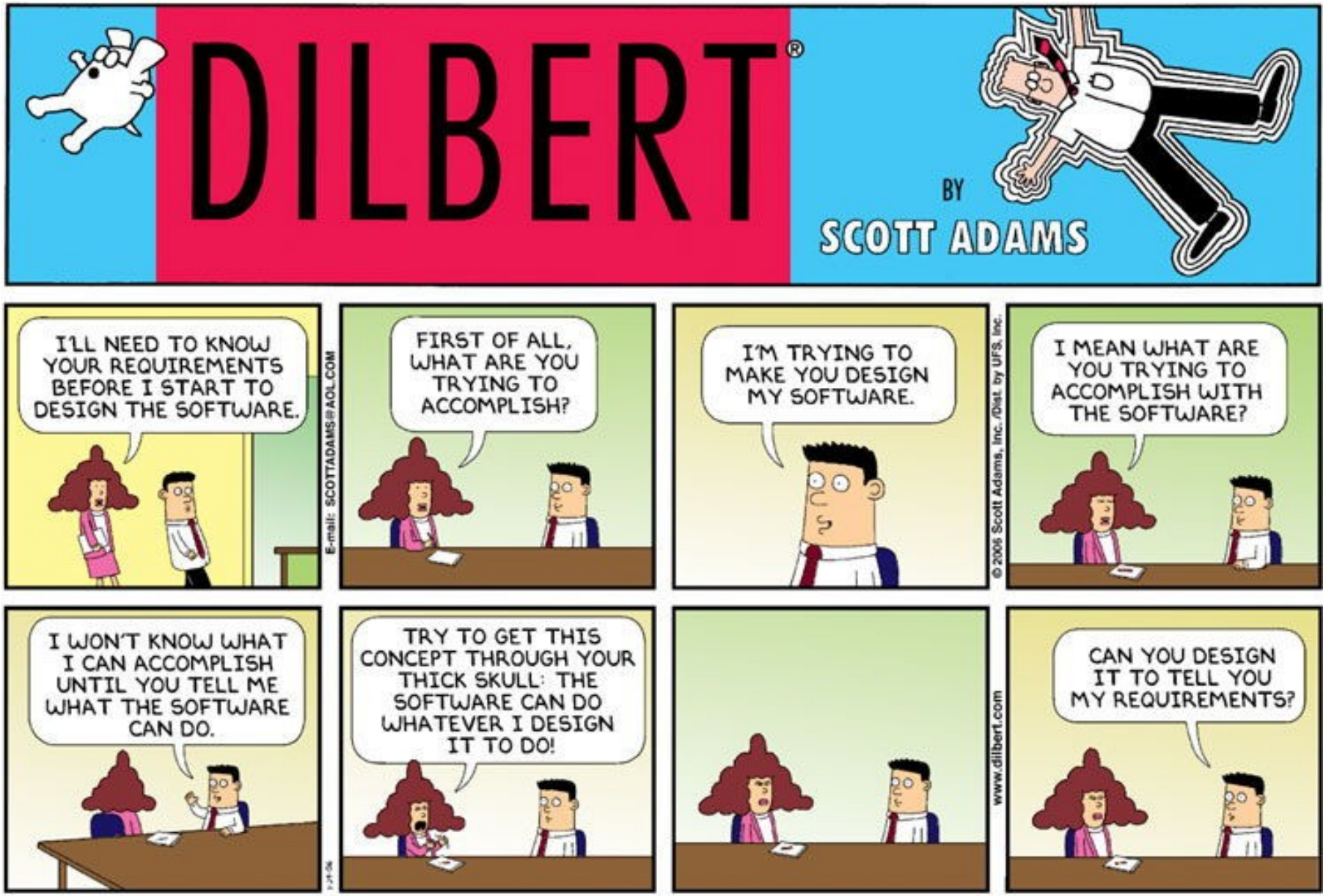


CPEN 321

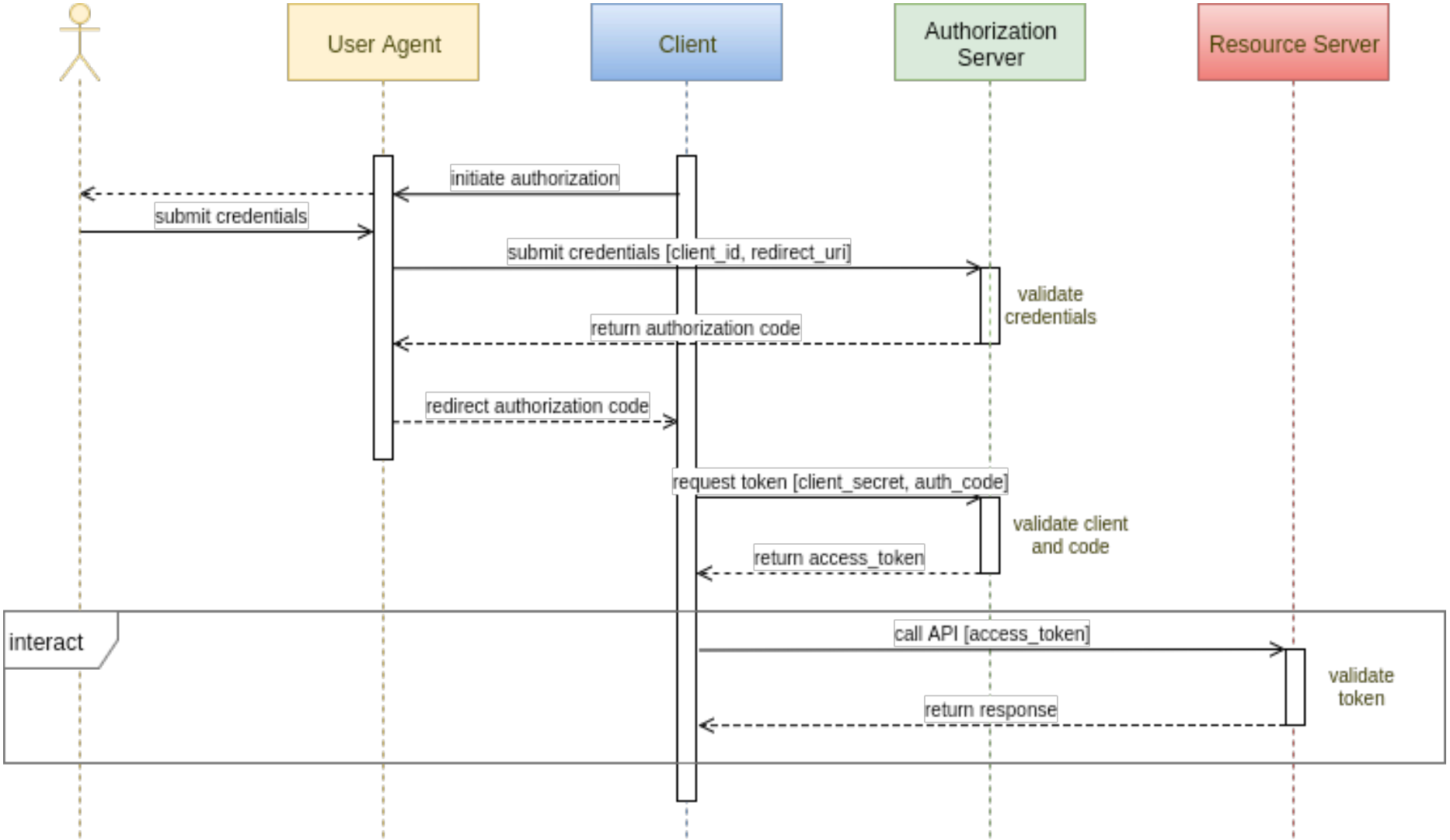
Requirement Engineering,
User Stories, Use Cases
+ *some logistics*

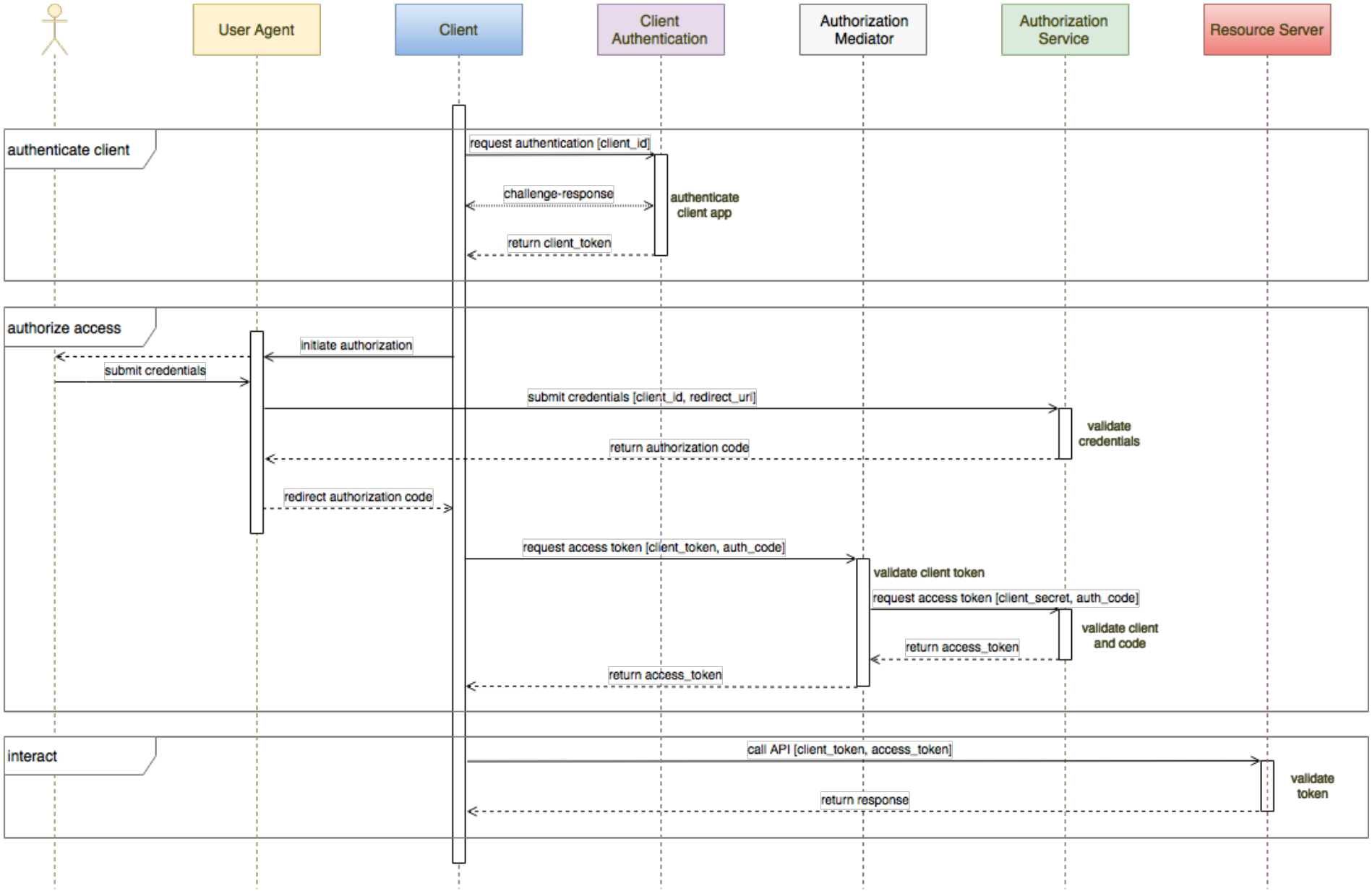
Outline

- UML (summary)
- Requirements
  - What are requirements?
  - How do we gather requirements?
  - How do we document requirements?
- Logistics

UML - Summary

- Many diagrams
- One might debate on how often UML is used in practice
  - Answer: Some diagrams are used more widely than others:
    - Simplified class diagrams
    - Activity diagrams (flowcharts)
    - Sequence diagrams
    - State machines (for full code generation, e.g., with IBM Rhapsody)
    - …
- <u>Main benefits</u>
  - Accurately specify design aspects to consider
  - Provide a standard language of communication


User Agent
Client
Authorization Server
Resource Server
initiate authorization
submit credentials
submit credentials [client_id, redirect_uri]
validate credentials
return authorization code
redirect authorization code
request token [client_secret, auth_code]
validate client and code
return access_token
interact
call API [access_token]
validate token
return response



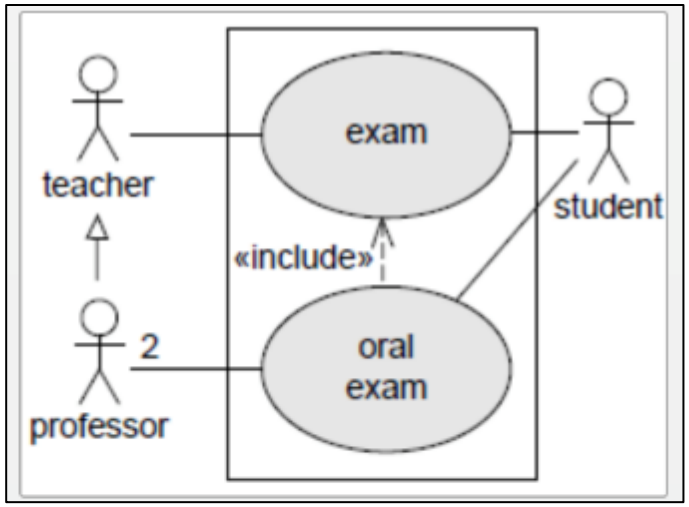
https://hackernoon.com/mobile-api-security-techniques-part-3-1e1e092aeacd

Quick Quiz

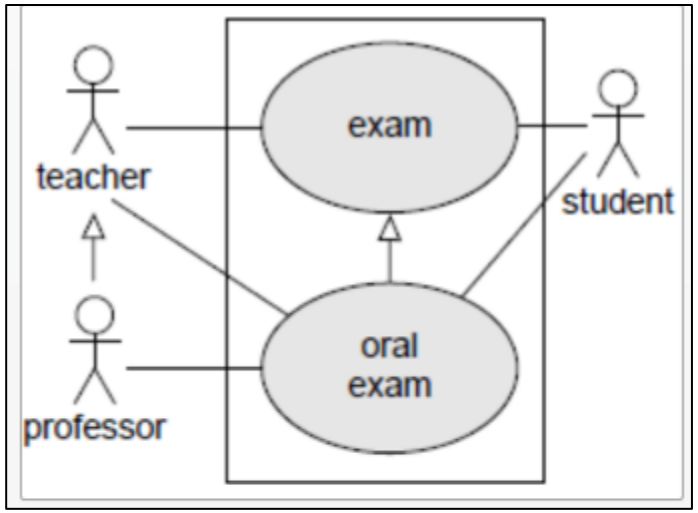
Which use case diagram describes this situation:

For an exam, a student and a teacher need to be present. If it is an oral exam, a professor has to act as second examiner.

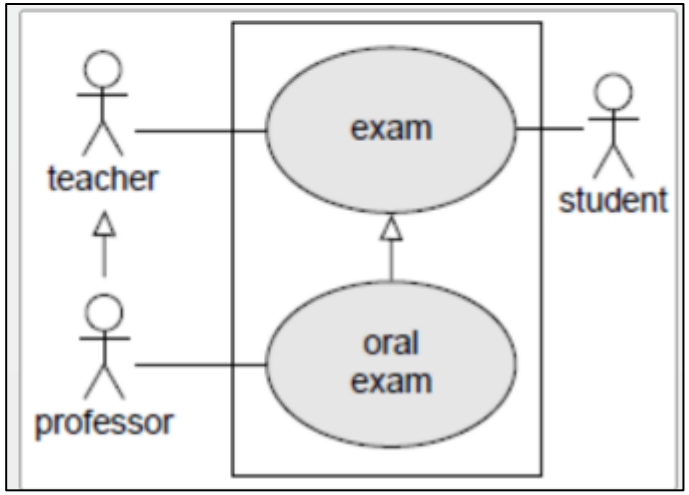
A



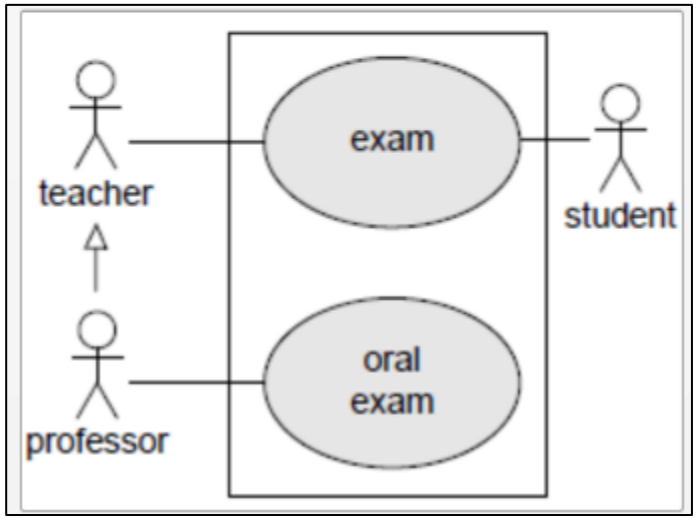
C



B



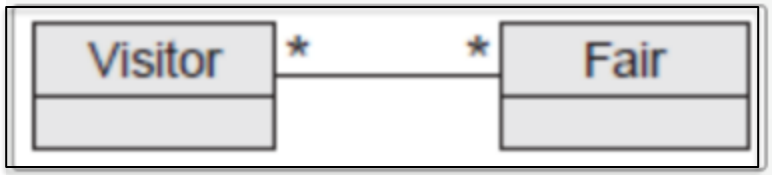
D



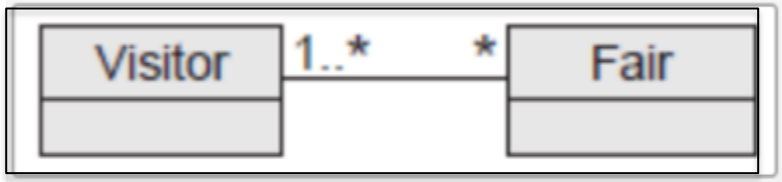
Which class diagram describes this situation:

A fair is visited by at least one visitor and each visitor has to visit at least one fair.

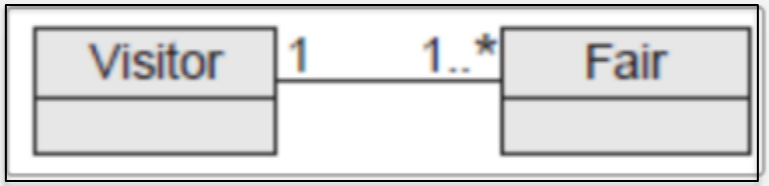
A



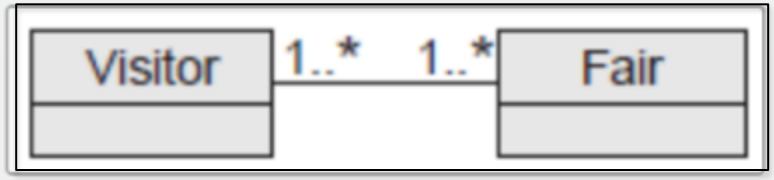
B



C



D



Outline

- UML (summary)
- **Requirements**
  - What are requirements?
  - How do we gather requirements?
  - How do we document requirements?
- Logistics

Software Requirements

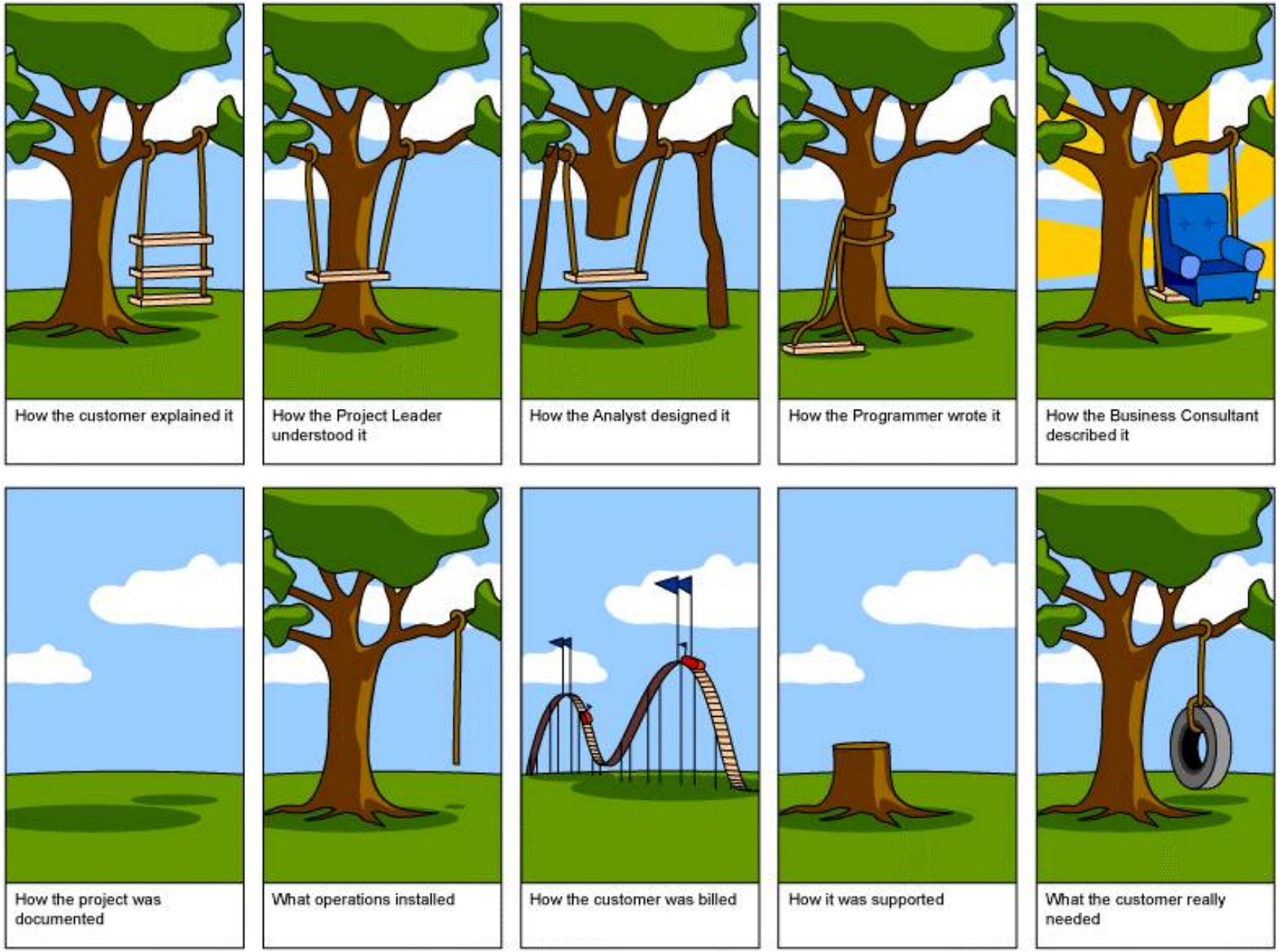
Requirements specify what to build:

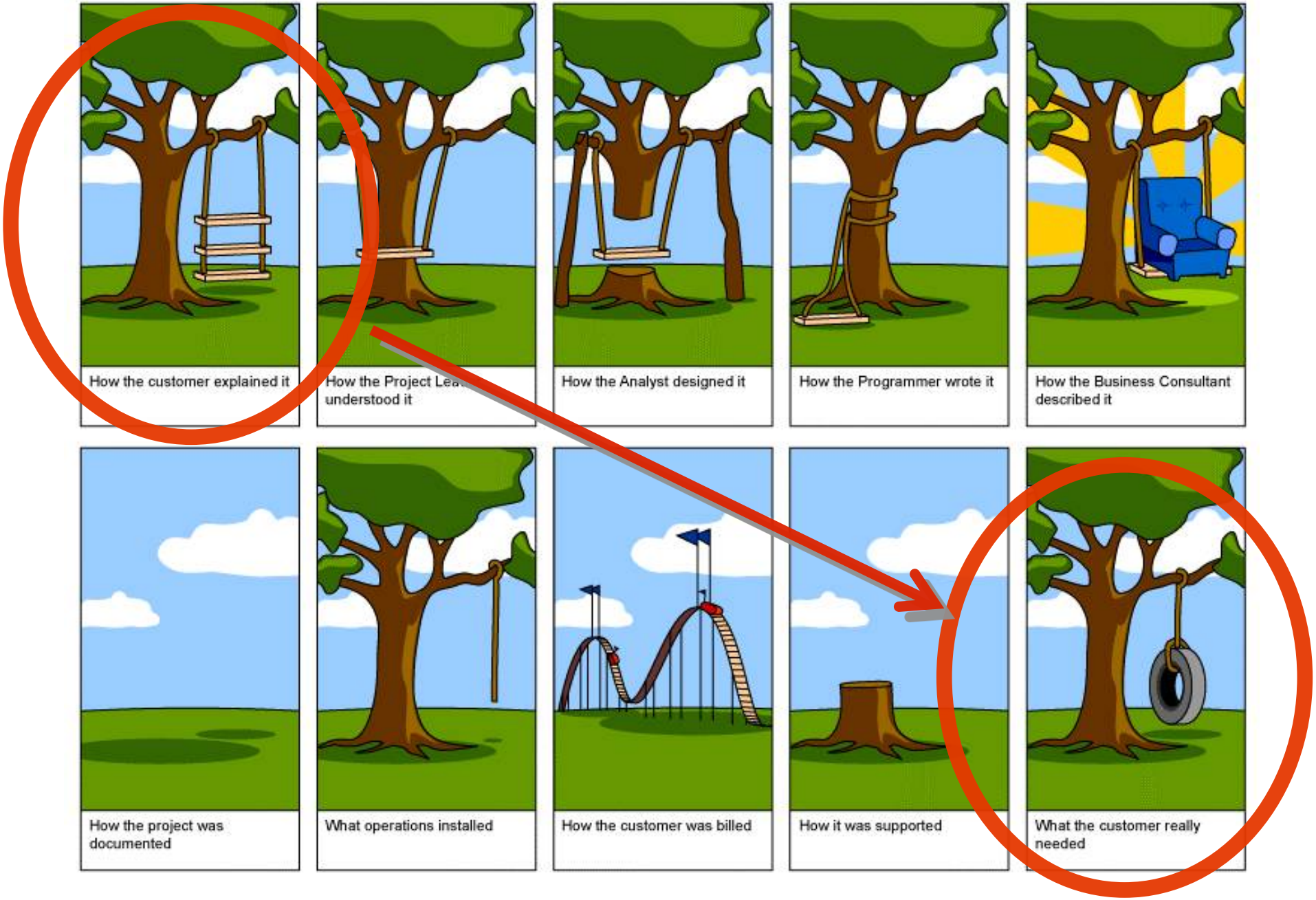
- and not “how” to build it
- tell the problem, not the solution

Classifying Requirements

- **Functional**: actors and actions
  - "The user can browse the catalog and select items."
  - "Every order gets an ID the user can save"
- **Non-functional**: other constraints
  - dependability, reusability, portability, scalability, performance, safety
  - "Our deliverable shall conform to the XYZ process."
  - "The system shall not disclose any personal user information."

Why Requirements?


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed


How the customer explained it
How the Project Lea
understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant
described it
How the project was
documented
What operations installed
How the customer was billed
How it was supported
What the customer really
needed

And we too, have, the German tanks, they weren't tested, for example, to prepare them for winter war. Instead we conducted trials to prove it was impossible to wage war in winter.
That is a different starting point [than the Soviet's]. In the fall of 1939 we always faced the question. I desperately wanted to attack, and I firmly believed we could finish France in six weeks.

4:23 / 11:39

Hitler Speaking Normally (Subtitles)

2,010,273 views

10K 206 SHARE

AlbusPercyDumbledore
Published on Mar 25, 2011

SUBSCRIBE 523

Defining Good Requirements

- Where do they come from?
- How do we record them?

Defining Good Requirements

- **Where do they come from?**
- How do we record them?

Eliciting Requirements from Users

- Access to users is a critical success factor in rapid-development projects.
  — Steve McConnell
- Good relations improve development speed
- But …
  - Users don't always know what they want
  - Even if they do know what they want, it changes over time

How to Gather Requirements

- Talk to the users, or work with them, to learn how they work.
- Ask questions throughout the process to "dig" for requirements.
- Think about **why** users want to do something in your app, not just what.
- Allow (and expect) requirements to change later.

Personas (in Agile Methods)

Personas are fictional characters, which you create in order to represent the different user types that might use your service, product, site, or brand in a similar way.

Personas

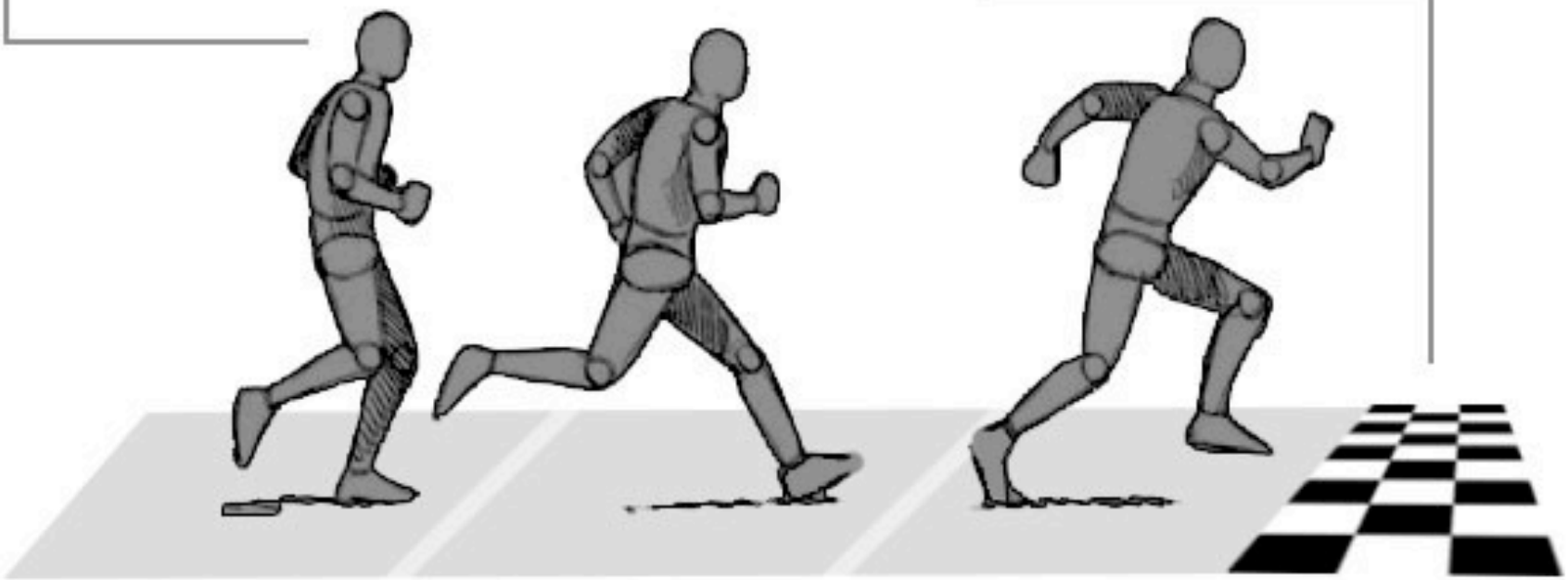
- What?
  - typical users of a system
  - examples of kinds of people who will use a system
  - most likely need several personas

1. Persona

Defines who the story is about. This main character has attitudes, motivations, goals, and pain points, etc.

3. Goal

Defines what the persona wants or needs to fulfill. The goal is the motivation of why the persona is taking action. When that goal is reached, the scenario ends.



2. Scenario

Defines when, where, and how the story of the persona takes place. The scenario is the narrative that describes how the persona behaves as a sequence of events.

Multiple Personas …

THE CASUAL USER

Pete

Uses most phone features

Uses phone to make, use contacts send texts and take pictures

Always has mobile device with him

THE BUSINESS USER

Jennifer

Whats a simple phone, but functions as an integrated device

Wants to easily read email and call back the sender

Needs "Popular" mail sever integration

THE POWER USER

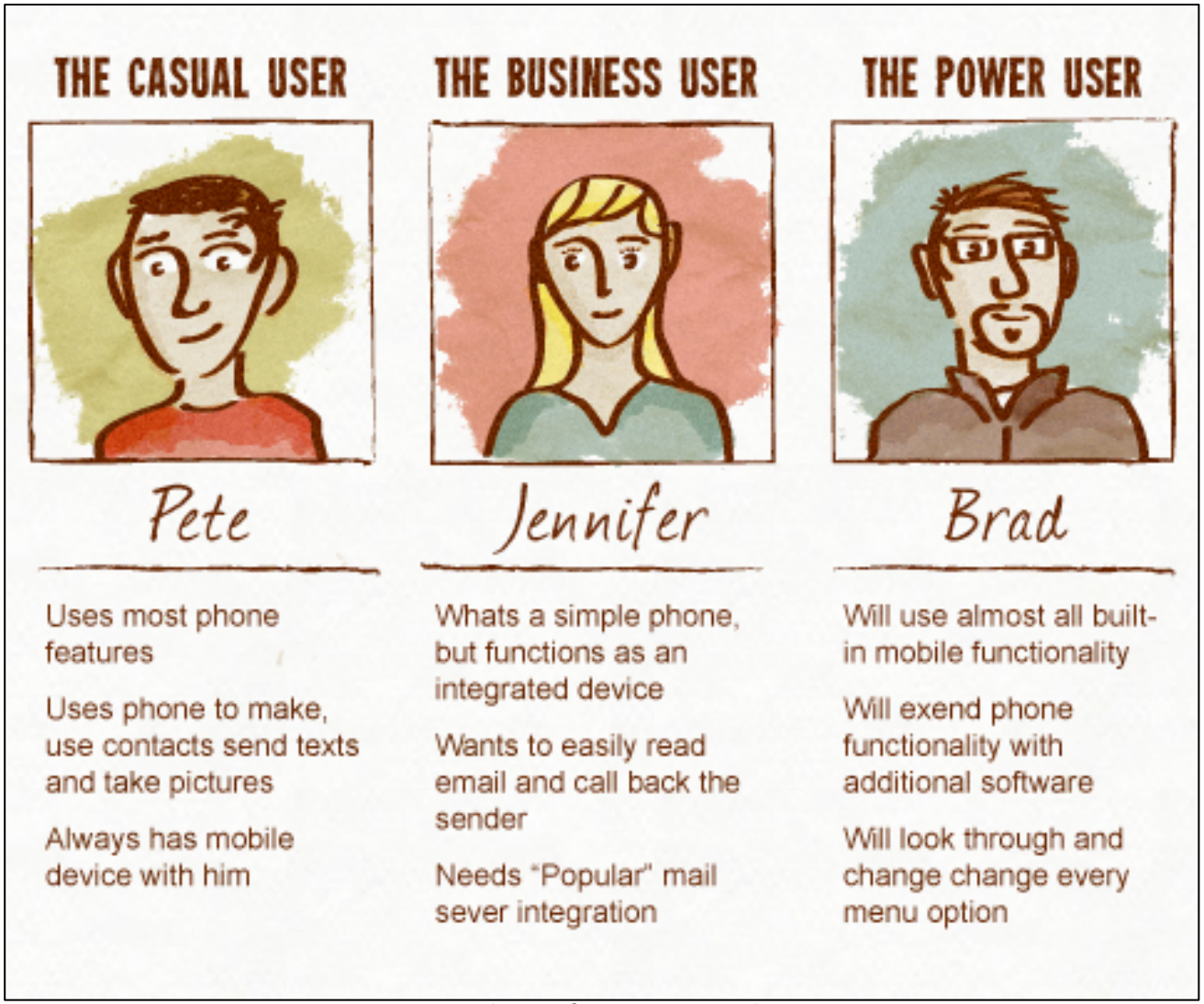
Brad

Will use almost all built-in mobile functionality

Will exend phone functionality with additional software

Will look through and change change every menu option

Multiple Personas …



But don't overdo it:

3-6 are enough in most, even very complex, cases

Personas – Used in Many Areas

- User Experience
- Marketing
- …

Personas – Advantages

- Give a “face” to the potential users. Help understand the customers and therefore to satisfy customer problems
  - Persona development characterizes your customers not by demographics, but by behaviors and desires
- Align the whole company
  - persona is a way to characterize customers for everyone in the company, so that we all understand their expectations and desires

Personas – Disadvantages

- May lead to a false sense of understanding
  - instead of talking to real customers
- Can be misused in order maintain distance from the actual people.
  - Are not real and can never produce human qualities

Defining Good Requirements

- Where do they come from?
- **How do we record them?**

Specifying Requirements

- Formal Specification (document)
- User Stories / Use Cases
- Prototype

Specifying Requirements

- **Formal Specification (document)**
- User Stories / Use Cases
- Prototype

Document
(a must in regulatory environments)

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules.

Specifying Requirements

- Formal Specification (document)
- **User Stories / Use Cases**
- Prototype

User Stories (Agile Flavor)

- A high-level definition of a requirement.

- Contains just enough information so that the developers can produce a reasonable estimate of the effort to implement it.

- When “invented”, supposed to write on index cards
  - nowadays mostly in online tools

Format ad Example

- "As persona (a role), I want something, so that benefit."

- For example:

  "As Alice (a student), I want to browse courses required for my program, so that I can prioritize my course choices."

- Depending on who you ask, the "so that" part may be
- optional.

Basic Guidelines

- Write from the stakeholder point of view.
- Optionally include a unique identifier.
- Indicate the estimated size.
- Indicate the priority.
- Use the simplest tool.
- Remember non-functional requirements

Be Specific

- As Jerry (a budget shopper), I would like to have a receipt that lists the price and any discounts, so that I have a record of my purchases.

- As Bina (a store manager), I want to offer a flat rate discount on items between Dec 26 and Dec 31.

Be Specific

- As Bina (a store manager), I want to offer a flat rate discount on items between Dec 26 and Dec 31

Not specific enough: ☹

- As a retailer, I want to offer items on sale for a reduced price for a limited time, so that I can increase traffic in the store.

Detailing out – Common Techniques

- Confirmations / acceptance criteria
- Screen sketches

Confirmations

1. Success – valid user logged in and referred to home page.
   a. 'Remember me' ticked – store cookie / automatic login next time.
   b. 'Remember me' not ticked – force login next time.

2. Failure – display message:
   a) "Email address in wrong format"
   b) "Unrecognised user name, please try again"
   c) "Incorrect password, please try again"
   d) "Service unavailable, please try again"
   e) Account has expired – refer to account renewal sales page.

- **Connect to acceptance testing**
- **Can be directly used in TDD!**

Confirmations

Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: [redacted] Should
Estimate: 4

Back of Card

Confirmations:

~~The student must pay the correct cost~~

One pass for one month is issued at a time

The student will not recieve a pass if the payment isn't sufficient

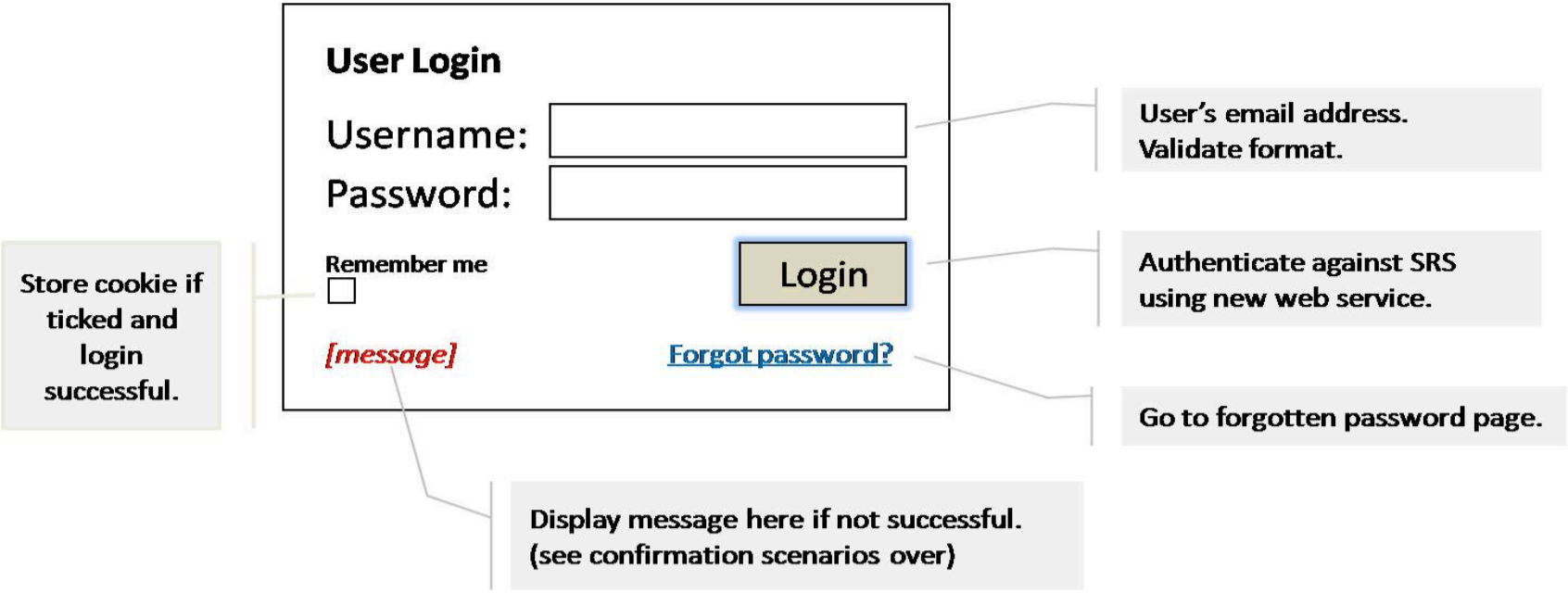
The person buying the pass must be a currently enrolled student.

The student may only buy one pass per month.

Screen Sketches

#0001	USER LOGIN	Fibonacci Size # 3
As a **[registered user]**, I want to **[log in]**, so I can **[access subscriber content]**.		

For new features, annotated wireframe. For bugs, steps to reproduce with screenshot. For non-functional stories, explain scope/standards.



Use Cases (More Formal / UML Flavor)

- Subtle difference:
  - user stories focus on a user's needs
  - use case focus on behaviors you build into the software to meet the user's needs
- Example:
  - User story: A user who realized he miscapitalized a word, wants to search for all occurrences of this word in the document and replace them with the correct word.
  - Use case: All occurrences of a search term have to be replaced with the replacement text

Format: Formal Use Case

Goal	Patron wishes to reserve a book using the online catalog
Primary actor	Patron
Scope	Library system
Level	User
Precondition	Patron is at the login screen
Success end	Book is reserved
Failure end condition	Book is not reserved
Trigger	Patron logs into system

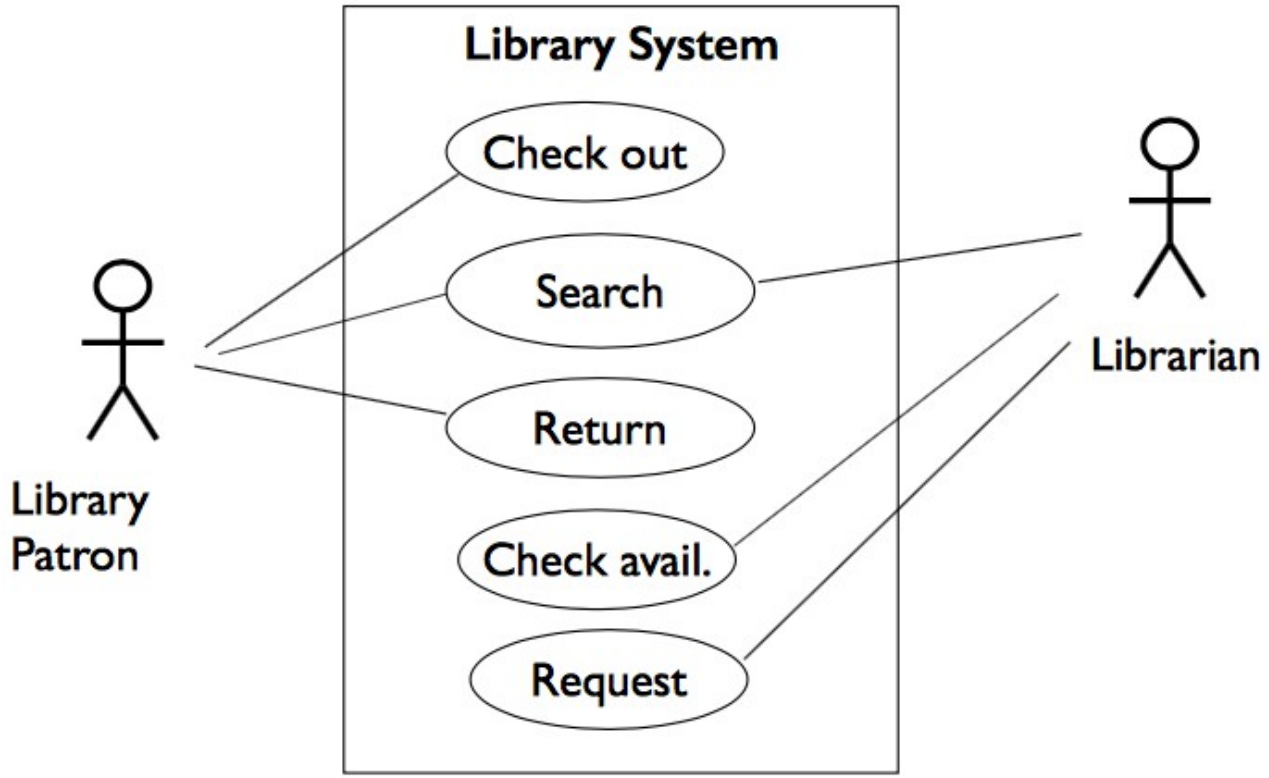
Format: Formal Use Case (continued)

Main success scenario	1. Patron enters account and password 2. System verifies and logs patron in 3. System presents catalog with search screen 4. Patron enters book title 5. System finds match and presents location choices 6. Patron selects location and reserves book 7. System confirms reservation and re-presents catalog

Format: Formal Use Case (continued)

Main success scenario	1. Patron enters account and password 2. System verifies and logs patron in 3. System presents catalog with search screen 4. Patron enters book title 5. System finds match and presents location choices 6. Patron selects location and reserves book 7. System confirms reservation and re-presents catalog
Extensions (error scenarios)	2a. Password is incorrect 2a.1 System returns patron to login screen 2a.2 Patron backs out or tries again 5a. System cannot find book 5a.1 …

Format: Use Case Diagrams (UML)



Detailed Steps: Informal

Patron loses a book

- The **library patron** reports to the librarian that she has lost a book. The **librarian** prints out the library record and asks patron to speak with the head librarian, who will arrange for the patron to pay a fee. The **system** will be updated to reflect lost book, and patron's record is updated as well. The **head librarian** may authorize purchase of a replacement book.

Informal use case is written as a paragraph describing the scenario / interaction.

Detailed Steps: Formal

- Sequence Diagrams in UML
  - Actors?

Summary

	Agile-like	More formal/ UML-like
The user	User, Persona	User, Actor
What the user wants to accomplish	User story (front of the card) e.g., "As Alice I want to borrow a book from the library"	Use case (ellipse) e.g., "borrow a book"
A sequence of steps to accomplish the user's goal	• Steps and confirmations (back of the card)	• Formal use case description • Sequence diagram

Notes

- Use case diagrams contain more info, such as relationships between actors and between use cases
- Multiple personas can correspond to one actor!

Summary: Qualities of a Good Use Case / User Stories

- Concise, clear, accessible to non-programmers
  - Easy to read.
  - Summary fits on a page.

- Focuses on essential behaviors, from the actor's point of view
  - Does not describe internal system activities, etc.

- Focuses on interaction
  - Starts with a request from an actor to the system.
  - Ends with the production of all the answers to the request.

Specifying Requirements

- Formal Specification (document)
- User Stories / Use Cases
- **Prototype (Next Class)**

Now Your Turn: W5 Milestone (1/2)

- Define two sets of requirements:
  - for your project
  - for your customer's project

- Goal:
  - effectively communicate the requirement
  - compare client and developer view
  - establish common understanding

Now Your Turn: W5 Milestone (2/2)

- Pick your technique: use cases, user stores, etc.
- Specify:
  - Users
  - What the users want to accomplish (a card, a UML use-case diagram, etc.)
  - Description of the story: use case (sequence of steps and confirmations, a sequence diagram, etc.)

Outline

- UML (summary)
- Requirements
  - What are requirements?
  - How do we gather requirements?
  - How do we document requirements?
- **Logistics**

Community Meetings

- Mandatory starting this week
- Schedule is posted on Piazza

Next milestones:

- W5: M1 – Requirements (both customer and development teams).
- W6: M2 – Design (development team).
- **W8: M3 – MVP (development team).**
- W9: M4 – Code review (development teams).
- W10: M5 – Test plan and results (development team).
- W11: M6 – Refined specifications (both customer and development teams).
- W12: M7 – Customer acceptance test (customer team).

Weekly Deliverables

- Submitted via Canvas in PDF format: https://canvas.ubc.ca/courses/13059/assignments
- Due **two days before** your Lab time.
- For example, the deliverable for W5 (requirements) is due:
  - End of the day on Saturday W4, Sept 29 for **Monday** groups.
  - End of the day on Sunday W4, Sept 30 for **Tuesday** groups.
  - End of the day on Monday W4, Oct 1 for **Wednesday** groups.

Content of the Deliverable

- 1-page progress report that includes:
  - a high-level description of the progress for the week,
  - the plans for next week,
  - major decisions and changes in the scope of the project,
  - the contributions of individual team members

- The deliverable itself, when required (like in W5 and W6).

See you on Wednesday!