University of Antwerp
Faculty of Applied Engineering

5-Software Design

Lab Session 9

17/12/2024

Jens Duym

Course Outline

Outline labs

- **Part A: UML diagrams**
  Sessions 1 – 2
- **Part B: Design Patterns**
  Session 3 – 5
- **Part C: Projects in groups of 2**
  Session 6 – 10

- **Evaluation in January:**
  - Entire portfolio: directories for each separate lab session, containing UML diagrams and code
  - Defence of projects

Part C
Project

Money Tracker

Money Tracker

Travelling in a group

- **Person A** pays dinner for whole group
- **Person B** pays movie tickets
- **Person C** pays lunch for everyone, but each of them eats something different

- How much do they need to **pay to each other**?

Money Tracker

Functional requirements

- People could be added or removed from application
- At the end of the trip, a global bill should be made, telling who should pay who and how much

- Functional GUI
  - The fact that it works, is top priority
  - How it looks like, is lowest priority

Money Tracker

Functional requirements

- Each person can input his/her ticket
  - They need to fill in:
    - Themselves as person who paid
    - How much they paid
- Tickets for different kinds of events:
  - Airplane tickets, restaurant tickets, taxi payments, concerts, …
  - Feel free to add more kinds
- At least two different kinds of tickets:
  - Everybody should pay the same: ticket is split evenly
  - Everybody had something different: ticket is not split evenly

Money Tracker

Non-functional requirements

- Single database for Persons
- Single database for Tickets
- Made with Design Patterns
  - All seen in the lab sessions:
    Singleton, Observer, (Abstract) Factory, MVC
  - At least one of the theory sessions:
    Strategy, Decorator, Command, Adapter, Façade, Proxy,
    Iterator, State, Template Method, Composite
- UML Diagrams
  - Class diagram of whole application
    GUI and "Jstuff" can be abstracted to just a GUI class
  - Use case diagram of application
  - Sequence Diagram of process when a user clicks "Calculate for whole trip"
- Tests
  - Unit tests for at least one class
  - At least one integration test

Money Tracker

Execution

- Groups of 2 students
  Fill in the BlackBoard form with the name of your student colleague by Monday evening November 29th!
- Using code or diagrams of others:
  - From fellow students: **discouraged**
    Only grades on what is yours, not from others
    Code or diagrams from other groups = **plagiarism**
  - From the internet:
    Feel free to reuse code -> give credit!
    (credits: websites, names, YouTube-links, StackOverflow, …)
- Try to use Git and GitHub
  - Ideal for group projects, with features as branching and committing
  - **It's for your own safety**
  - In case you've never heard from Git: an introduction
    (https://www.youtube.com/watch?v=SWYqp7iY_Tc)

Money Tracker

Evaluation

- Present and defend your project
  - A small demonstration that shows the functional requirements
  - A small presentation that shows the non-functional requirements
    - UML Schemes
    - Which patterns you implemented
    - Which tests you have written
    - Something in your program that you're very proud of
    - …
  - All in a limited time frame (± 5 minutes)
    Make it a pitch in which you sell your app!
    Every cool feature that wasn't mentioned in your presentation is NOT evaluated
  - We will ask one or two questions

Money Tracker

Delivery

- Handing in your portfolio on January 26th via Blackboard
  Hard deadline at noon, 12.00pm
  - A single zip-file named “Portfolio_FirstnameLastname.zip”
  - Contains:
    - All individual lab sessions in individual zip files
      - Lab1_FirstnameLastname.zip contains two zips: one for code and one for UML.
      - Lab2_FirstnameLastname.zip contains two zips: one for code and one for UML.
      - Lab3_FirstnameLastname.zip contains one zip for code. UML zip is optional, feel free to add.
      - …
    - One big zip of your project, containing two zips: one for code and one for UML diagrams
      - Project_FirstnameLastname.zip
    - The presentation you presented
    - If you used Git: accessible link to your repository
  - **Everyone sends portfolio with their version of the group project**
  - Zips for UML: export in the right way via Visual Paradigm (see Lab 1)
    Empty or wrongly exported zip = no grades