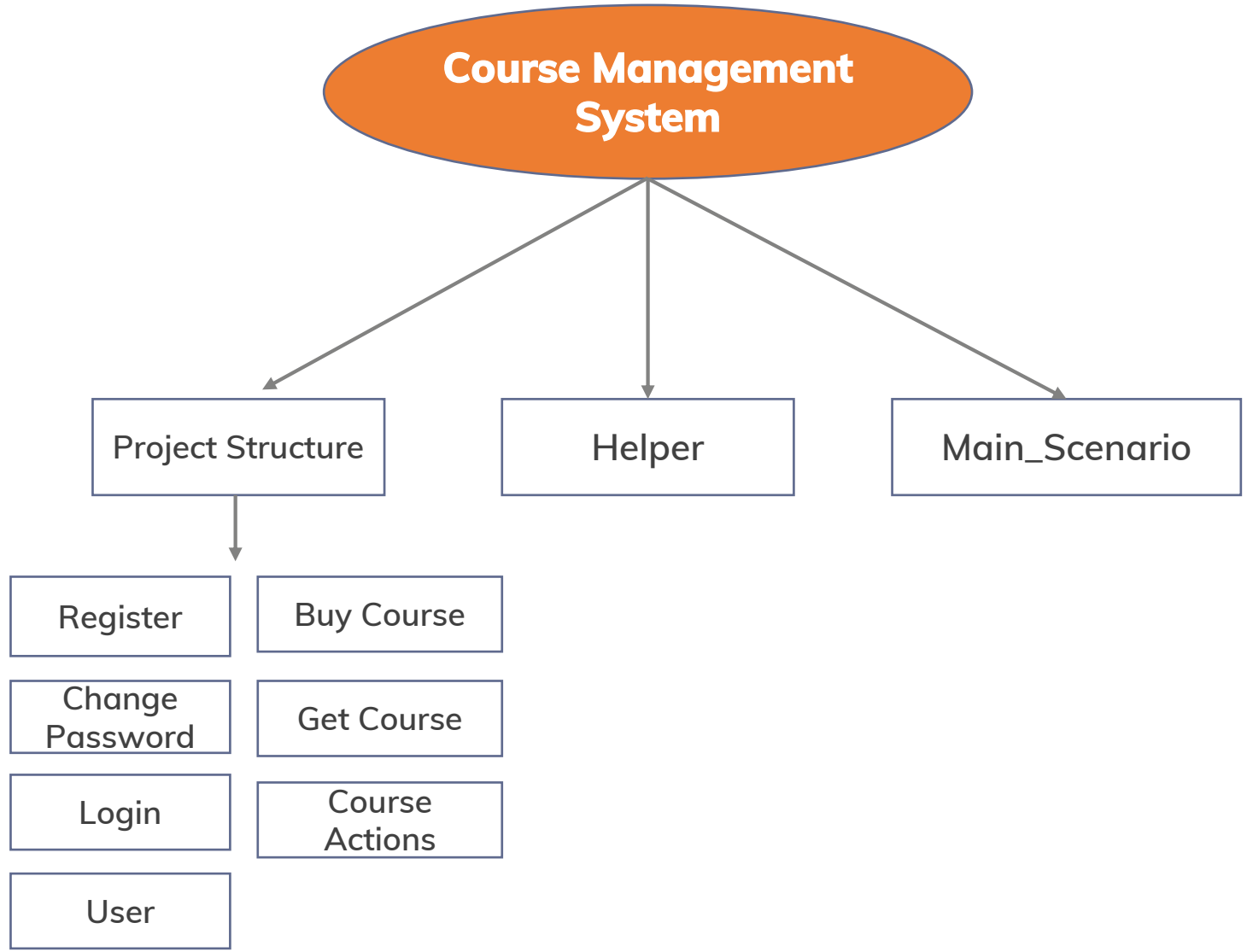
12 Workshop

Task Description

Create a Python program that simulates a Courses Management System and Data Storage.

The program should consist of several classes and methods to manage users, courses, and data files. The program should also incorporate logging to track actions in a project.log file.

Program design



Requirement implementation

QWALLITY

1. Create a **Helper** class:

This class should take **dirname**="Dataset", **user_excel_name**="users.xlsx", **course_excel_name**="courses.xlsx".
Logging should be configured during Helper object creation.

Implement the following methods:

- **create_dir**: Create dirname=<Dataset> folder, return folder path
- **create_and_write_users_excel:** Excel sheet should contain username, email, password, account_balance, user_role, logged_in columns
- **create_and_write_courses_excel:** Excel sheet should contain title, price, description, course_type(fundamental, advanced), course_buyer(empty) columns
- **read_from_excel:** Read row data from corresponding excel by column_value name.
- **clean_up:** Clean all data(directory with excel files and log file) created during code execution.

2. Create a **Register** class:
Class should get **username, email, password, account_balance,user_role** variables during initialization.

Implement the following methods:

- **check_validation:**
  username: required(8 symbol).
  email: contains @ symbol.
  password: contains at least 1 uppercase, 1 digit(8 symbol)
  account_balance: Is numeric
  user_role: "admin", "non-admin"
  Validation message: In case of data issue, should give validation message of corresponding field

- **register_user:** Open users.xlsx file, and write corresponding user data

3. Create a **Login** class:

Implement the following methods:

- **user_login**: When Log in to the system, add True under logged_in column
- **user_logout:** When Log out, change False under logged_in column

4. Create a **User** class which gets username during initialization

Implement the following methods:

- **check_user_role**: Check user role and return True in case of admin, False in case of non-admin
- **change_role:** Change given username role
- **delete_user:** Delete user by given username

5. Create an **Get_Courses** class:

Implement the following methods:

- **get_course_data_by_title**: Return row with full data from courses.xlsx by title
- **get_total_courses:** Get all rows from courses.xlsx and return total rows number

6. Create an **Change_Password** class which gets **username** and **new password** during initialization

Implement the following methods:

- **check_password**: Check given password validation
- **change_password:** Logged in user navigate to users.xlsx, find logged in user row and update password column with new given password

7. Create an **Course_Actions** class.

Implement the following methods:

- **add_course**: Add new course(row) in course.xlsx with title, price, description, course_type data
- **edit_course:** Find course with title and update given data
- **delete_course:** Find course with title and delete from courses.xlsx

8. Create a **Buy_Course** class.

Implement the following method:

buy_course: Logged in user find course with given course_name and add in the same line course_buyer

Main_scenario.py (main run):

1. Register two users: admin and non-admin.
2. Login with the admin user and add one fundamental course.
3. Login with the non-admin user, get the added course by title and buy it.
4. The non-admin user changes his/her password and login again, login should be successful
5. Admin user changes the non-admin user role to admin and delete that user.
6. Admin user delete the course and get message course is deleted.
7. Admin user is logged out from the system

Important

All necessary info should be logged in project.log

Good Luck