

LABS 5 - 7

Requirements

You will be assigned with a problem from the list below (P1, P2, P3, P4) and you are required to fulfil the following:

- Use simple feature development process
- Identify features and plan for 3 iterations
- Use layered architecture (UI, Controller, Domain, Repository)
- Data validation - when the user gives invalid inputs values, it will be notified about the invalid command (exceptions).
- Use test driven development approach. All the needed functions will be specified and tested.
- The documentation will contain the problem statement, feature list, iteration plan. For each iteration: usage scenarios, work items/tasks

P1. Students registers management

A faculty stores informations about:

- students: <studentID>, <name>
- grades: <discipline>, <studentID>, <teacher>, <grade>

Create an application which allows to:

- manage a list of students and a list of disciplines.
- add, remove, update, lists students and disciplines
- search for a student based on his/her ID and searching a discipline based on the title.
- create statistics: list of students and grades at a single discipline ordered: alphabetically, by their grades, the first 20% of students according to the average grades at all the disciplines

P2. Students lab assignments

Write an application that manages lab assignments for students at a given discipline.

The application will store:

- students: <studentID>, <name>, <group>
- assignment: <studentID>, <description>, <deadline>, <grade>

Create an application which allows to:

- manage a list of students and a list assignment.
- add, remove, update, lists students and assignments
- search for a student based on his/her ID.
- create statistics: list of students and grades at a single assignment ordered: alphabetically, by their grades, all students with the average grade lower than 5.

P3. Movie rental

Write an application for movie rental.

The application will store:

- movies: <id>, <title>, <description>, <type>, etc
- clients: <id>, <name>, <CNP>, etc

Create an application which allows to:

- manage the list of movies and clients.
- add, remove, update and list movies and clients
- search for a movie; search for a client.
- Rent/return movies
- Reporting. The reporting part of the application will allow generating a list of clients or books (maybe of certain type) ordered based on the user preference. Examples: most rented movies, most active clients, clients with rented movies ordered alphabetically, by number of rents, by year of apparition

P4. Library

Write an application for a book library

The application will store:

- books: <id>, <title>, <description>, <author>, etc
- clients: <id>, <name>, <CNP>, etc

Create an application which allows to:

- manage the list of books and clients.
- add, remove, update and list books and clients
- search for a book; search for a client.
- Rent/return book
- Reporting. The reporting part of the application will allow generating a list of clients or books (maybe of certain type) ordered based on the user preference. Examples: most rented book, most active clients, clients with rented books ordered alphabetically, by number of rents, by year of publishing, etc.

P5. Name and address book (NAB) management

In a NAB the following information may be stored:

- persons: <personID>, <name>, <phone number>, <address>
- activities: <personID>, <date>, <time>, <description>

Create an application which allows to:

- manage the list of persons and activities.
- add, remove, update and list persons and activities
- search for a person; search for an activity.
- searching a person based on his/her name, and search for an activity on a given date.
- create statistics: list of activities for a person ordered: alphabetically, by their date; list of persons having activities in a certain interval [date1, date2], ordered according to different criteria (date1, alphabetic order of the description), etc