EECS 2311: Iteration 2, New features, refactoring, testing

Project Name: MyGamingList
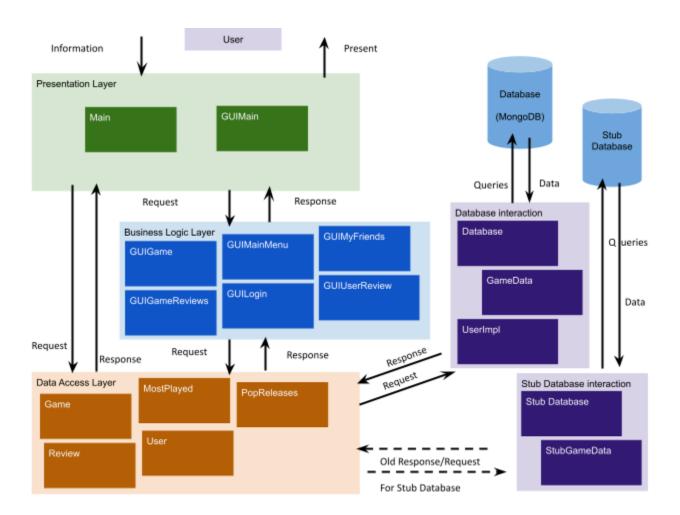
Group 5

Date: March 11, 2024

Names:
Syed Mustafa Jamal
Richard Carmichael
Alex Fulop-Boca
Youwei Qin
Rocco Fernando Gadista
Abdelrahman Merdan

Updated Plan:

Things that changed were:
- Big User Story 3 is now *View Game Statistics*. Part of Friends is in iteration 2. Friend Reviews will be integrated into the next iteration.

After talking to the client, we felt that this decision would suit the client's needs as they would like to know more about the game stats. We felt that this should also include their friends' reviews as well, as it's part of community statistics.

Big User Story:

2. Rate and Comment: As a user, I would like to be able to rate the video games I play. I would like to leave a comment on a game to explain how I feel. These comments should be shared with people who are wanting to know more about the game. Priority - High Cost - 4 days
3. View Game Statistics As a user, I would like to view various stats of a game to be more informed Priority: Medium Cost: 6 days

Iteration 2 Detailed User Stories:

Account I'd like to create an account so that I can review games and engage the community. Priority-High Cost: 1 day Youwei
Write and View Other User Reviews I'd like to write reviews for games, and view other user reviews. Priority- High Cost: 2 days Abdelrahman Merdan
React to User Reviews I'd like to comment and engage in civilized discussion with other users. Priority-Low Cost: 1 day Alex

My Reviewed Games
I'd like to see the list of games I've reviewed, and view my reviews. Priority - High Cost: 1 day Rocco
Friends
I'd like to be able to add a friend and remove friends to My Friends. Priority-High Cost: 1 day Mustafa

Iteration 3 Detailed User Stories:

View Friend Reviews
I'd like to be able to see the reviews my friend has done Priority-High Cost: 1 day Mustafa
View Game Statistics
I'd like to view the stats of various games. Priority-Medium Cost: 2 days

More detailed user stories to be discussed in next iteration.

Database classes

<<interface>>
StubDataBase

- CONNECTION : String = mongodb+srv://2311team5:qn431J86d7xMEdpc@gaminglist.a2jzcoi.mongodb.net/?retryWrites=true&w=majority&appName=GamingList
- ~SERVERAPI : ServerApi.builder()
- ~SETTINGS : MongoClientSettings =MongoClientSettings.builder()
- ~_CLIENT : MongoClient = MongoClients.create(settings)
- ~_DATABASE : MongoDatabase : client.getDatabase("MyGamingList"

StubGameData

+ changeConnection () : void
+ noAppExists (id : int) : boolean
+ noAppDetails (id : int) : boolean
+ noAppReviews (id : int) : boolean
+ updateAppDetails (id : int) : void
+ removeApp (id : int) : void
+ addApp (id : int) : void
+ getName(id : int) : String
+ getGame(id : int) : Game

<<interface>>
Database

- CONNECTION : String =
"mongodb+srv://2311team5:qn431J86d7xMEdpc@mygaminglist.igshqok.mongodb.net/?retryWrites=true&w=majority&appName=MyGamingList"
- ~SERVERAPI : ServerApi.builder()
- ~SETTINGS : MongoClientSettings =MongoClientSettings.builder()
- ~_CLIENT : MongoClient = MongoClients.create(settings)
- ~_DATABASE : MongoDatabase : client.getDatabase("MyGamingList"

MongoDB

GameData

+ games : MongoCollection<Document> = database.getCollection("Games")
+ MAP : Objectmapper : new ObjectMapper();

+ noAppExists(id : int) : boolean
+ noAppDetails(id : int) : boolean
+ noAppReviews(id : int) : boolean
+ updateAppDetails(id : int) : void
+ removeApp(id : int) : void
+ addApp(id : int) : void
+ getName(id : int) : String
+ getGame(id : int) : Game
+ updateShortDesc(jsonResponce : JsonNode) : Bson
+ updateDesc(jsonResponce : JsonNode) : Bson
+ updateCover(jsonResponce : JsonNode) : Bson
+ updateSysRequire(jsonResponce : JsonNode) : Bson
+ updateMetacritic(jsonResponce : JsonNode) : Bson
+ updateReview() : Bson
+ updateComment() : Bson

Json imports

UsersImpl

- USERS : MongoCollection<Document>
- TABLE_NAME : String = "Users";
- USER_KEY : String = "username";
- PWD_KEY : String = "password";
- GAMES_KEY : String = "games";
- FRIENDS_KEY : String = "Friends";

+ UsersImpl()
+ createAccount(user : User) : boolean
+ login(user : User) : String
+ updateFriend(username : String, friendName : String, operation : String) : void
+ listFriend(username : String) : List<String>
+ get(username : String) : User
+ delete(username : String) : boolean
- documentToUser(doc : Document) : User
- userToDocument(user : User) : Document

User

- USERNAME : String
- PASSWORD : String
- games : List<String>
- friends : List<String>

User is part of src but I included it for easier viewing

+ User (USERNAME : String, PASSWORD : String)
+ User (USERNAME : String, PASSWORD : String, GAMES : List<String>, FRIENDS : List<String>)
+ getUsername() : String
+ getPassword() : String
+ getGames() : List<String>
+ getFriends() : List<String>

Jackson

Uses bson Doc part of MongoDB

Uses · Extends

Description for UML:

The database UML features all of our Java files in the database folder. We have a collection of 6 classes, with the UserImpl accessing the real User database. The User class, (belonging to the class folder), is being added for clarity. The Database is an interface for the GameData and UserImpl which has fields for the connection to the MongoDB. The GameData houses the code to deal with the Game database. And we have the stubGameData extending the GameData since from there, we change the connection to the StubDataBase, where we can deal with and access the stub database. I also felt that I should separate the database and the UML class diagram since putting the two images together would have made the files hard to read and clunky.

Source Classes

Mutiple classes use MongoDB, Jackson and Json so for the sake of clarity, I did which classes use them

Review

- CONNECTION : String = "mongodb+srv://2311team5:qn431J86d7xMEdpc@mygaminglist.igshqpk.mongodb.net/?retryWrites=true&w=majority&appName=MyGamingList"
- SERVERAPI : ServerApi.builder()
- SETTINGS : MongoClientSettings =MongoClientSettings.builder()
- CLIENT : MongoClient = MongoClients.create(settings)
- DATABASE : MongoDatabase : client.getDatabase("MyGamingList")
- GAMES : MongoCollection<Document> = database.getCollection("Games")
- USERS : MongoCollection<Document> = database.getCollection("Users")

+ UserExists(name : String) : boolean
+ noAppReviews(id : int) : boolean
+ CheckUpdateGame(id : int) : boolean
+ review_game(username : String, game : Game, review : int, comment : String, reccomendation : String) : void
+ UpdateNumReview(game : Game) : Bson
+ UpdateSumOfRevies(review : int, game : Game) : Bson
+ UpdateComment(Review(game : Game, username : String, review : int, comment : String, reccomendation : String) : Bson
+ find_user(username : String) : Document
+ find_game(game : Game) : Document
+ addUserReview(game : Game, username : String, review : int, comment : String, reccomendation : String) : Bson
+ getAllComments(game : Game) : List<Object>
+ getAllUsersForComments(game : Game) : List<Object>
+ retrieveIndex(user User, game :Game) : int
+ getComment (userWithReccomandation : User , game : Game) : Object
+ getuserForComment(userWithReccomandation : User , game : Game) : Object
+ addCommentToUserReview(user : User, message : String, userWithTheReview : User, game :Game) : void

User

- USERNAME : String
- PASSWORD : String
- games : List<String>
- friends : List<String>

+ User (USERNAME : String, PASSWORD : String)
+ User (USERNAME : String, PASSWORD : String, GAMES : List<String>, FRIENDS : List<String>)
+ getUsername() : String
+ getPassword() : String
+ getGames() : List<String>
+ getFriends() : List<String>

MostPlayed

+ids : ArrayList<Integer> = new ArrayList<>()

+ MostPlayed()
+ getID (i : int) : int
+ getIDs() : int : ids
+ getSize() : ids.size : int
+ toString() : ids.toString() : String

PopReleases

+ids : ArrayList<Integer>

+ PopReleases()
+ getID (i : int) : int
+ getIDs() : int : ids
+ getSize() : ids.size : int
+ toString() : ids.toString() : String

Main

+main(args : String[])

Game

- id : int
- sum_of_all_reviews : int
- num_of_reviews : int
- name : String
- shortDesc : String
- desc : String
- cover : String
- sysRequire : String
- metaScore : String
- metaURL : String
- comments : List<Object>

+ Game(id : int, sum_of_all_reviews : int, num_of_reviews : int, name : String, shortDesc : String, desc : String, cover : String, sysRequire : String, metaScore : String, metaURL : String, comments : List<Object>)
+ setName(name : String) : void
+ setID(id : int) : void
+ setShortDesc(shortDesc : String) : void
+ setDescription(desc : String) : void
+ setCover(cover : String) : void
+ setSysRequire(sysRequire : String) : void
+ setMetaScore(metaScore : String) : void
+ setMetaURL(metaURL : String) : void
+ setSumOfAllReviews(SumOfReviews : int) : void
+ addComment(comment : List<Object>) : void

+ getName() : name : String
+ getId() : id : int
+ getDescription() : description : String
+ getSysRequire() : sysRequire : String
+ getMetaScore() : String
+ getMetaURL() : String
+ getSumOfAllReviews() : int
+ getNumOfReviews() : int
+ getComment() List<Object>

+ toString() : name : String
+ equals(obj : Object) : boolean
+ hashCode() : int

GUIGame

-gameTitleLabel : JLabel
-gameImageLabel : JLabel
-bigDescriptionText : JEditorPane
-descriptionText : JEditorPane
-reviewGameButton : JButton
-criticReviewButton : JEditorPane
-criticReviewLabel : JButton

+ GUIGame(cardPane : JPanel)
+ loadGame(game : Game)

JDialog

GUINewUserReview

+ GUINewUserReview (game : Game)

GUITest not included since it is unused

GUIMyFriends

- users : UserImpl

+ GUIMyFriends(cardPane : JPanel, usernameLoggedIn : String)

Uses UserImpl

JPanel

JFrame

GUIMainMenu

+ GUIMainMenu(cardPane : JPanel, mostPlayed : MostPlayed, popReleases : PopReleases)

GUIMain

-basePane : JPanel
-cardPane : JPanel
-mainPane : JPanel
-loginButton : JButton
-usernameLoggedIn: String = null

+ GUIMain()

GUILogin

- passwordField : JPasswordField
- usernameField : JTextField
- newUsernameField : JTextField
- newPasswordField : JTextField
- confirmPasswordField : JTextField
- mainPane : JPanel
- usernamePrompt : String = "Username"
- passwordPrompt : String = "Password"
- confirmPasswordPrompt : String = "Confirm Password"

+ GUILogin()
+ getMainPane () : JPanel
+ login () : void
+ reset () : void

GUIGameReviews

- SERIALVERSIONUID : long = 1L
- reviewBox : Box
- reviewTitleLabel : JLabel
- backLocation : String
- buttonPanel : JPanel

+ GUIGameReviews(cardPane : JPanel)
+ loadGameReviews(cardPane : JPanel, game : Game) : void
+ loadUserReviews(cardPane : JPanel, user : String) : void
- generateScrollable(box : Box)
- review() : void
- comment(commentBox : Box): void

Extends (multiple relationships shown)

Description for UML:

We have quite a bit of class in the UML source diagram, but a majority of them are GUI's, which from the title, are self-explanatory with the functions, and purposes. We have a User class, which will hold our User object and the same purpose. We have the Review class, which deals with reviews and comments to reviews for games. Most Played is a class that deals with getting the most played games and the same idea applies to the games in the PopReleases, where it gets the very popular games. The main class is the class we use for running the application.