

Week 7

CONGRATS

- Iteration 2 is super chonky and huge, big workload
- You guys did well!
- Holidays only a few weeks away :D

This tutorial there will be a lot of group work (tute might end early)

- *I'll come around while on mute to mark attendance*

Demos at the end

- *Get the frontend ready to demo as well if you can + gitlab to show code*
- *Get ready to screenshare*

Functional v Non-Functional

Functional requirements specify a specific capability/service that the system should provide. It's *what* the system does.

Non-functional requirements place a constraint on *how* the system can achieve that. Typically this is a performance characteristic.

Great reading on the topic

- Functional requirements: what a system should do, what it should input or output, what does something do
- Think of a car

Non functional requirements

- Bit more on how it works, constraints

Functional v Non-Functional

For example:

Functional: The system must send a notification to all users whenever there is a new post, or someone comments on an existing post

Non-functional: The system must send emails no later than 30 minutes after from such an activity

User Stories - Structure

ROLE

GOAL

BENEFIT

When a customer tells you what they want, try and express it in the form **As a < type of user >, I want < some goal > so that < some reason >**

Important!

We use this to know what to develop!

Eg. apps for hackathons, COMP3900

E.G. They say:

- E.G. They say:
 - A student can purchase monthly parking passes online
- But your story becomes:
 - As a **student**, I want to **purchase a parking pass** so that **I can drive to school**

User Acceptance Criteria

- Break down a user story into criteria that must be met for the user, or customer, to accept
- Written in natural language
- Can be refined before implementation

Shouldn't be too complicated and talk about backend stuff, it should be understandable by the client

When writing these, imaging you are a client navigating a website (e.g. moodle)

RULE BASED AC

Example

As a user, I want to use a search field to type a city, name, or street, so that I can find matching hotel options.

- The search field is placed on the top bar
- Search starts once the user clicks "Search"
- The field contains a placeholder with a grey-colored text: "Where are you going?"
- The placeholder disappears once the user starts typing
- Search is performed if a user types in a city, hotel name, street, or all combined
- The user can't type more than 200 symbols

Example 3:

SCENARIO BASED AC

As a user, I want to be able to recover the password to my account, so that I will be able to access my account in case I forgot the password.

Scenario: Forgot password

Given: The user has navigated to the login page

When: The user selected forgot password option

And: Entered a valid email to receive a link for password recovery

Then: The system sent the link to the entered email

Given: The user received the link via the email

When: The user navigated through the link received in the email

Then: The system enables the user to set a new password

Which one to use?

- Rule-based acceptance criteria are simpler and generally work for all sorts of stories
- Scenario-based AC work for stories that imply specific user actions, but don't work for higher-level system properties (e.g. design)
- Scenario-based AC are more likely to be implementable as tests

Use cases

- Represent a *dialogue* between the user and the system, with the aim of helping the user achieve a business goal
- The user initiates *actions* and the system responds with *reactions*
- They consider systems as a black box, and are only focused on high level understanding of flow

Use-case (Background)

- **Use Case:** Withdraw Money
- **Goal in Context:** Customers need to withdraw money from their accounts without entering the bank
- **Scope:** ATM, banking infrastructure
- **Level:** Primary Task
- **Preconditions:** The customer has an account with the bank
- **Success End Condition:** The customer has the money they needed to withdraw
- **Failed End Condition:** The customer has no money
- **Primary Actor:** Customer
- **Trigger:** Customer puts card into ATM

Template for background

- **Use Case:** <the name should be the goal as a short active verb phrase>
- **Goal in Context:** <a longer statement of the goal, if needed>
- **Scope:** <what system is being considered black-box under design>
- **Level:** <one of: Summary, Primary task, Subfunction>
- **Preconditions:** <what we expect is already the state of the world>
- **Success End Condition:** <the state of the world upon successful completion>
- **Failed End Condition:** <the state of the world if goal abandoned>
- **Primary Actor:** <a role name for the primary actor, or description>
- **Trigger:** <the action upon the system that starts the use case, may be time event>

