

“If it can’t do, I don’t want it.”

“It must be able to”

“I want a system that will”

Requirements

Often stated as User Stories

	A	B	C	D
1	Agile User Story Template			
2				
3				
4				
5	User Story ID	As a <type of user>	I want to <perform some task>	so that I can <achieve some goal>
6	1	Project manager	View a status report from each team member	Ensure the project stays on track.
7	2	Employee	Be reminded of upcoming deadlines	Complete my tasks on time.
8	3	Director	See the big picture view of department work	Stay in the loop.
9				
10				



- Two Kinds of Requirements
 - Functional = WHAT
 - Non-Functional = HOW

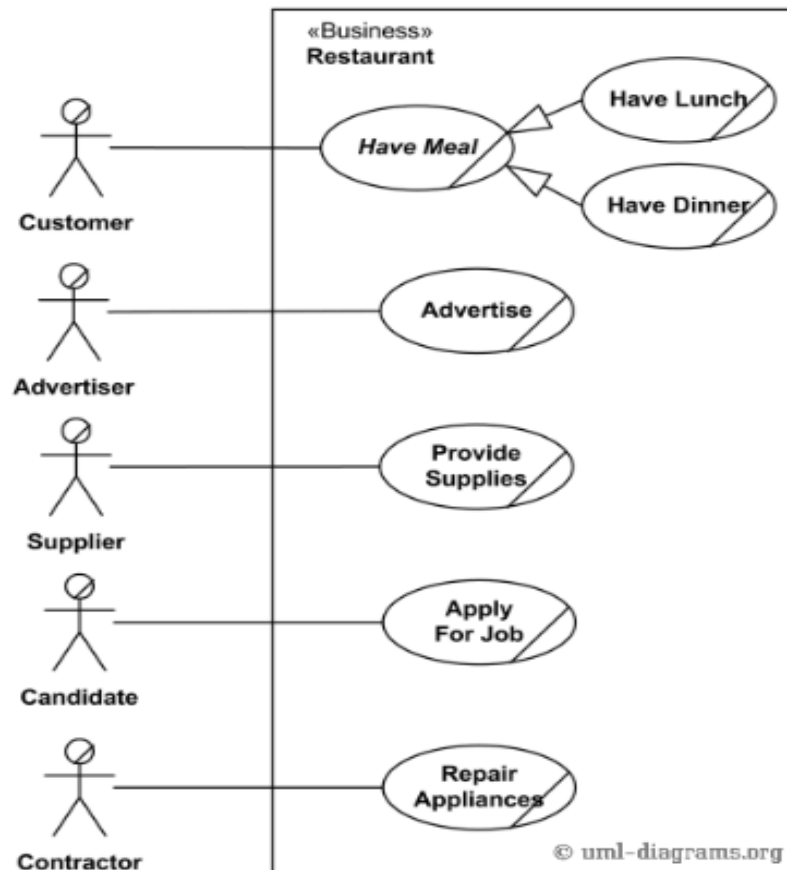
- Functional Requirements
 - User Requirements, Business Requirements
 - Documented by Use Case
 - The sum of the Use Cases = “User Requirements”
 - These requirements describe the users’ expectations for what the system does for them
 - “WHAT I want the system to do for our organization”
 - A functional requirement for an everyday object like a cup would be: “ability to contain tea or coffee without leaking”.
- The definition of a functional requirement is:
 - “Any requirement which specifies **what** the system should do.”

Restaurant

UML Use Case Diagram Example

Here we provide two alternative examples of a **business use case** diagram for a Restaurant, rendered in a notation used by **Rational Unified Process** (RUP).

First example shows **external business view** of a restaurant. We can see several **business actors** having some needs and goals as related to the restaurant and **business use cases** expressing expectations of the actors from the business.



Business use case diagram for Restaurant - External view

For example, **Customer** wants to **Have Meal**, **Candidate** - to **Apply for Job**, and **Contractor** - to fix some appliances. Note, that we don't have such actors as **Chef** or **Waiter**. They are not external roles but part of the business we model - the Restaurant, thus - they are not actors. In terms of RUP Chef and Waiter are **business workers**.

- **Actor** – The person or people who will perform the steps of this use case.
- **Preconditions** – A description of the system prior to the use case starting.
- **Normal course** – A description of the use case itself. This description can either be in narrative form, or a numbered list of specific user steps. When a use case (such as “User approves/rejects customer requests”) has more than one way that a user can accomplish the needed steps, the most common way is shown here – only a single path is shown.
- **Alternate courses** – Descriptions of alternatives to, or deviations from the normal course. For example, the most common course might be to view the oldest unaddressed customer requests. An alternate course may be to view the unaddressed requests from the largest customers.
- **Exception courses** – Descriptions of what the user will experience when something goes wrong.
- **Post-conditions** – Description of the affected portions of the system after the use case has completed.
- **Frequency of use** – An estimate of how often a particular use case will be exercised.
- **Assumptions** – Any assumptions that are implicit in the definition of the use case.

EXAMPLE: https://en.wikipedia.org/wiki/Use_case

- Non- Functional Requirements
 - Ties Use Cases to Technical Implementation
 - Describes HOW the system should behave
 - A non-functional requirement for the cup mentioned previously would be: “contain hot liquid without heating up to more than 45 ° C”.
 - These requirements impact the users’ experience
 - For example, “The home page must load within 1.4 seconds.”
 - Or, “If a document is flagged as private, only the user who created it can see it.”
- The definition of a non-functional requirement is:
 - “ Any requirement which specifies **how** the system performs a certain function.”

- **Interface requirements**
 - Field 1 accepts numeric data entry.
 - Field 2 only accepts dates before the current date.
 - Screen 1 can print on-screen data to the printer.
- **Regulatory/Compliance Requirements**
 - The database will have a functional audit trail.
 - The system will limit access to authorized users.