Software Engineering

Assignment 2

Name : ABID ALI

Roll : 2019380141

Send to : lining@nwpu.edu.cn , 480755534@qq.com

Assignment2 Submission Deadline: 29 Feb -> March 6

For your planned software, do the following requirement analysis

1) Function requirement list;
2) Non-function quality requirement list;
3) Usecase diagram;
4) State diagram
5) Activity diagram

UML Tool: http://www.umlet.com/ or other tools

Functional requirements list:

Functional requirements describe ways a product must behave. Such requirements describe system behavior under specific conditions and include the product features and functions which web & app developers must add to the solution. Such requirements should be precise both for the development team and stakeholders. The list of examples of functional requirements includes:

- Business Rules
- Transaction corrections, adjustments, and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements

- Reporting Requirements
- Historical Data

If your team uses Agile methodology, they will design most of the requirements in written form. Still, to present some requirements more clearly, the team can visualize them.

Non-functional requirements:

The definition of non-functional requirements is quality attributes that describe ways your product should behave. The list of basic non-functional requirements includes:

Usability

Usability is the degree of ease with which the user will interact with your products to achieve required goals effectively and efficiently.

Legal or Regulatory Requirements

Legal or regulatory requirements describe product adherence to laws. If your product violates these regulations, it may result in legal punishment, including federal fines.

For example, look at legal requirements for our recent project, a mobile taxi platform:

"*To operate in London, the platform should be licensed by the local transport authority - Transport for London*."

Reliability

Such a metric shows the possibility of your solution to fail. To achieve high reliability, your team should eliminate all bugs that may influence the code safety and issues with system components.

Performance

Performance describes how your solution behaves when users interact with it in various scenarios. Poor performance may lead to a negative user experience and jeopardize system safety.

Example:

"*The application shows cars nearby for three seconds.*"

As we said, some non-functional requirements are not so distinct and might be missed by the team and stakeholders due to:

Subjective nature: Different users can view, interpret, and evaluate Nonfunctional characteristics in different ways.

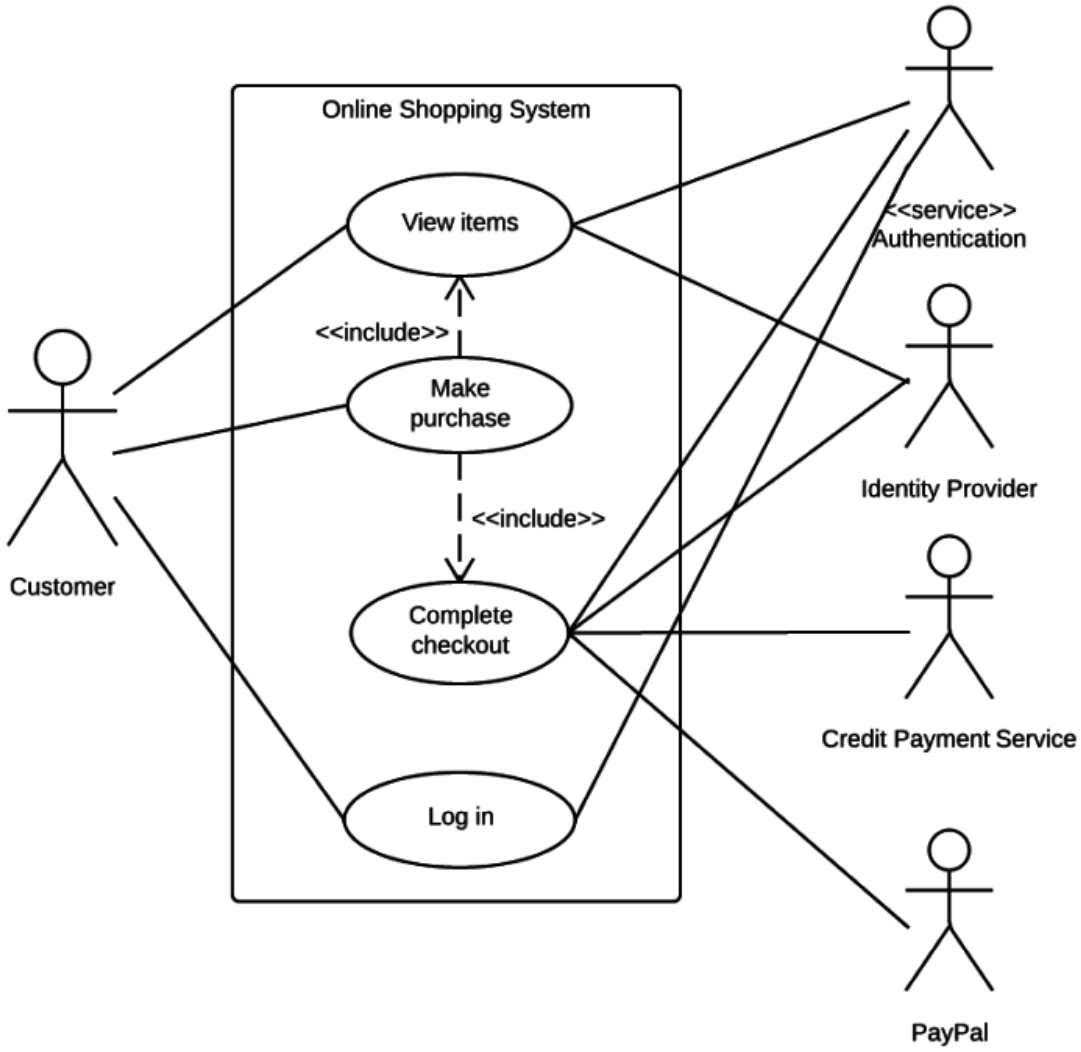
Integrated nature: The goals of one non-functional requirement may conflict with another since they typically have a broad effect on systems.

Don't know what they [NFRs] are: Unclear terminology, confusing definitions, and the absence of a universally accepted classification scheme make understanding of non-functional requirements a challenge.

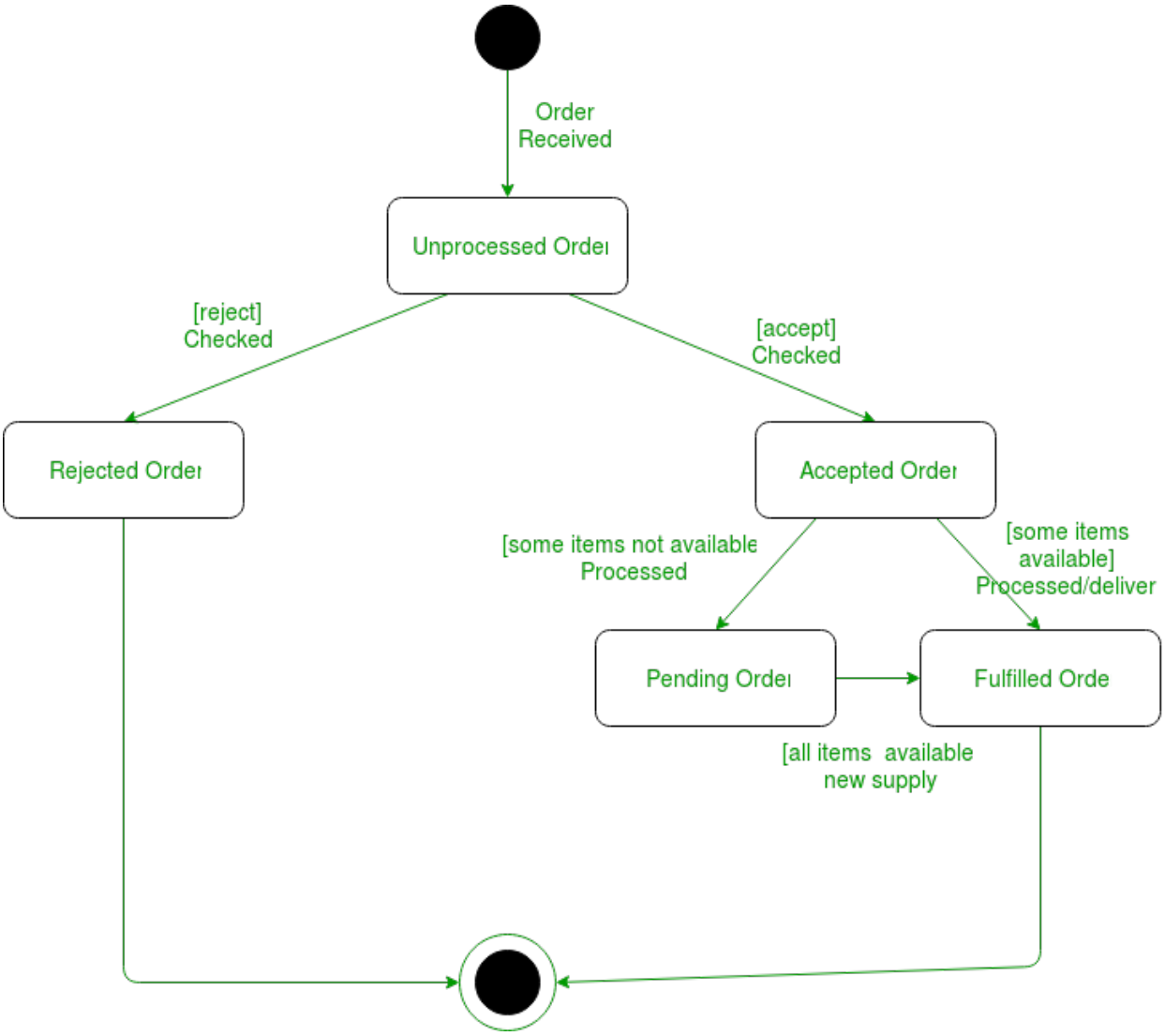
Assuming that "Everybody knows.": During the discovery (inception) phase, both the client and the team might forget about some non-functional requirements because some of them are hard to define from the perspective of a business idea. Therefore, they might arise only after the project release.

Still, non-functional requirements mark the difference between a development project's success and its failure.

UML Use Case Diagram:



State diagram:



Activity diagram: