

# Course Application Design

Creating beautiful and reliable applications  
**From requirements to specifications**

Michiel Noback  
Institute for Life Sciences and Technology  
Hanze University of Applied Sciences



# Presentation contents

- In this presentation we'll address the first part of the design process:
- Going from assignment to specifications and initial design

# **Part one**

## **Gathering requirements**

# Requirements analysis

- Requirements analysis encompasses
  - Meeting with the client, interviewing her to get an idea of what is needed, how the software will be used and by what type of users
  - Translating this information into a set of specifications (or use cases or features) that both of you agree upon

# Phases in requirements analysis

Requirements analysis includes three types of activity:

- ***Eliciting requirements***: the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.
- ***Analyzing requirements***: determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.
- ***Recording requirements***: Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

# Interviewing rules (1)

- Have a Question/Checklist. At best this is a list of what the interviewer wants to know.
- Listen. Really listen
- Get specific
- Mirror:
  - "Let me see if I got this right: ..."
  - "So you're saying that "
- Give the interviewee ample time to respond

# Interviewing rules (2)

- Breaking the Jargon Barrier
- Find out how important each feature is
- Reduce it To Writing: you should reduce every interview to writing and show a summary to the interviewee

# User stories

- A user story is one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function
- It often has this form:  
"As a <role>, I want <goal/desire> so that <benefit>"



# Use case

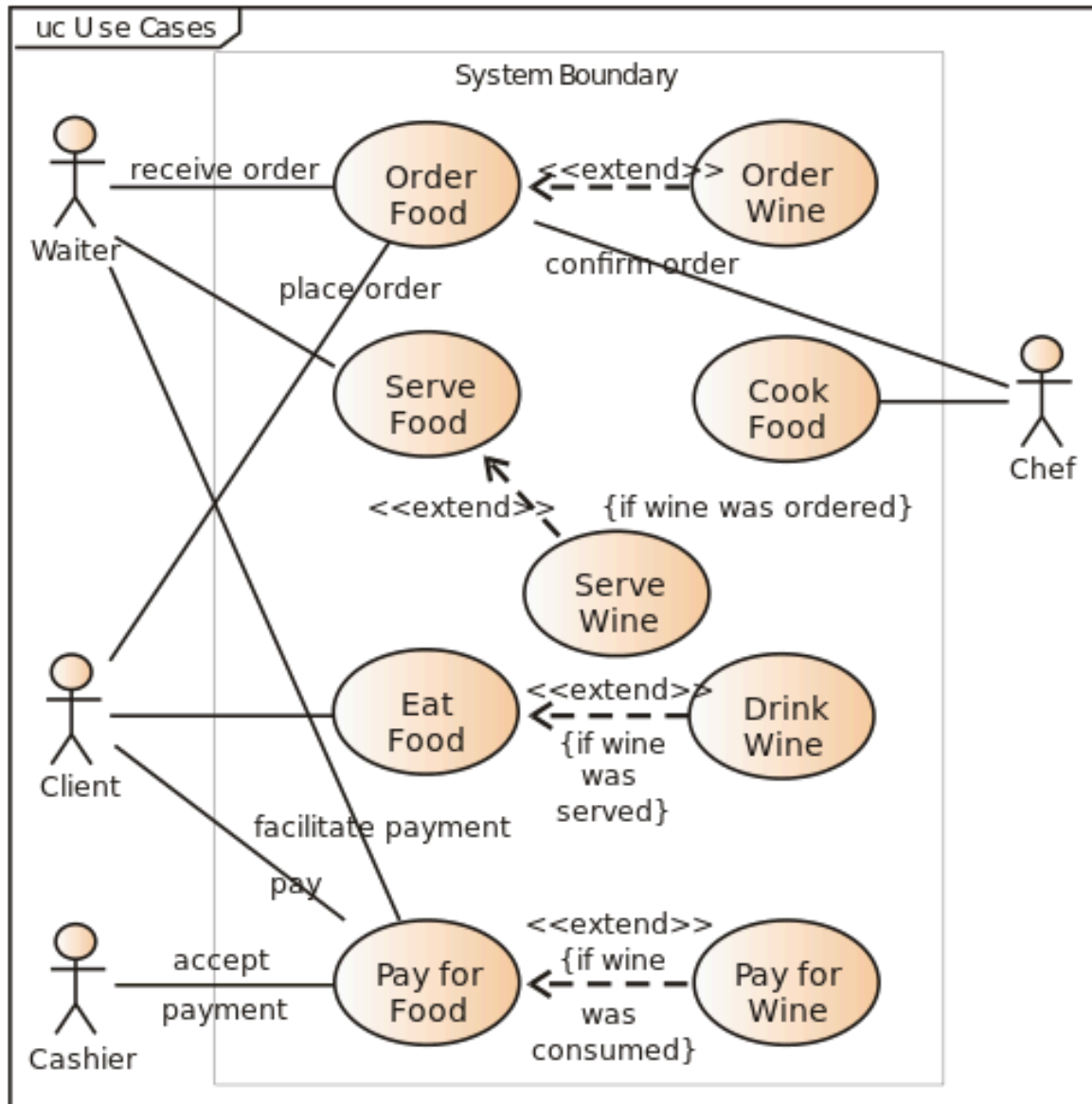
- A use case is a list of steps, typically defining interactions between a role ("actor") and a system, to achieve a goal. The actor can be a human or an external system.

Use Case  
Number: 1

Use Case  
Name: Buyer Places a Bid

Description: An EBAY buyer has identified an item they wish to buy, so they will place a bid for an item with the intent of winning the auction and paying for the item.

# Use case UML

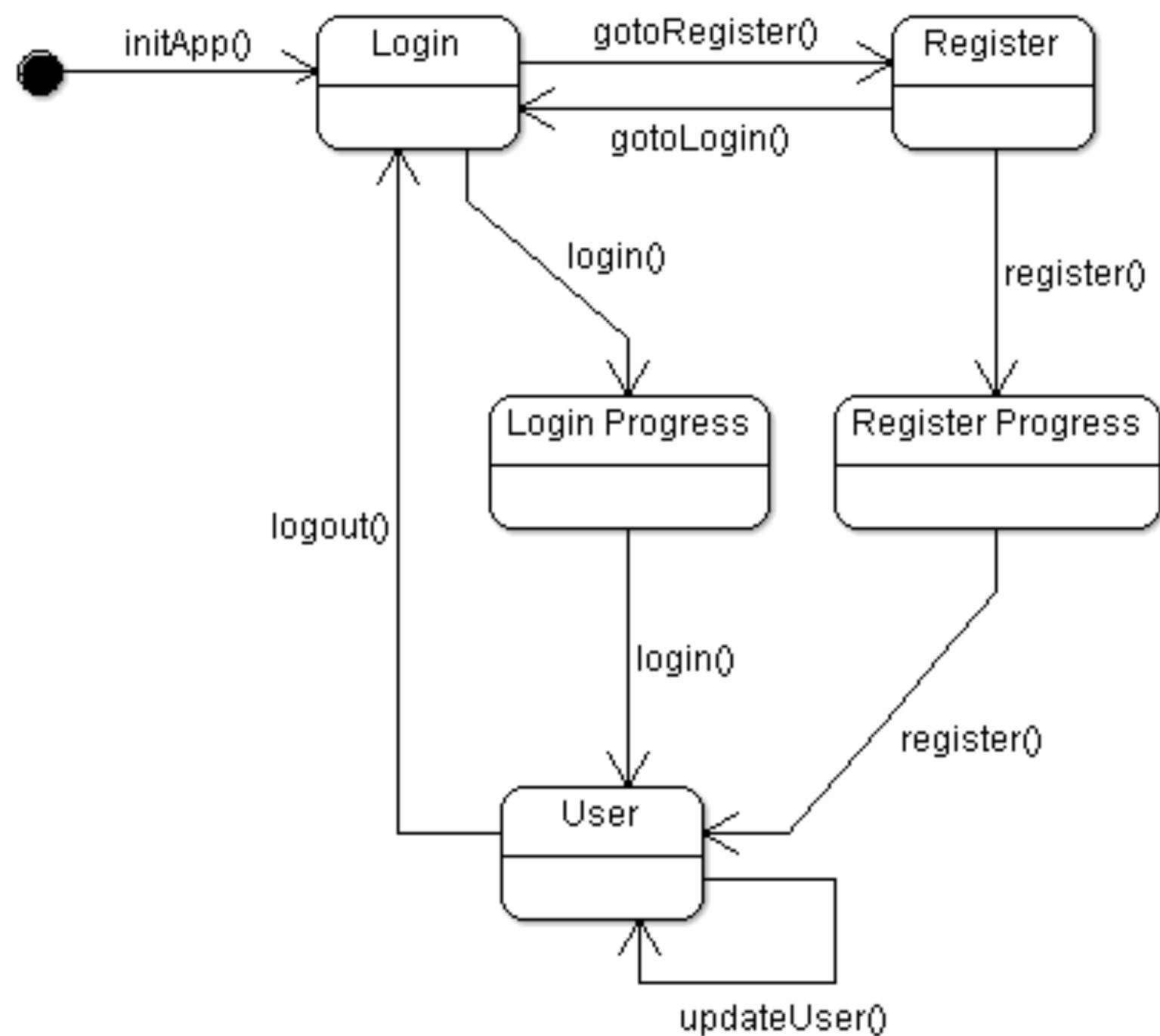


# Use case

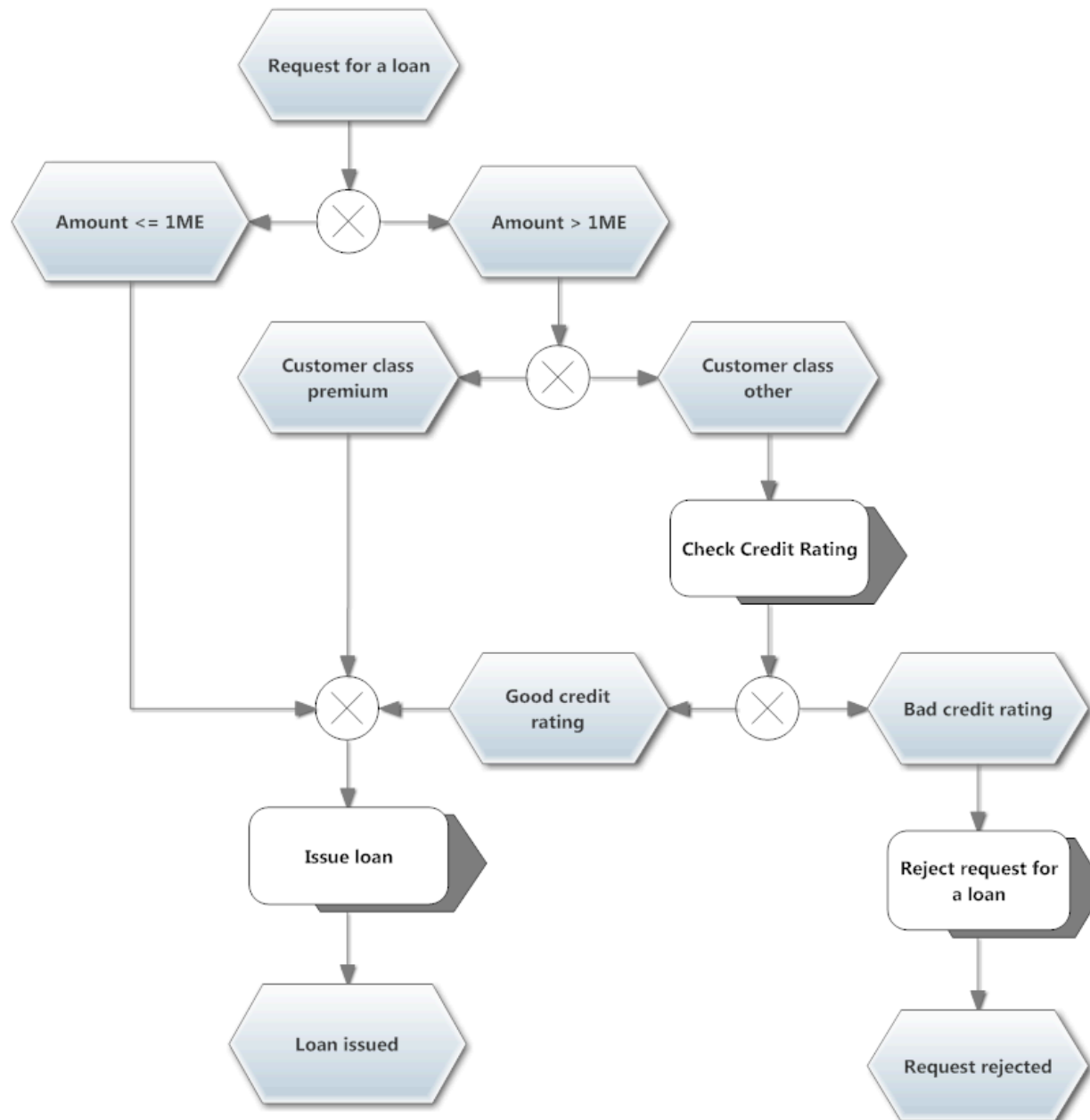
- The following link provides an excellent set of instructions for specifying use cases  
[http://www.gatherspace.com/static/use\\_case\\_example.html](http://www.gatherspace.com/static/use_case_example.html)

# Process specifications

- Process specification is the provision of clear, detailed directions to follow a series of steps to complete a task



# Loan Application Process Example



# Types of requirements

Requirements are categorized in several ways:

- Customer
- Operational
- Performance
- Utilization
- Effectiveness
- Environment
- Architectural
- Behavioral
- Functional
- Design Requirements
- ...

see [http://en.wikipedia.org/wiki/Requirements\\_analysis](http://en.wikipedia.org/wiki/Requirements_analysis)

## **Part two**

# **Defining specifications: Writing Software Requirements Specification**



# Software Requirements Specification

- Establishes the basis for agreement between the customers and the suppliers on what the software product is to do.

# What should the SRS address?

- a) Functionality. What is the software supposed to do?
- b) External interfaces. How does the software interact with people, the system's hardware, other hardware, and other software?
- c) Performance. What is the speed, availability, response time, recovery time of various software functions, etc.?
- d) Attributes. What are the portability, correctness, maintainability, security, etc. considerations?
- e) Design constraints imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

# What are the characteristics of a great SRS?

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

# Part three

## Interface design

# Interface/UX design

- Actually, this is still a part of the specifications phase
- However, since it is really important, and will in practice extend well into the implementation phase, it deserves a separate section in this presentation

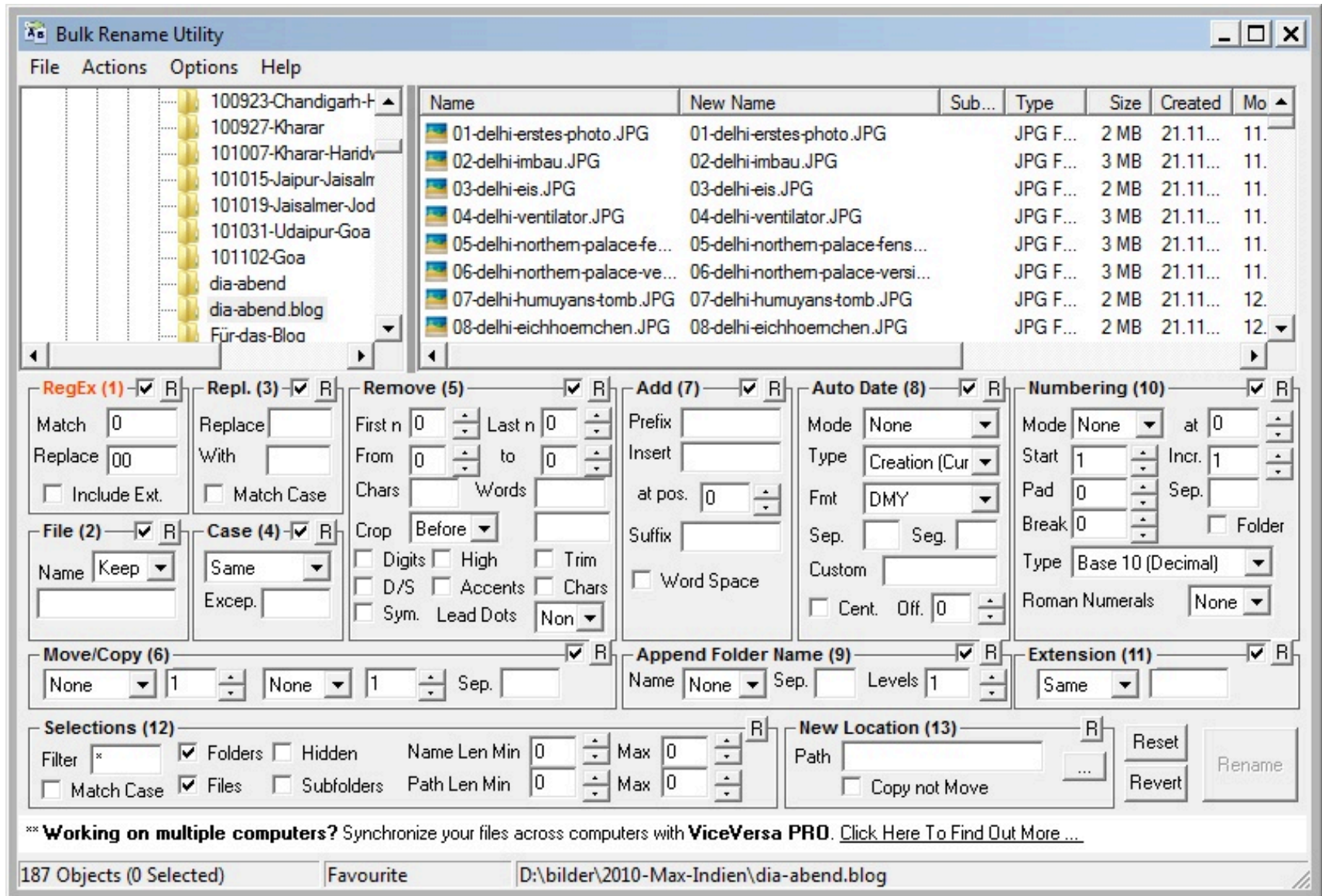
# User Experience

- Let's take the oldest definition of this concept
- *All the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they're using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it (Alben 1996).*

# User Experience guidelines

- These are the guidelines we will address in this course:
  - The tool serves the purpose of the end-user well
  - The end-user understands how the tool works
  - The end-user feels good about using the tool

# What not to make





# PROBE MATCH

CLOSE

HELP

Target String UGGAGGGGGGAUACUACU

MATCH

Use weighted mismatches

Clear

RevCompl

Compl

Auto



EXPERT

IUPAC

Check rev.compl. too

Mark in database



Write Result to field 'tmp'



PT\_SERVER

localhost: 55U\_rRNA.arb

Search depth

SEARCH UP TO NULL MISMATCHES

Number of Hits: 497

PRINT

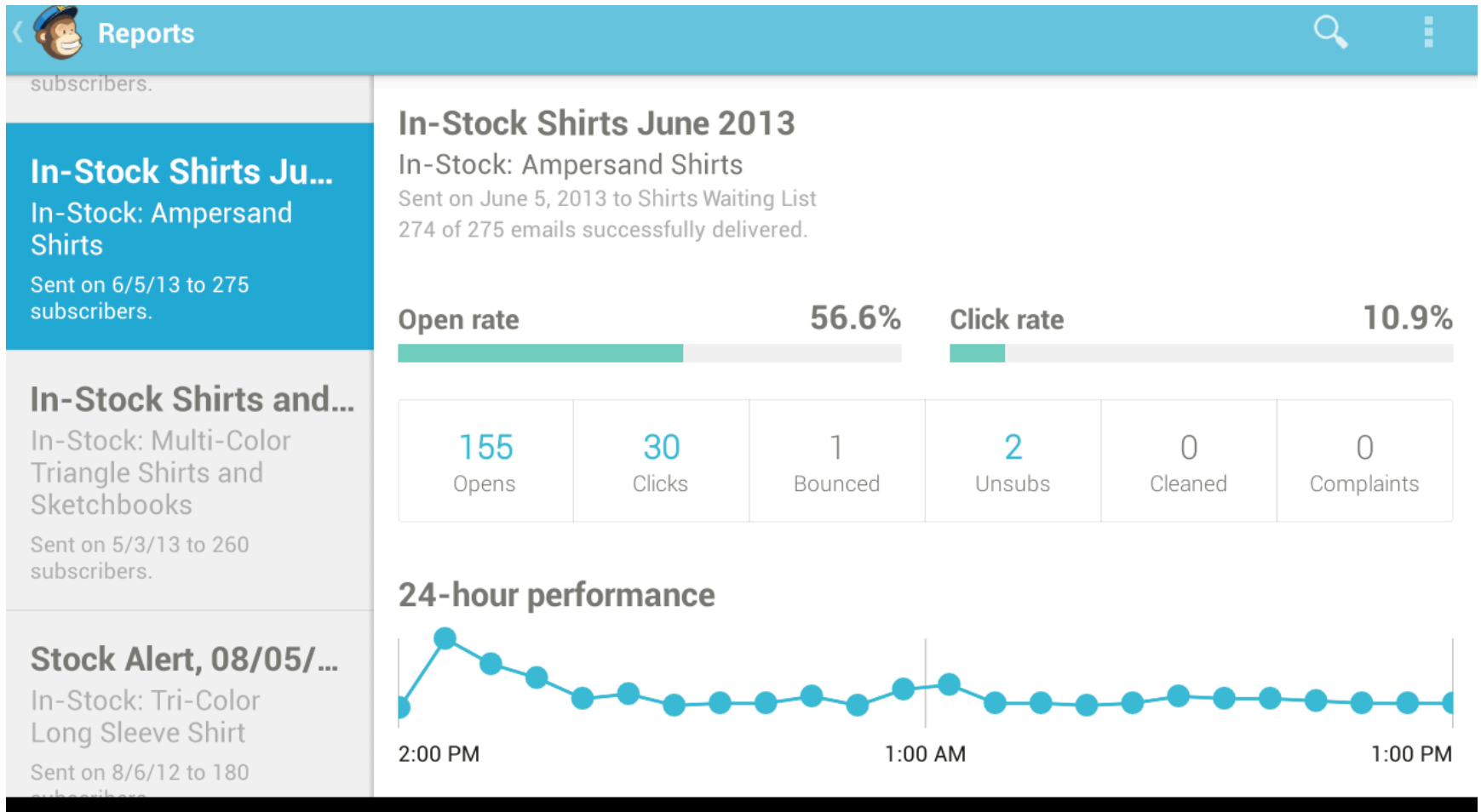
Match SAI

Searched for

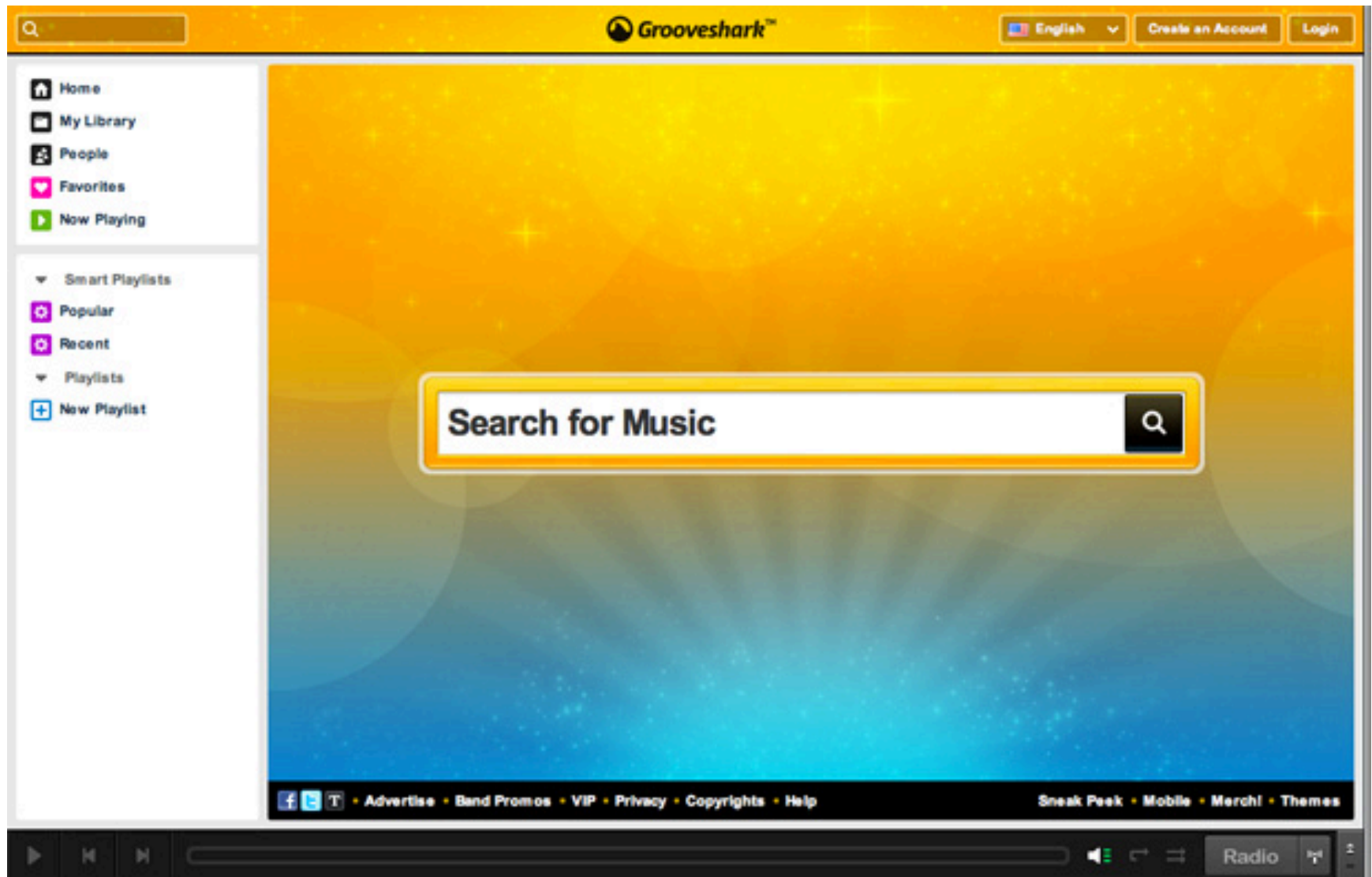
UGGAGGGGGGAUACUACU

name	fullname	mis	N_mis	unis	pos	ecoli	rev	'UGGAGGGGGGAUACUACU'
* EscCo100	Escherichia coli	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntAggl5	Enterobacter agglomerans	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntAsbu2	Enterobacter asburiae	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* SerFica4	Serratia ficaria	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntSaka2	Enterobacter sakazakii	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntInte2	Enterobacter intermedius	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntGerg2	Enterobacter gergoviae	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntAnni2	Enterobacter amnigenus	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* EntAero5	Enterobacter aerogenes	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* SerLiqu2	Serratia liquefaciens	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* KlePne20	Klebsiella pneumoniae	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* KleOxyt9	Klebsiella oxytoca	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU
* KlePlan2	Klebsiella planticola	0	0	0.0	1861	139	0	ACUGCCUGA-----GGAACCGU

# This could work



# It can be this simple



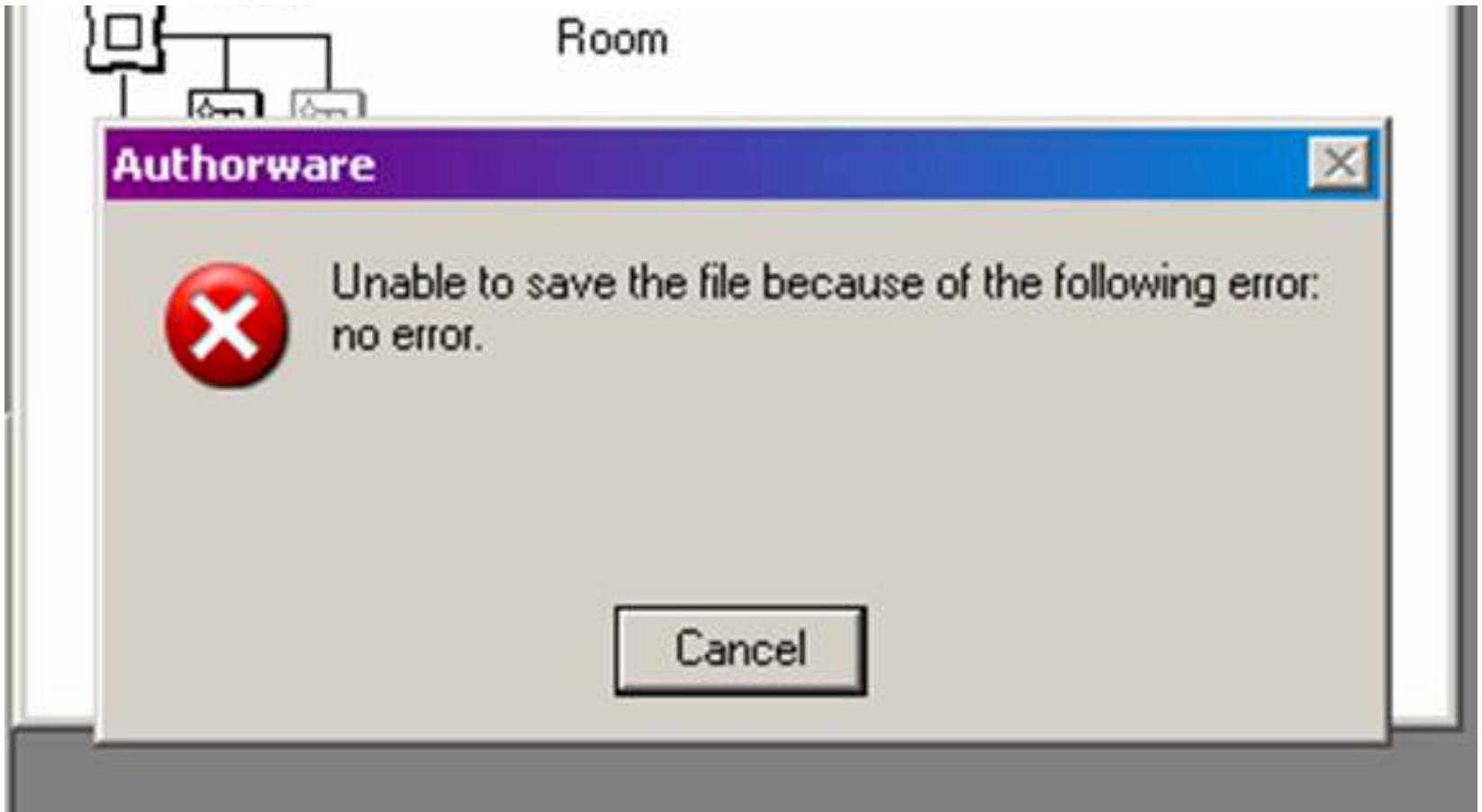
# User Interface design rules (1)

- Consistency between pages, functions, and options is vital
- Make it easy to undo or repair mistakes
- Highlight changes
- Enable Keyboard Shortcuts
- Use Familiar Standards and Conventions
- Offer Personalization Options

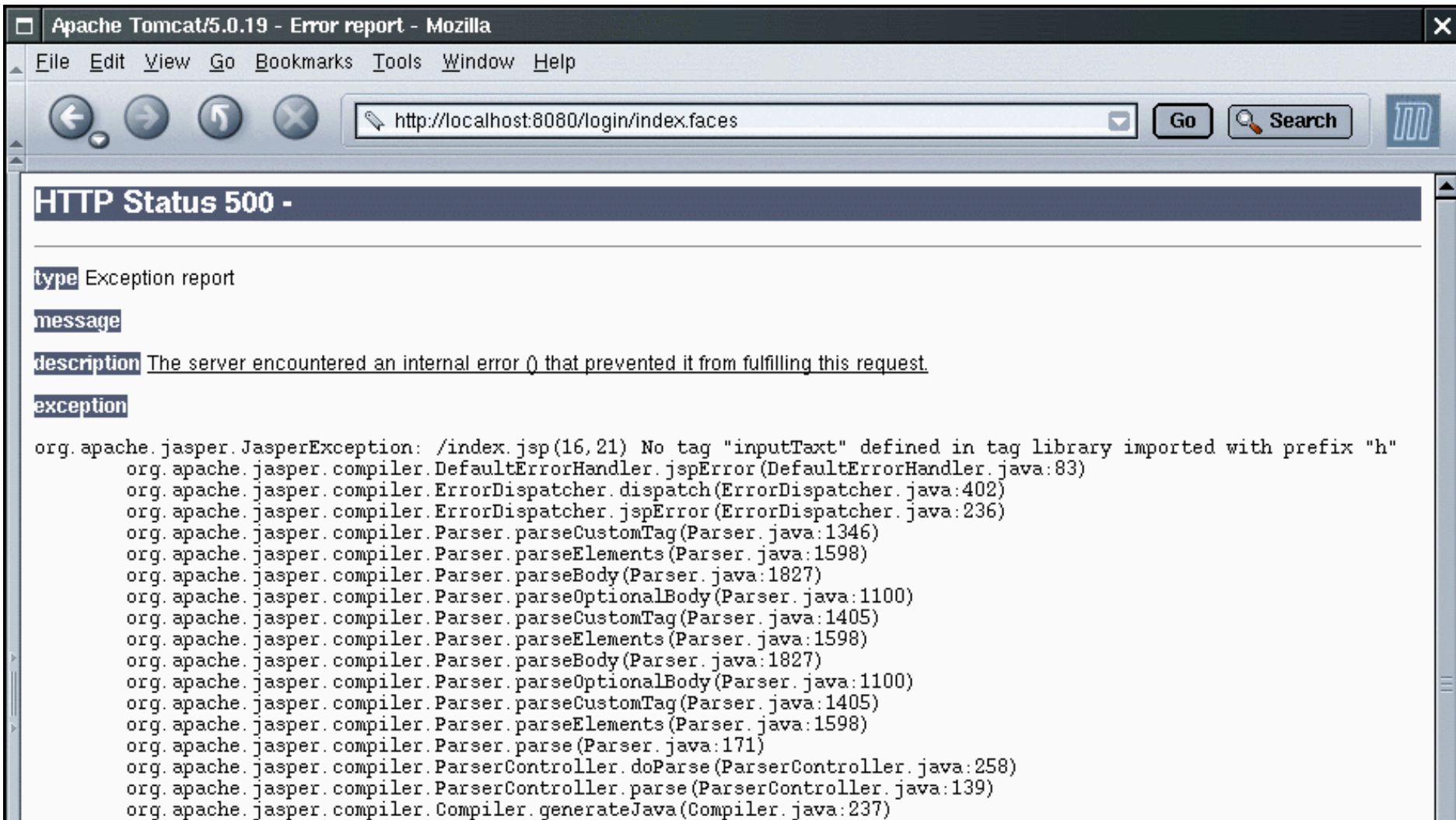
# User Interface design rules (2)

- Use Tooltips and Integrated Help Messages
- Use Tabbed Navigation and Buttons for Actions
- Use Relevant Icons and Labels
- Keep Things Simple: e.g. hide expert options from everyday use

# Be informative of errors



# Don't let your guts spill out





**Suggest workable solutions**

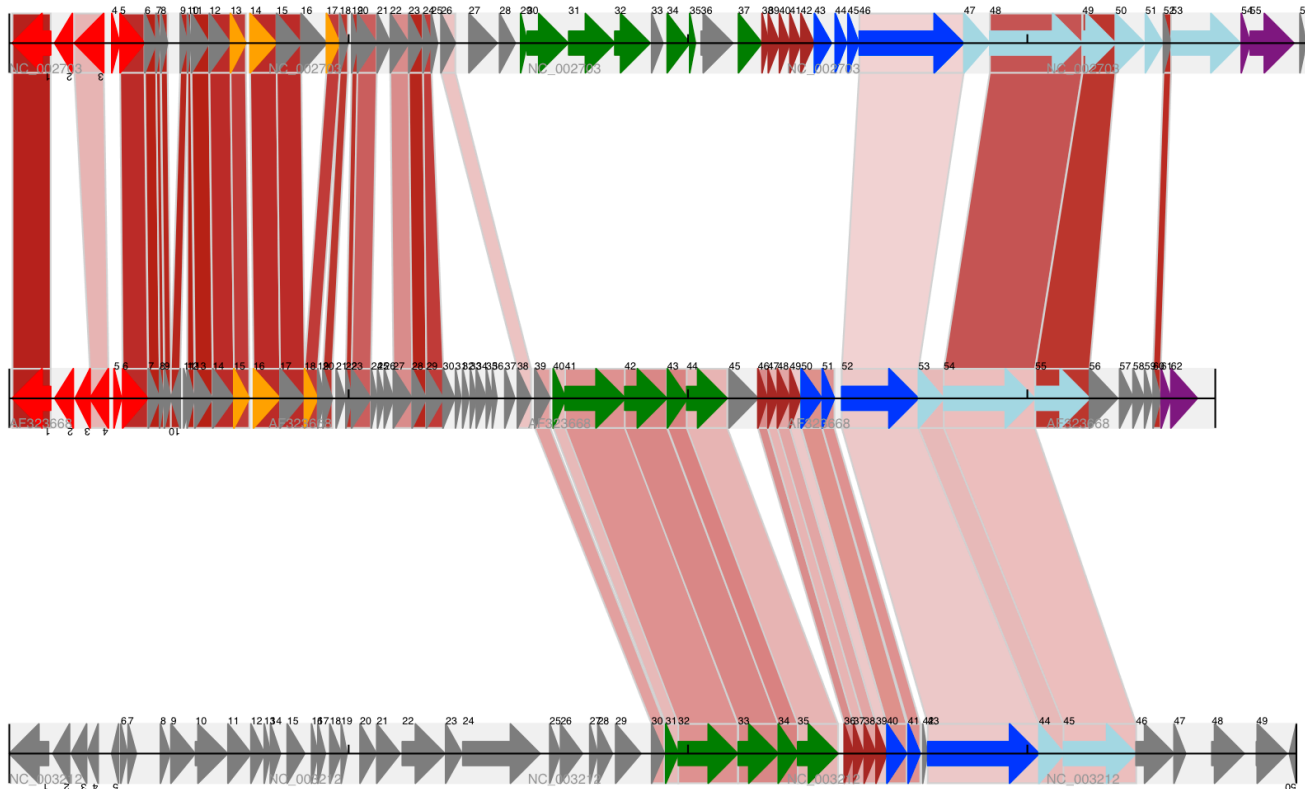
**Cannot connect to the internet!**

**Please go online to fix this  
problem**



# Data visualization challenges

- Often, you will need to reduce large amounts of complex data to a visual form the user intuitively understands



# Data visualization techniques

- Use standards where they apply: pie charts, bar charts and scatter plots are understood by everyone
- Keep tables (yes you can use those too!) readable

## **Part four**

# **Putting it all together**

# End-products

- The end-product of the requirements gathering & specification phase is a document describing the what the software will do in the form of
  - Use cases
  - User stories
  - Features
  - Processes
  - User interface examples
- You are free to use the combination of forms that suits your needs!