CoGrammar

Tutorial Session

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Lecture Objectives

- **Practice Test Driven Development using coding challenges**
- **Build a project from scratch with a professional structure**

Code Challenge

Use Red Green Refactor to solve the coding challenges

- Reverse the integer
- String to integer atoi

Take the solution code and write unit tests for it

- Caesar Cipher

Challenge Application

We have been asked to develop a mini game for a local school, they would like their students to be able to challenge each other to different educational challenges.

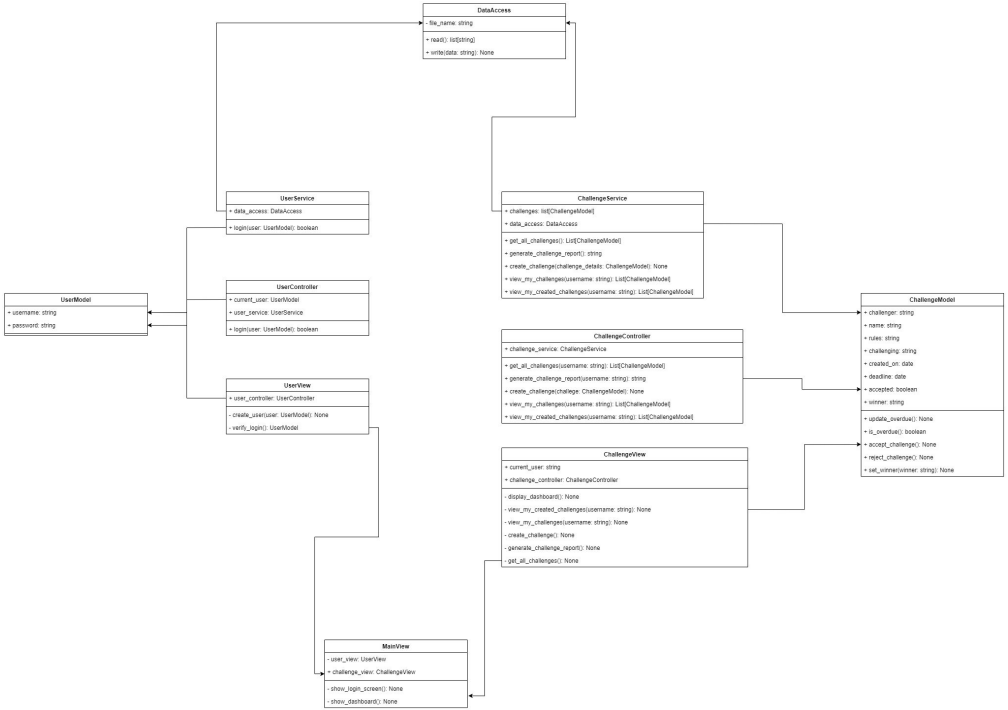
Each player will be able to challenge other players and the person making the challenge should be able to track the winner of the challenge.

File Structure

User.txt
username, password

Challenge.txt
challenger, title, rules, challenging, created on, deadline, accepted, winner

Design


DataAccess
- file_name: string
+ read(): list[string]
+ write(data: string): None
UserService
+ data_access: DataAccess
+ login(user: UserModel): boolean
ChallengeService
+ challenges: list[ChallengeModel]
+ data_access: DataAccess
+ get_all_challenges(): List[ChallengeModel]
+ generate_challenge_report(): string
+ create_challenge(challenge_details: ChallengeModel): None
+ view_my_challenges(username: string): List[ChallengeModel]
+ view_my_created_challenges(username: string): List[ChallengeModel]
UserModel
+ username: string
+ password: string
UserController
+ current_user: UserModel
+ user_service: UserService
+ login(user: UserModel): boolean
ChallengeModel
+ challenger: string
+ name: string
+ rules: string
+ challenging: string
+ created_on: date
+ deadline: date
+ accepted: boolean
+ winner: string
+ update_overdue(): None
+ is_overdue(): boolean
+ accept_challenge(): None
+ reject_challenge(): None
+ set_winner(winner: string): None
ChallengeController
+ challenge_service: ChallengeService
+ get_all_challenges(username: string): List[ChallengeModel]
+ generate_challenge_report(username: string): string
+ create_challenge(challege: ChallengeModel): None
+ view_my_challenges(username: string): List[ChallengeModel]
+ view_my_created_challenges(username: string): List[ChallengeModel]
UserView
+ user_controller: UserController
- create_user(user: UserModel): None
- verify_login(): UserModel
ChallengeView
+ current_user: string
+ challenge_controller: ChallengeController
- display_dashboard(): None
- view_my_created_challenges(username: string): None
- view_my_challenges(username: string): None
- create_challenge(): None
- generate_challenge_report(): None
- get_all_challenges(): None
MainView
- user_view: UserView
+ challenge_view: ChallengeView
- show_login_screen(): None
- show_dashboard(): None

Steps

1. **Create virtual environment**
2. **Create .gitignore file**
    a. **Include** .venv **and** __pycache__
3. **Initialize Git**
4. **Link to GitHub repository**
5. **Perform steps shown on GitHub Repo page**
6. **Set up Flake8 in Visual Studio**
7. **Create file structure**
8. **Write code**