

Lecture 2

Requirements + Design

21 Jan 2017

Good Requirements are

- Correct
- Valuable
- Easy to Read
- Design Free
- Attainable
- Complete
- Consistent
- Unambiguous
- Verifiable
- Atomic

In-class Exercise

security must be
considered as part of the
software requirements

Security Threats

Spoofing

Tampering

Repudiation

Interception
Info disclosure

Denial of service

Elevation of privilege

As a customer I want to be
able to view my account
balance

precondition: “user must be
authenticated”

another way is to specify
what the user should
NOT be able to do

Abuser Stories Misuse Cases

as a thief, **I want to** try all
possible passwords **so that**
I can login as my victim

In-class Exercise

As a customer, I want to:

- log in with my username and password so that I can access my account online
- transfer money to another account after I logged in
- be able to log out after I login
- reset my password when I forgot what my password is so that I do not loose access to my account

Software Design

The “how?”

- which framework/library to use?
- what platform to support?
- what's the database schema?
- what the UI looks like?
- what are the major components? how do they interact?
- what are the APIs?

- ER diagrams
- UI design
- Sequence diagrams
- Architectural diagrams
 - development view
 - logical view
 - physical view

- No fixed rule for architectural diagrams
- Just be consistent, clear, readable

You need to explain
your decisions!

Target for Next Week

- Major decisions (platform, framework, high-level architecture)
- UI sketches
- ER models
- Refined requirements