Project of Software Development

Lab 4 - 6Mar and 8Mar
Practical Exercise 3: OutFenix

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

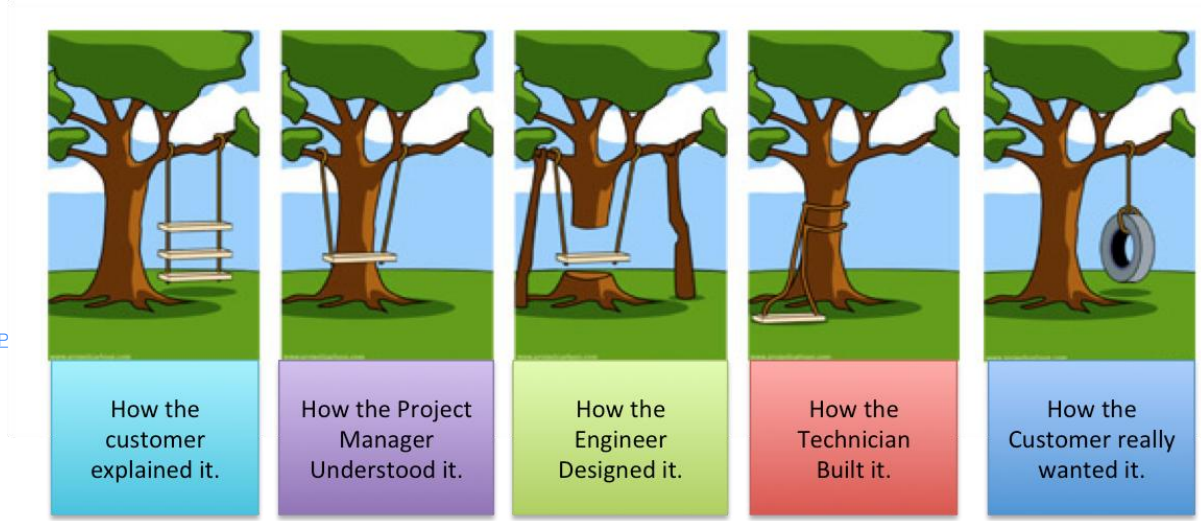
Lecture Topic

Exercise:

- OutFenix

Bibliography:

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



ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

Presentation of your business idea on EBUSI 26Mar

I might attend your EBUSI lesson to see your presentation.

I will NOT be there to evaluate you or your presentation.

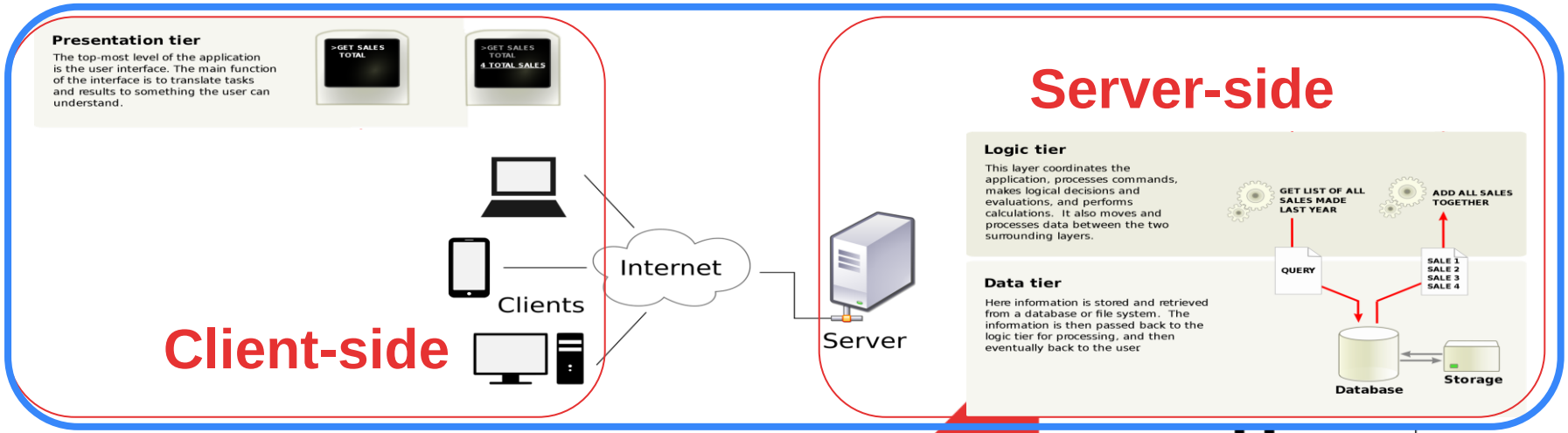
My only goal is to understand your business idea and to prepare your group projects. Don’t be afraid to think big: I will define a feasible subset for your OutSystems project.

Tip: Practice your presentation with the new *Rehearse with Coach* feature in *Microsoft PowerPoint*.

outsystems

OutSystems Online Training: Becoming a Reactive Web Developer

Web Application: practical exercise OutFenix



Lisbon School of Economics & Management Universidade de Lisboa

LISBOA | UNIVERSIDADE DE LISBOA

OutSystems exercise - OutFenix

Please download the slides from Fenix/Lab Materials from today's lab nr 4

Open it and start on page 5

OutFenix (OutSystems Fenix)

Website Architecture

Functional aspects: Create a Web App to handle Students and Courses:

- The Web App will handle courses and students
- Students are enrolled in courses
- All information is stored in the database

Visual appearance: use the Design Guidelines defined by ISEG

Technical constraints: -

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

User stories:

- As a teacher, I want to see all the courses available on each semester so that I know the subjects learnt by students
- 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 students so that I find my friends
- As a student, I want to see all the courses in which each student is enrolled

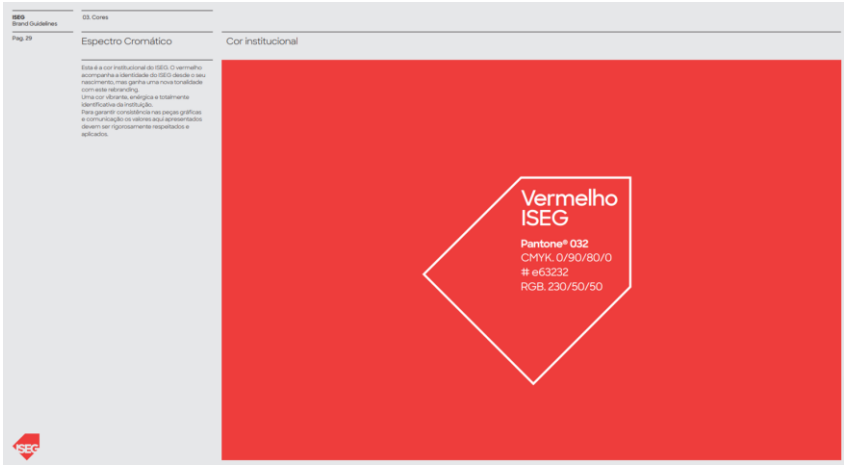
ISEG Lisbon School of Economics & Management Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

Step 1 - Create a new WebApp

Visual appearance:

1. Go to www.iseg.ulisboa.pt, (EN) *About /Media and Brand Identity* or (PT) *Sobre / Media e Identidade de Marca*,
2. Scroll to (EN) *Brand Guidelines* or (PT) *Manual de Normas Gráficas* and click on View/Download:
   a. On page 29 you have the colours
   b. On page 32 you have the fonts
3. Scroll to Logo and download the ISEG symbol
4. In Service Studio, create a new WebApp using these colors and logos



Lisbon School of Economics & Management
Universidade de Lisboa

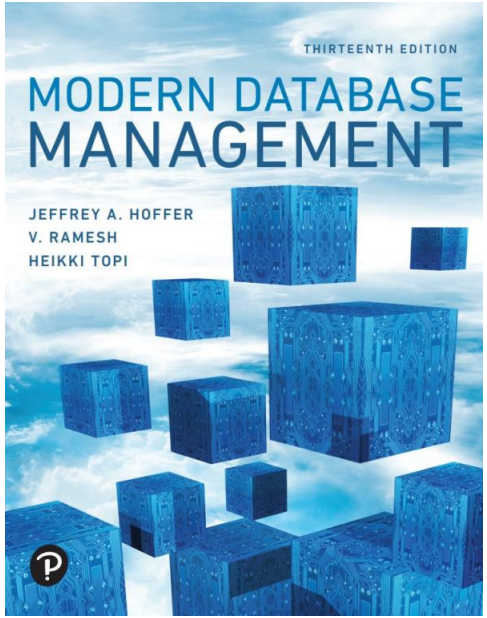
U LISBOA | UNIVERSIDADE DE LISBOA

Step 2 - Create the entities

- You will need to store courses, students, and students enrolled in courses
- All entities must be in the 3rd normal form (Hoffer, chapter 4, “Introduction to normalization, pages 176-185)
- All entities must have the keys defined (primary and foreign)
- Optional (advanced): ensure that the same student cannot be enrolled twice in the same course

Minimum requirements:

- Insert at least 4 courses, 3 students and enroll each student in at least 2 courses



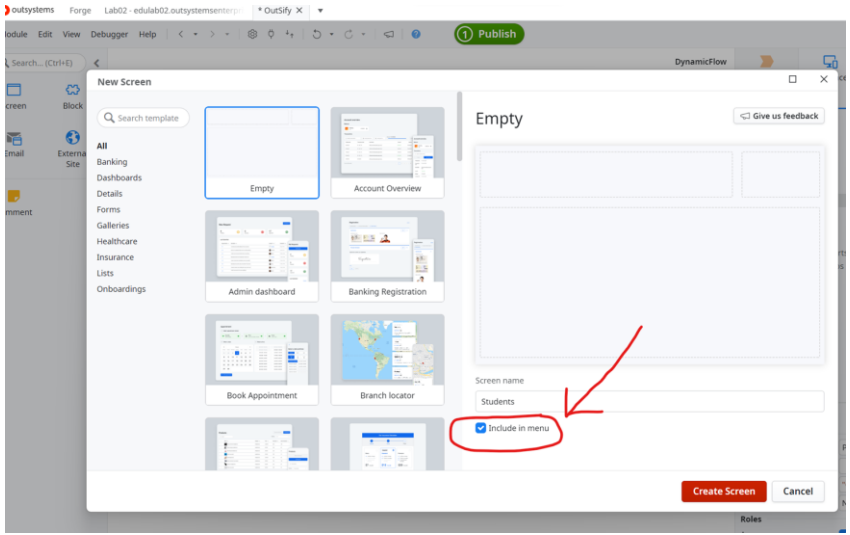
Step 3 - create 2 WebScreens: “Courses” and “Students”

Implement the following user stories:

- As a teacher, I want to see all the courses available on each semester so that I know the subjects learnt by students
- As a student, I want to see all the students so that I find my friends

Checklist:

- Use OutSystems default structure
- When creating the Web Screen, click on “Include in menu”
- **What is the programming pattern for this screen?**
- In the main content:
  - Use a list
  - Create the aggregate to fetch courses from the database
  - Drag & drop the Course entity attributes
- **Adjust the look & feel, test in your smartphone**
- Repeat the same steps above for the 2nd WebScreen for “Students”



Lisbon School of Economics & Management
Universidade de Lisboa

U LISBOA | UNIVERSIDADE DE LISBOA

Step 4 - create 1 WebScreen: “Enrollment”

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

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” WebScreens to this “Enrollment” one

Technical constraints:

- Only **ONE** WebScreen 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

Step 5 - create 2 WebScreens: “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 WebScreen to insert, delete, update Courses
- **What is the programming pattern for this screen?**
- Add a link to this WebScreen in the “Courses” WebScreen, “Actions” placeholder
- Repeat the same steps for the management of Students

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 6 - create 1 WebScreen: “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?**

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 (for the Teacher role, user username = teacher and password = teacher to allow me to test 🤓)
- Set the access roles in all the WebScreens
- Implement the logic to implement the access restrictions wherever needed

Learning goals

To understand how to create Dynamic Web pages in OutSystems

Creating a Web Application in OutSystems: Entities, fetching, updating, deleting, and inserting data using Web Screens

How to develop a Reactive WebApp with content that adapts to screen size

Homework: exercise

Finish the OutFenix exercise:

1. Mandatory: up to step 5. Optional: OutFenix v2 (steps 6 and 7)
2. For each entity, fill in some data (at least 4 rows per entity)
3. Don't forget to add the “anonymous” role 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

Homework

OutSystems Online Training: Becoming a Web Developer
https://www.outsystems.com/training/paths/18/becoming-a-reactive-web-developer/

Form Validations (45 minutes)
https://www.outsystems.com/training/courses/130/form-validations/?LearningPathId=18

Role-based Security (50 minutes)
https://www.outsystems.com/training/courses/131/role-based-security/?LearningPathId=18

Debugging in OutSystems (15 minutes)
https://learn.outsystems.com/training/journeys/web-developer-662/debugging-apps/o11/202