Slide 1

Tracking Wine

Overview

- Groups of 4 - 5 people
- 1st hour:
  - Requirements
  - Design activity diagram
- break
- 2nd hour:
  - Discussion: another group's design
  - Design class diagram
  - Discussion: another group's design
- $\hat{x}\mathrm{abc}\alpha$

Case Study

1333	*2424*
3	4

> - Swiss love their **wine** and cheese
> - Let's create a *simple* wine tracker system

La Cave Vivante

<Youtube id="IsaW8wF2hbg" />

Simple idea

> - Bottles have a RFID tag
> - RFID reader (emits and read signal)
> - **Raspberry Pi**
> - **Server (online shop)**
> - Mobile app

The Raspberry Pi needs to handle:

> - Bottles in
> - Bottles out
> - Communication with server
> - Addition: Cellar identity (update cellar)
> - Addition: Employees have Tag (identity)
> - Important: Unreachable server?

What happens when the server is unreachable?

Requirements II

The online shop needs to handle:

> - credit cards,
> - invoices,
> - manage cellar (incr / dec)
> - auto-order (#wine < 5 => send more)
> - auto-order alerts (emails, sms, etc)
> - drink with responsibility alarm
> - ...

Break

Class Diagram

- Static view of the application
- Represent software entities (classes, interfaces, etc)
- High level overview of the implementation [1]

Hint: Domain objects

> - Wine
> - Tag ID
> - Location
> - Region
> - Cellar
> - Type of grape
> - User
> - RFID reader (polls data)
> - Server (where to send data)
> - Online shop (credit card, emails, sms, alerts)

Conclusion

- Activity diagram
  - convey general idea (business)
  - sequence of steps
- Static diagram
  - models software entities