between the feedback.

Review summary:

The high-level description was pretty neat, maybe a little bit too low-level for example at the use case diagram. There are some little inconsistencies in our opinion, e.g. regarding the different stakeholder diagrams and the reasoning could be better at the quality grid.

Discussing the data requirements we think that the ER diagram is problematic in this section because it is too general and there are inconsistencies between the ER diagram and the data dictionary. The functional requirements are fine, we just found minor aspects which could be improved. It is nice that you included the stakeholders in the prioritization process, but it would be nice to have a final prioritization.

In the system requirements we like the work areas. The UI prototype fulfils its purpose but we did not find any descriptions how the requirements will be implemented.

The tracing can be found throughout the whole document, but a summarizing diagram would be a good addition. The CRUD check found one missing part for the entity contract.

Finally, we did a structure check like in the slides of the last lecture. In general it was good, we only found some minor points.

Overall, we really like the document, especially the first chapter which contains many nice diagrams. We had the feeling that you paid much attention to the stakeholders and also considered a lot of them in several directions.

Text	Positive feedback
Text	Minor critique / Suggestions
Text	Critique
Criteria	**Observations**
1. High-level description	
Goal	- Very good goal with a neat background, clearly displaying a gap in the market.
Scope	- Very detailed context diagram and clear scope.
Business goals	- Solution vision in BG1 is a bit to “low level” - Maybe propose a business model (“how to make money from it”, etc.) - Mismatch between business goal description and table (regarding BG1).
Stakeholders	- Nice thinking about potential business partners. A lot of information, but well displayed which makes it a non-issue. - Figure A.2 has stakeholders which are missing in figure A.3 without any real explanation to why, which kind of makes the stakeholders identified in figure A.2 and not included in figure A.3 not important at all.
Core functionality	- Text is well written - Use-case diagrams should be on a higher level in our opinion. - Using boxes to note actors, UML notation does not support this in our opinion
Quality grid	- Making interoperability both critical and unimportant felt forced. - The reasoning behind the reliability/availability is too general and applicable to almost all systems. - Is efficiency kind of important? Even listed in PLanguage as to be very fast. Seems like it's important since a main use-case is probably to filter accommodations based on various preferences, which will be computationally heavy we assume.
PLanguage	- The aim is very high. Having a must of 100% of the time, consider perhaps a lower must, but you could keep the wish.
2. User requirements specification	
Data requirements	- ER diagram is very general. - Mismatch between attributes in ER diagram and data dictionaries. - Missing attributes in the ER diagram and data dictionary e.g. contract has no file. The option to extend the contract is missing. The contract being a “living thing” has to be open for changes.

	- Address attributes should be more divided into street, postal code, city, making it more atomic. - Datatype of ID's shouldn't be char<length>. Consider String or Integer. - Make current education an entity with data types such as grad year, msc/bsc/phd and science/economics/ etc. - Photos shouldn't be datatype "media", but instead an entity. In order to assign multiple photos to a single entity consider creating a photo entity with a relationship 1-n.
Functional requirements	- "Also part of one of the business goals." too often in the requirements. Consider using links to business goals instead. Also: What about the other requirements? Are they not important regarding the business goals? - Nice links to the work areas and the user stories. - Some functional requirements are not atomic, e.g. FR.01. Consider splitting it up into one requirement where you sign up, one where you allow for editing and one for deleting accounts. This would make one requirement always address one "issue", which would be nice. - Not divided into goal, domain, product and design.
Prioritization	- The two methods used are good. Nice that the stakeholders were included to prioritize the "features" themselves. - In the text the prioritization is motivated in order to prioritize and budget for features while the methods only addresses what is most appreciated and "fun". The result would indicate that account management is lowly prioritized while it's still an essential part of the system. - In this part developers are taken into account without a motivation to why they are specifically targeted instead of others also less mentioned in general. - "15" in the Homeowners column (100$ test) should not be fat… :-) - There exists no summary of the result from the prioritization / final prioritization.
3. System requirements	
System requirements / Further specified requirements	- We really liked the work areas. Nice additional information. - There is no information regarding the mapping system which api to use, develop your own etc. - There is no description regarding how the requirements will be implemented, it is only implied. For example, for signing up you use a form, for search of accomodations you use a listview, etc.
UI prototype	- We understand that you propose a web based prototype. However, there is no information for other platforms. Is the application only meant to be a web application? - Maps well to the work areas

4. Consistency check	
Evidence of tracing	- Nice mapping from functional requirements to work areas and from business goals to functional requirements. - Changes would cause the editor to have to look in multiple places as the “related requirements/tasks” are in multiple places. One summarized tracing diagram would be neat.
CRUD check	- CRUD for contracts is not sufficient. No specification on how a contract can be cancelled or ended. - CRUD for account management and accommodation is good.
5. Structure check	
ID for each requirement	- It’s clear and scalable.
Verifiable requirements	- They are verifiable. Sometimes a bit vague such as “popular language”: what is a popular language?
Purpose for each requirement	- The purpose of the functional requirements is only implied, but sufficient in our opinion.
Examples of ways to meet requirement	- In general good. Specified task descriptions help at understanding, but not sufficient for the developers. - It would be nice, if the system requirements section would include further details how the respective requirements would be implemented, e.g. “The system must allow for users to sign up using a form”.
Plain-text explanation of diagrams	- Every diagram has a clear description in the text. - Some tables are missing a reference.
Importance and stability for each requirement	- Importance is deduced by the prioritization.
Cross refs rather than duplicate information	- There is usually good cross referencing. There are however some instances of duplication of information such as the detailed data requirements in relation to data dictionaries.