Project of Software Development

Lab 5 - 8Mar and 11Mar
Practical Exercise 3: OutFenix

ISEG Lisbon School of Economics & Management Universidade de Lisboa
U LISBOA | UNIVERSIDADE DE LISBOA

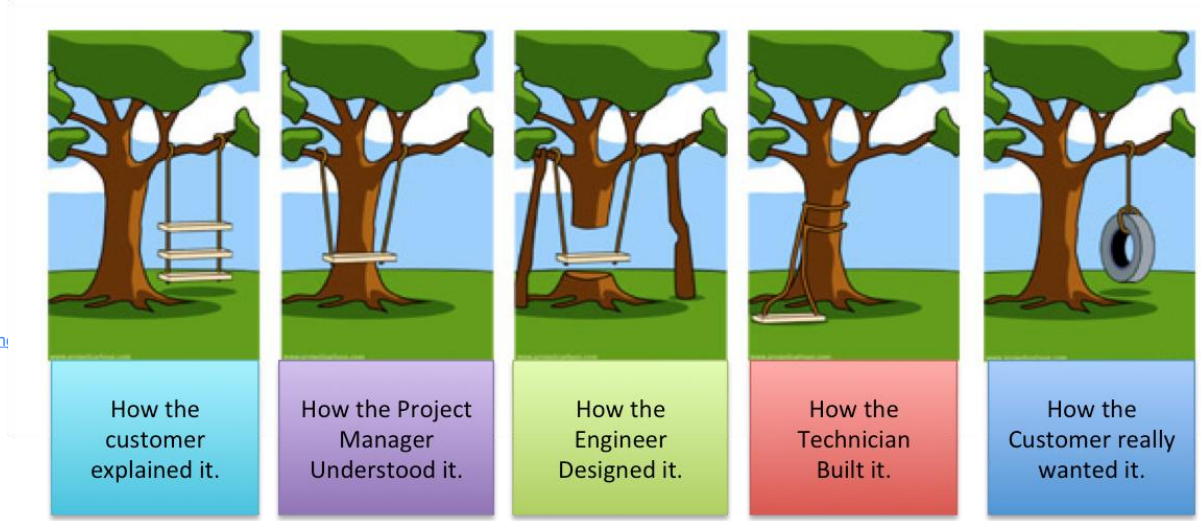
Lecture Topic

Exercise:

- OutFenix v2

Bibliography:

- OutSystems online training:
  https://www.outsystems.com/training/courses/125/logic/?Learnin



ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

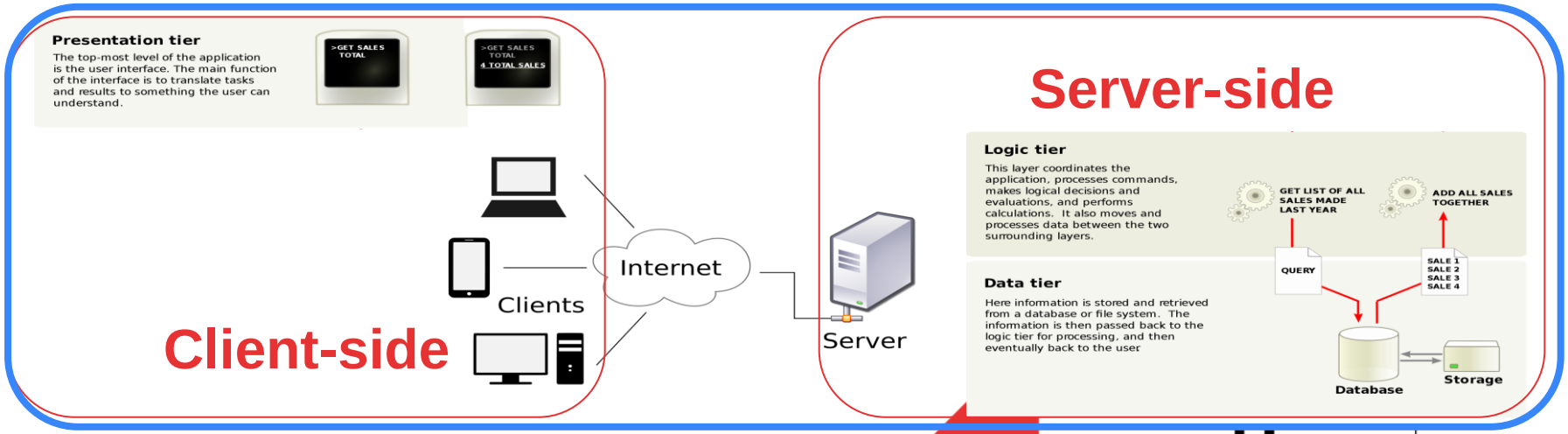
Live Coding Challenge

- Dates?
- Scope: create a functional WebApp, using the programming patterns from the OutSystems training courses and practised in the labs.
- Restrictions:
  - 1 hour to complete with tolerance until the end of the class
  - The only software allowed is ServiceStudio and the browser
  - Opening any other OutSystems exercises is NOT ALLOWED
- Grading criteria: it is more important to have all features implemented incompletely or with bugs (it will get a higher score) than to have fewer features implemented without bugs
- Material: PC in the classroom. Memorize your email and password from your OutSystems Personal Environment (or bring it written in a paper)!

outsystems

OutSystems Online Training: Becoming a Reactive Web Developer

Web Application: practical exercise OutFenix v2



ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

Step 4 - create 1 Screen: “Enrollments”

Implement the following user stories:

- As a teacher, I want to see all the students enrolled in each course so that I choose the classrooms with enough seatings
- As a student, I want to see all the courses in which each student is enrolled
- 🎖 Best implementation is from ____

Checklist:

- Use OutSystems default structure
- **What is the programming pattern for this screen?**
- In the main content:
  - Use a table
  - Drag & drop the Student and Course entity attributes
- **Adjust the look & feel, test in your smartphone**
- Link the “Courses” and “Students” Screens to this “Enrollment” one

Technical constraints:

- Only **ONE** Screen to list both “students in a course” and “the courses of a student”
- Use **ONE** list only
- Create **ONE** aggregate only to fetch from the database the courses and students requested
- Solution? The filter of the aggregate must take into consideration which input parameter was sent when the screen is called

ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

OutFenix v2

Website Architecture

Functional aspects: Add functionalities for security and grade management.

Visual appearance: no changes

Technical constraints: -

Dependencies: all in-class content and all homeworks up to and including Lesson 10

User stories:

- As a teacher, I want to insert the final grade of each student in each course enrolled so that students can know their grade
- As a teacher, I want my homepage to be the list of courses
- As a student, I want my homepage to be the list of students

Security parameters:

- Only teachers can manage students, courses and final grades data

Functional aspects:

- The grades must be in the 0 to 20 range

Step 5 - create 2 Screens: “Manage Courses” and “Manage Students”

Implement the following user stories:

- As a teacher, I want to manage the courses’ data
- As a teacher, I want to manage the students’ information

Checklist:

- Create a new Screen to insert, delete, update Courses
- **What is the programming pattern for this screen?**
- Add a link to this Screen in the “Courses” Screen, “Actions” placeholder
- Repeat the same steps for the management of Students
- Validations:
  - Student number must be higher than 1
  - Course semester must be 1 or 2
- User interface:
  - Title must give the correct context (creating or updating)
  - All actions that change data (create, update, delete) must have feedback
  - Links are for navigation. Buttons/icons are to change (create, update, delete) the data
  - Delete button/icon must have a confirmation message

Step 6 - create 1 Screen: “Student grades”

Implement the following user story:

- As a teacher, I want to insert the final grade of each student in each course enrolled so that students can know their grade

Checklist:

- Use OutSystems default structure
- **What is the programming pattern for this screen?**
- Analyze the changes needed in the database.
- Analyze the interface changes.
- Implement accordingly

Functional aspects:

- The grades must be in the 0 to 20 range

Checklist:

- **What is the programming pattern for this screen?**
- Alternative solution: batch edition in the Enrollments screen
  - Problem: Forms are only for individual records. Forms cannot be used for a list of records.
  - Solution: provide feedback using the widget *Message* or a visual hint (check/uncheck mark)

Step 7 - add security to the WebApp

Implement the following security parameter:

- Only teachers can manage students, courses and final grades data

Checklist:

- **What is the programming pattern for this screen?**
- Create the roles
- Create 1 user per role
- Set the access roles in all the Screens
- Implement the logic to implement the access restrictions wherever needed
- Solution:
    a. Create an user for the teacher: go to /Users (on your environment) create a new user and add the role Teacher. Set username = teacher and password = teacher to allow me to test
    b. Screen roles (manage students, courses, grades)
    c. Don't allow the navigation to these screens. How to check? Using a server action
    d. On the login action (Common/Login/Login) check the role for the user logged and implement the corresponding logic.

Step 8 - congratulate students with high grade

Implement the following:

- On the Enrollments screen, show a congratulation message for each grade >= 16

Step 9 (optional) - new screen “Student dashboard”

Implement a new Student Dashboard (link from the “Students” page):

- Show each grade of the student, with a different colour for grades < 10, 10 <= grade < 16, grade >= 16
- Show all courses not enrolled
- Add a button to enrol the student in a course not enrolled:
  - Caution: you will get a database error when saving.
  - Tip: use the debugger to see the values of the Enrollment record before saving it (CreateEnrollment / CreateOrUpdateEnrollment) and find out there the reason for the database error when saving.

Learning goals

Creating a Web Application in OutSystems:

- Data:
  - Creating entities and attributes on the database
  - Fetching, updating, deleting, and inserting data using Web Screens
- Logic: implementing data validation and differentiated functionality, per role.
- Presentation: lists, tables, links, forms, inputs, buttons, feedback, visual structure, images and icons. With differentiated functionality, per role.
- Security: creating secure pages and differentiated content and logic, per role.
- Reactive: developing content that adapts to screen size

Homework 

Finish the OutFenix v2 exercise:

1. All steps up to step 8. Optional: step 9.
2. For each entity, fill in some data (at least 4 records per entity)
3. Now you need to correctly set the role (“anonymous” or teacher) to all your Web Screens
4. Submit the address of your Web Application by following up on the email

Expected total effort: 30 to 90 minutes