Lab 1

1. Data dictionary
2. Functional and non-functional requirements
3. Use Case diagram
4. Use Case descriptions

Data dictionary

1.Focus on problem domain instead of implementation
2. Do not explain too simple nouns.

Functional and non-functional requirements

1. The requirements are organized too casually.

   Imagine your finalized application and simulate each step of how the user and system are organized. Clarify the main function(use case) and explain it gradually, leading to a hierarchical requirements.

2. Requirement should be **Atomic** – do not combine multiple functions in one requirement.

3. Use Active Voice

4. Complete, Verifiable, Unambiguous, so Do not use e.g. or some of ……

Use Case diagram

1. Name of the Use Case: Verb-Noun
2. Direction of association arrow

   A includes B: from A to B

   B Extends A: from B to A
3. Pay attention to the relation between use cases

   1) Build the hierarchy based on the inclusion relationship between features, rather than their order.

   1) **include** means the function that highly relevant to its upper use case

   2) the upper use case is complete by itself, but can be **extend**ed in usual situation
4. Organize the Use Case Diagram neatly

Use Case descriptions

1. **Align the Diagram and Description!**

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Preconditions:

State of the System before conduct current use case.

Note the Differences from special requirement and assumption.

User is authenticated. / The filter condition is not empty …

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Postconditions:

State of the System After conduct current use case

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Flow of Events:

1.The Includes and extends should be mentioned in the flow. You can Skip the specific content of these use cases and just specify the necessary content like Condition.

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Alternative Flows:

1.Write clearly from which step AF ? (of the main flow) to enter the alternative flow.

2.Specify the condition

3.End with Go back to Step N.

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Exceptions:

Flows that lead to a failure of the use case's function.

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Includes/Extends:

Specify the included(Extended) use case, and replace the corresponding flow with something “go to use case A and ……”

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Special Requirements:

A special requirement is typically a **non-functional requirement** that is specific to a use case but which is not easily or naturally specified in the text of the use case's event flow. These special requirements may be regulatory requirements, application standards, performance standards or design constraints.

Same standards with (non) functional requirements:

verifiable/atomic/unambiguous

Use Case descriptions

Actor:
Description:
Preconditions:
Postconditions:
Priority:
Frequency of Use:
Flow of Events:
Alternative Flows:
Exceptions:
Includes:
Special Requirements:
Assumptions:
Notes and Issues:

Assumptions:

some additional conditions that are outside the scope of the function of our system.