Software Engineering

CSC648-848 05

Use Cases, Requirements and Specifications

Jose Ortiz
jortizco@sfsu.edu

USE CASES

Use Cases

- Drive high level **requirements** and **specifications**
- **Satisfy user needs**, not engineer needs.
- They are also used by marketing, business analysts, and product managers.
- Actors in use cases have **feelings** and use the **environment** around them.
- Create a story based on those actors focusing on **WHAT and not HOW**
- One of the most powerful components to define the client's needs for your application

Use Cases

- Methods to gather information:
  - Ask Users
  - Observe Users
  - Validation of the use case
  - Understand your user’s needs, not your needs.
  - Modeling use cases help to visualize better the high level requirements required for the app.

Use Cases Level of Detail

- High level
  - **What** the user does at high level
    - Critical: by **user needs**, not compromised by your vision
    - **How** to design the system **is not** in the scope of use cases
- Lower level
  - More details, functions, steps

Uses Cases: Online Shopping

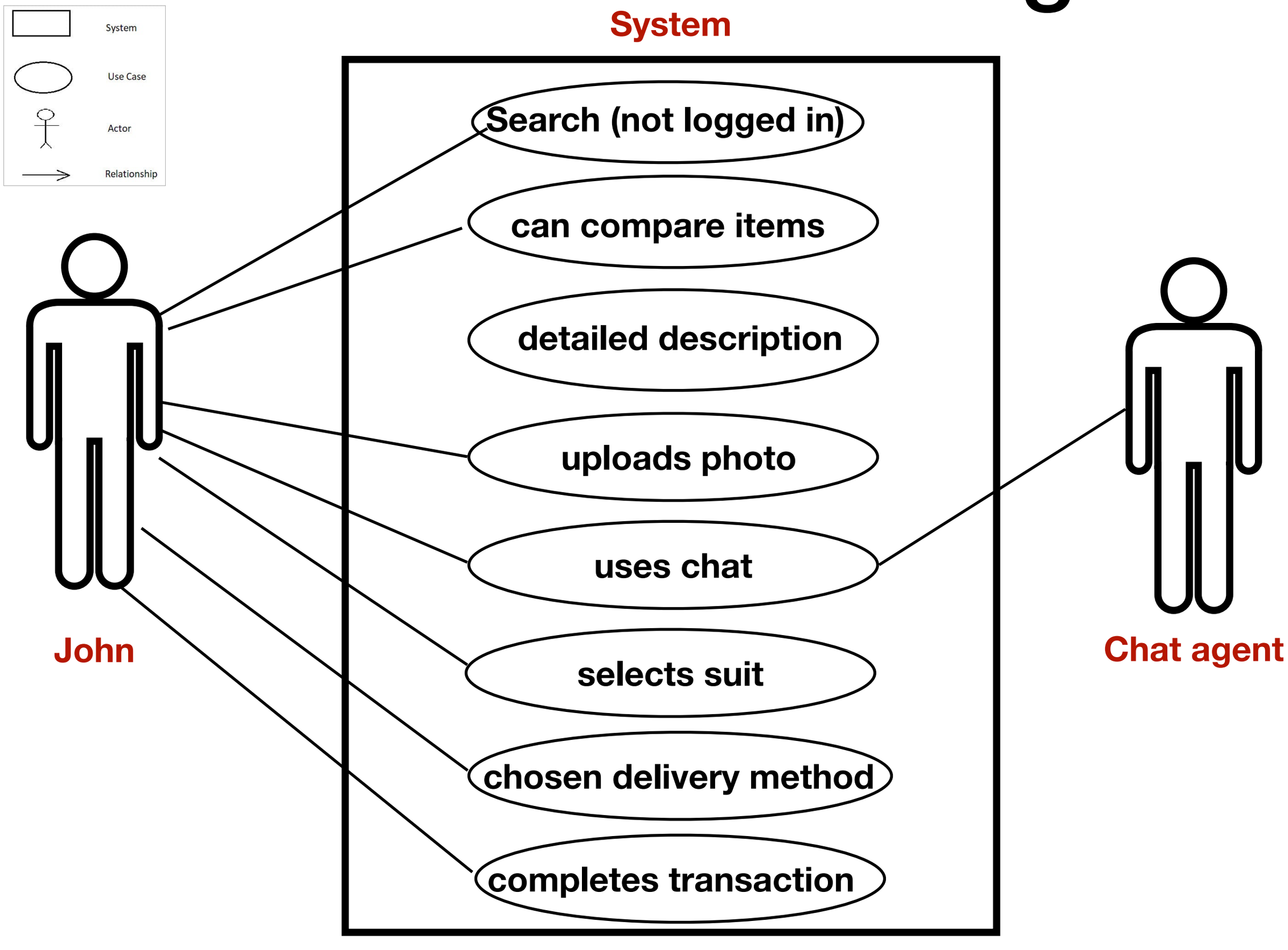
- **Actors:** John (Customer), MySuits (company)
- **Assumptions:**
  - John has internet access and a device to browse online stores
  - John has his body measurements or knows his suit size.
- **Use Case:**

John is planning to attend his sister's weeding, and he feels a bit stressed about finding the right suit on short notice. He is concerned about the time it will would take to visit physical stores. The convenience of shopping online appeals to him as it would save time and allow him to avoid crowded malls. John searches for suits on an online shopping platform (MySuits). He appreciates the wide variety of options available, which are more diverse than what he might find in a local store. The comfort of browsing from his home, at his own pace, reduces his initial stress. The website offers high-resolution images and detailed descriptions of each suit, along with customer reviews. John can easily compare different styles, fabrics, and prices. The virtual try-on feature, where he can upload his photo to see how each suit might look on him, enhances his confidence in making a choice. John uses the online chat feature to ask questions about suit care and material. He feels reassured by the prompt and helpful responses, enhancing his trust in the online store. Once John selects a suit, he inputs his size, chooses a delivery option, and completes the transaction. He is relieved and pleased with the straightforward and secure payment process. John receives immediate email confirmation with his order details and a tracking number for his shipment. The website also suggests accessories that would complement his suit, which he finds helpful.

Benefits for John:

- Shopping online saves John time and effort
- Online and customer service provides quick and tailored support
- The shopping experience reduces the anxiety associated with suit shopping for an important event

Use Case Modeling



Uses Cases: Banking System

- **Actors:** Alice (Working Professional), Bob (Alice's Friend), PrettyBank (Company)

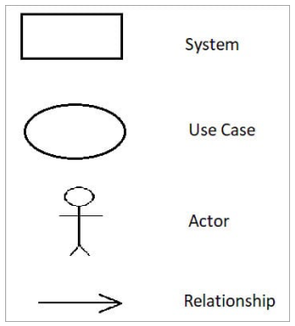
- **Assumptions:**
  - Alice and Bob have accounts with PrettyBank
  - Alice has a secure and fast internet connection

- **Use Case:**

Alice is at home, preparing for the next workday, She is feeling overwhelmed with her busy schedule but is motivated to handle her banking needs efficiently. In order to accomplish that, she needs quick and easy access to banking services because she needs to repay Bob for a shared expense. Her actual bank doesn't offer online services. However, she remembered that in a rainy day while Bob's was giving her a ride home, Bob was talking about PrettyBank, an effective and fast app that provides seamlessly banking services. She decides to gave it try and installs PrettyBank in her phone. Alice is a bit concerned about security but appreciates the app's robust login process, including biometric authentication, which eases her worry. She, finally, successfully logs into the app. Alice goes ahead, creates a new account, deposits $50 dollars, and uses the app's dashboard to quickly view her current balance. Reassured upon seeing her correct balance, confirming she can proceed with transactions. Alice, then, proceeds to pay Bob. She chooses the fund transfer option, enters Bob's account details (will be saved from previous transactions for convenience), and the amount. She is happy and appreciates the simplicity of the process and the instant confirmation of the transfer. Bob (and Alice) get a notification informing that the transaction was successful, and that the funds were credited to Bob's account, feeling satisfied with the swift transaction. Finally, Alice logs out of the app and she feels content and relieved because she trust the security measures in place.

- **Benefits for Alice:**
  - Efficiency, security, easy and quick transactions
  - Accessibility of banking services from anywhere, anytime
  - Empowered by the ability to manage and overview her finances effectively

Use Case Modeling



To be done in class

Use Case Example
Library System

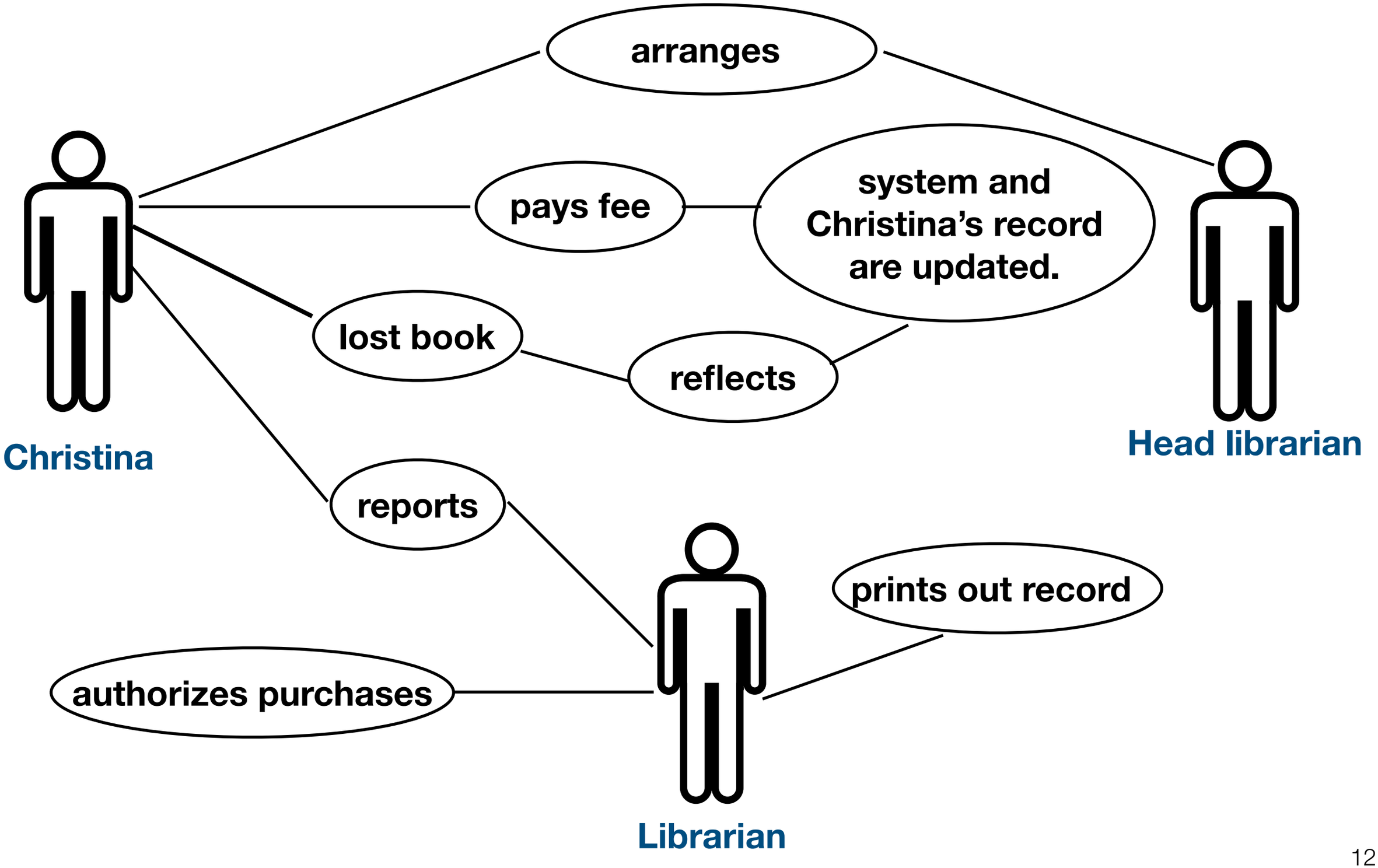
- Actors:
    - Librarian: helps users to check out/in books in the system
    - Head Librarian: supervisor/manager
    - General User: borrow, return books in the library system

Use Case Example
Library System

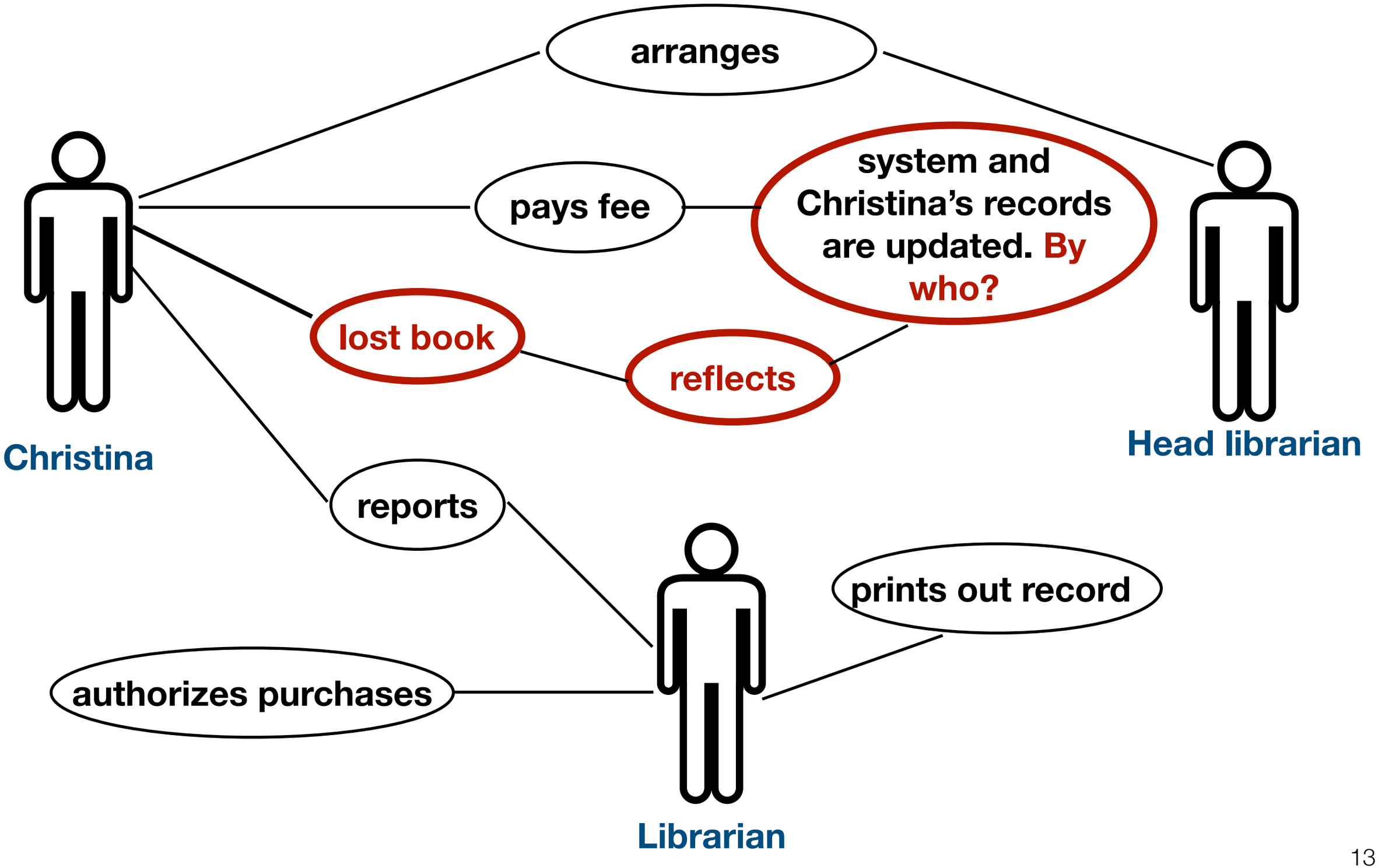
- **Informal user case with three actors**
- **Actors: librarian, Christina and head librarian**

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

Testing Use Cases
Preliminary testing



Use Cases
Preliminary test



Bad Use Case

- Mary wants to search for environmental issues. She goes to our site, types “Golden Gate” in yellow search button which is upper right corner of the site, hits GO button then goes to the list of search results presented at the bottom of the screen in two columns

Lazy References

- UI Lazy references are ok.
    - e.g. “upon submission is prompted to register or login”
    - Suggest register or login for next level of interaction.
- More examples of lazy references?

SW REQUIREMENTS

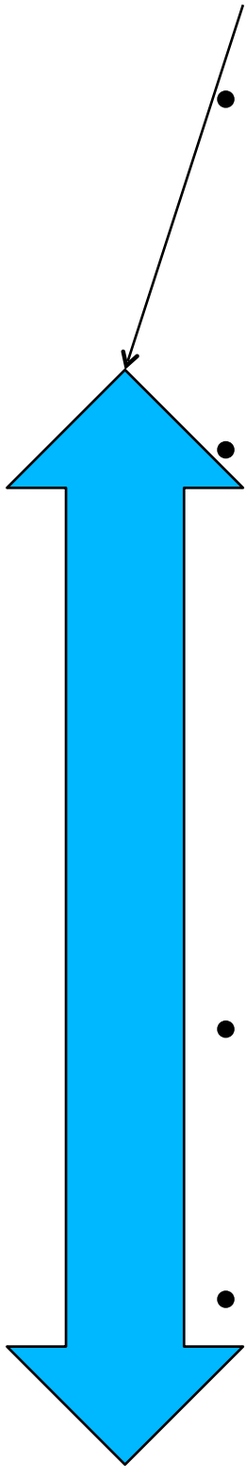
SW Requirements

- Description of services
  - Starting point for specs, design and implementation
  - Contraints to operate
  - Enables communication:
    - Clients and SW providers
    - Developers, marketing, customers….
  - Natural language and diagrams

Use Cases, Requirements, Specifications

Iterations!!!!!

Important



- **Use Cases**: describe who are the main actors, and then a story ("*use cases*") of *WHAT services/functions users will need*, in plain English. Focus on WHAT not HOW. Involve marketing, sales, competitive study, focus groups
- **Requirements**: developed from use cases, usually by marketing, engineering, sales
  - **Functional** (what functions the SW has, what does it do) (Always **<u>focus on WHAT)</u>**
  - **Non-functional** – what properties and constraints the SW has (security, privacy, performance, deploy in Cloud? etc.) (Focus on WHAT more emphasis on system services)
- **Specifications** – more concrete engineering **<u>guidance of HOW</u>** to develop the SW (often combined with requirements into SW requirement specifications - **SRS**)
- Followed by **architectural design, implementation, testing in an *iterative fashion***

Functional and Non-functional Requirements

Functional requirements:

- ***Functions:*** Describe functions and services that system provides
  - Also***: Data description:*** describe key data items and entities in the system (non compulsory but extremely useful)
- ***Usability requirements****:* describe specific usability and UI issues, users who will use the system, delivery clients

Non-functional requirements:

- ***System requirements:*** describe the system requirements (architecture, system services, networks, platforms etc.)
- ***Performance requirements****:* describe system performance (speed, accuracy, latency, delay, bandwidth etc.)
- ***Storage, security, environmental requirements***
- ***Marketing, legal requirements*** (logos, branding, licensing)
- ***Content*** (size, formats…)
- ***Privacy*** (what data is collected, how is it used…)

Functional Requirements (prioritized in milestone 2)

1) ***Functions***: Describe functions and services provided by the system

Use SHALL
Not: will, may, should…

- Examples:
  - Users <u>shall</u> be able to search using image categories
  - Users shall be able to purchase music files using credit card
  - Music shall be accessible by author and title

Note: may contains some usability requirements (in choices of functions available), but usability can be addressed also in non-functional requirements too)

Non-Functional Requirements

- Refer to system properties and constraints such as:
  - Reliability
  - Response time
  - HW and networking requirements
  - Usability requirements
  - Marketing, legal, licensing
  - Media content (formats, size…)
  - Privacy: what is the data collected, how is the data used
  - Compatibility (e.g. which browsers…)

- Can refer to:
  - Product (product behavior like speed, reliability)
  - Organization (e.g. process, standards used)
  - External factors (e.g. branding, legal disclaimers displayed)

As important as functional requirements.

Non-Functional Requirements (2)

- Examples:
  - System shall response visually within 5 seconds
  - File size in no time shall exceed 2 Mbytes
  - Users with high-school diploma, after 1 hour training, shall complete the task in 5 minutes with no more than 2 errors.
  - Tools from XYZ shall be used for requirement management
  - Each requirement shall have identifiable portion of code associated with it referenced by module name and code line number
  - Each WWW page shall have official company logo in upper left corner
  - The following user data shall be collected…; the data shall be used ONLY for for ….

Issues and problems with requirements for SW

- Full requirement set is almost **never known** at early stage
- Requirements **most often change in the process** due to real reasons (people learned more, business changed, users requested it after evaluation) and some other reasons (marketing wants more, competition is there, miscommunication) ➔ recall why it is hard to manage SW projects….
- We will address this in the class (e.g. this is why Waterfall method does not work in most of the cases)
- This is one of the key problems in building and managing large SW systems

Issues and problems with requirements (2)

- Ambiguity: can be interpreted (and will be interpreted) differently by different people or groups.
- Incompleteness: do not specify all what needs to be specified. (Authors assumed that others know too much)
- Conflicting
- Hard to verify
- Hard to use formats and document management

- **It is very hard to create perfect requirements. Don't try to be perfect, be good enough for product to succeed!**
- **Requirements are only means to an end (I.e. successful product)**

Requirements vs. Specifications

- Distinction is blurred
- One way of thinking: requirements focus on WHAT and specifications add more of HOW information
- SW specs: more concrete design and programming guidance (program design language, block diagrams, choice of platforms, compilers, etc,)
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
- Google analytics shall be added for major site functions.

.

High Level Requirements for User Interface – what to cover

- UI style
- Platforms and standards (browser, client OS, screen size etc.)
- Consistency with other playroom SW
- Key screen contents
- Use of media, graphics, logos, colors, branding
- Screen behavior
- Appearance
- User interaction techniques
- Response time
- Navigation features
- Saving data etc.
- Legacy data issues
- Support, training, help
- Error handling
- Accessibility features
- Localization, Internationalization
- …..

(Initially do high level.
Validate early.
Iterate.
Focus on WHAT not HOW.
More specific later)

Competitive Analysis

Goals

- What is a competitive analysis
- How to do a competitive analysis
- Where to use competitive analysis

What is a competitive analysis ?

- Researching before building a product
- Compare against similar existing products on the market
- Strengths and weakness from your competitors
- Product manager do this, but all the teams are involved in the process.

What is a competitive analysis ?

- Benefits:
  - Improves product strategy
  - Acquire intimate knowledge about your competitors
  - Flexible to market changes
  - Positions your product in the market place

How to do a competitive analysis ?

Stay organized:

1. Identify competitors
2. Identify important features that differentiate each competitor
3. Compare your product to the others in the market
4. Identify areas in which you can improve your product
5. Update as the product develops
6. Package the competitive analysis

Identify competitors in the space

- Identify your competitors
- Find the market you fit into.
- Identify your target audience
- Consider also competitors with small market niche
- Consider future partnership with some of your competitors
- Focus on strengths and weakness

Identify important features that differentiate each competitor

- Use each competitor product and know them well
- Gaining perspective from the user point of view
- Think about some features to be improved (i.e pricing, social media presence, on-boarding and usability)

Identify important features that differentiate each competitor

- On-boarding and Usability
  - From user perspective:
    - Easy to find what the user needs
    - Easy to use
    - Usable

Identify important features that differentiate each competitor

- Pricing:
  - Start thinking about the pricing structure (if any)
    - Low prices to attract customers, but offer bad quality products?
    - Pricing tiers organization

Identify important features that differentiate each competitor

- Social media presence:
    - Presence in the most important social media sites?
    - How often they post
    - Do they update blogs regularly

Example of identifying important features

Feature/Company	Competitor 1	Competitor 2	Competitor 3	Competitor 4
Strengths	Seamless onboarding experience, great design	Uses social media to their advantage	Great design, usability	Emphasizes security
Weaknesses	Not much interaction with customers	Documentation is hard to navigate	Sporadic social media presence	Language is formal, not user-friendly
Pricing	$800 per month	$900 per month	$850 a month	$950 a month
Social media	Blog posts, Twitter	Blog posts, Instagram	Blog posts, Instagram, Twitter	Blog posts
Onboarding experience	Smooth instructions	Not much support after first step	Seamless, very few steps involved	Moderate number of steps

https://www.getcloudapp.com/blog/competitive-analysis-for-software-products

Typical Competitive features table

Feature	Competitor A	Competitor B	Our future product
Text Search	++	+	+
Boolean Search	+	-	+
Browse	+	+	++
Shopping cart	+	+	+

+ feature exists; ++ superior; - does not exist

Highlight your product

Compare your products to the others in the market

- Factor strengths and weaknesses of your product
- Determine your target features to be developed or improved.
- Update as your product evolves

Identify areas in which you can improve your product

- Target features that need to be improved
- Focus on specific areas you'll need to compete
- Example:
  - Usability: investigate with UX/UI designers how to reduce numbers of steps that the user needs in order to find the description or pricing of a specific product.
  - Social media: may need to hire a professional blog writer to enhance or products.

Update competitive analysis as product develops

- Find new market niches as the product evolves
- Re-evaluate the product as the market keeps changing
- Conduct usability tests and new use cases as the product evolves
- Adopts good marketing strategies focused on targeting your new users.

Package the competitive analysis

- Share your work with your users:
    - Competitive analysis is not only for internal references or teams.
    - Why not let the user know about it?
    - Create visualizations about how to company compares to other for the user.

Our Company Competitive Analysis

Goodnesss

Our Company

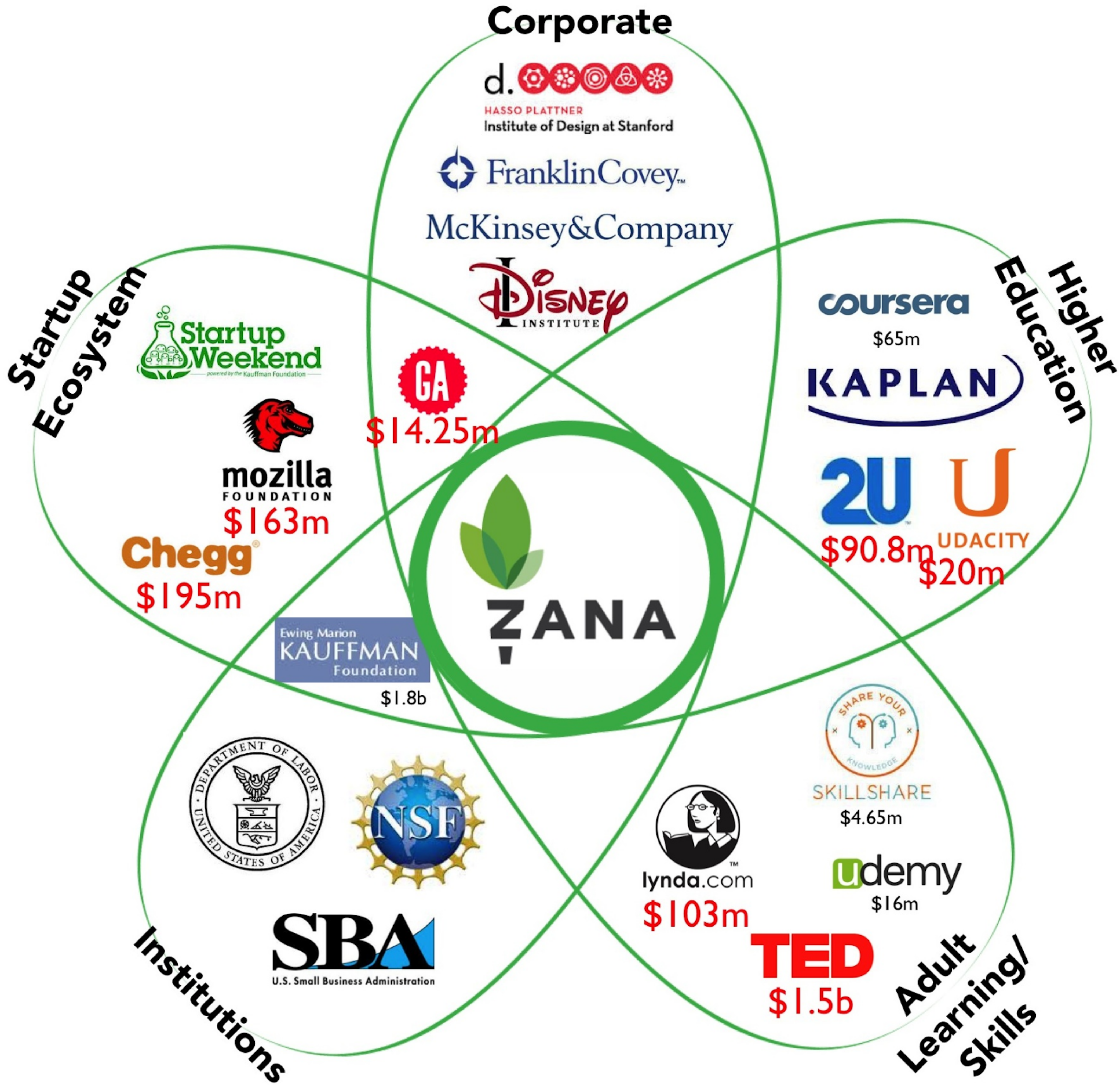
Our Competitors

Badness

Goodness

COMPETITIVE LANDSCAPE

https://www.getcloudapp.com/blog/competitive-analysis-for-software-products


Corporate
d.
HASSO PLATTNER
Institute of Design at Stanford
FranklinCovey
McKinsey&Company
Disney
INSTITUTE
GA
$14.25m
Startup
Ecosystem
Startup
Weekend
mozilla
FOUNDATION
$163m
Chegg
$195m
Ewing Marion
KAUFFMAN
Foundation
$1.8b
ZANA
Higher
Education
coursera
$65m
KAPLAN
2U
$90.8m
UDACITY
$20m
SHARE YOUR KNOWLEDGE
SKILLSHARE
$4.65m
lynda.com
$103m
udemy
$16m
TED
$1.5b
Adult
Learning/
Skills
NSF
SBA
U.S. Small Business Administration
Institutions

Where to use the competitive analysis

- Pick decks
  - Investors love to see a good competitive analysis
- Marketing material: helps to design marketing campaigns
- Team meetings: clear reasons behind your product design choices.
- Users: they love to find out why your product is better than other similar products that they need.

Competitive Analysis Summary

- Provides clear picture of the market
- Defines your target audience
- Creates a data-driven strategy for your product
- Helps in the future evolution of your product.