RateMyPlace 🏠

Technical Documentation

Software Engineering CS4320
https://github.com/Djhyv2/RateMyPlace.git

Group 13: Connor Fitzmaurice, Dustin Hengel, Mercy Housh, Madison Williams, Bradley Worthen

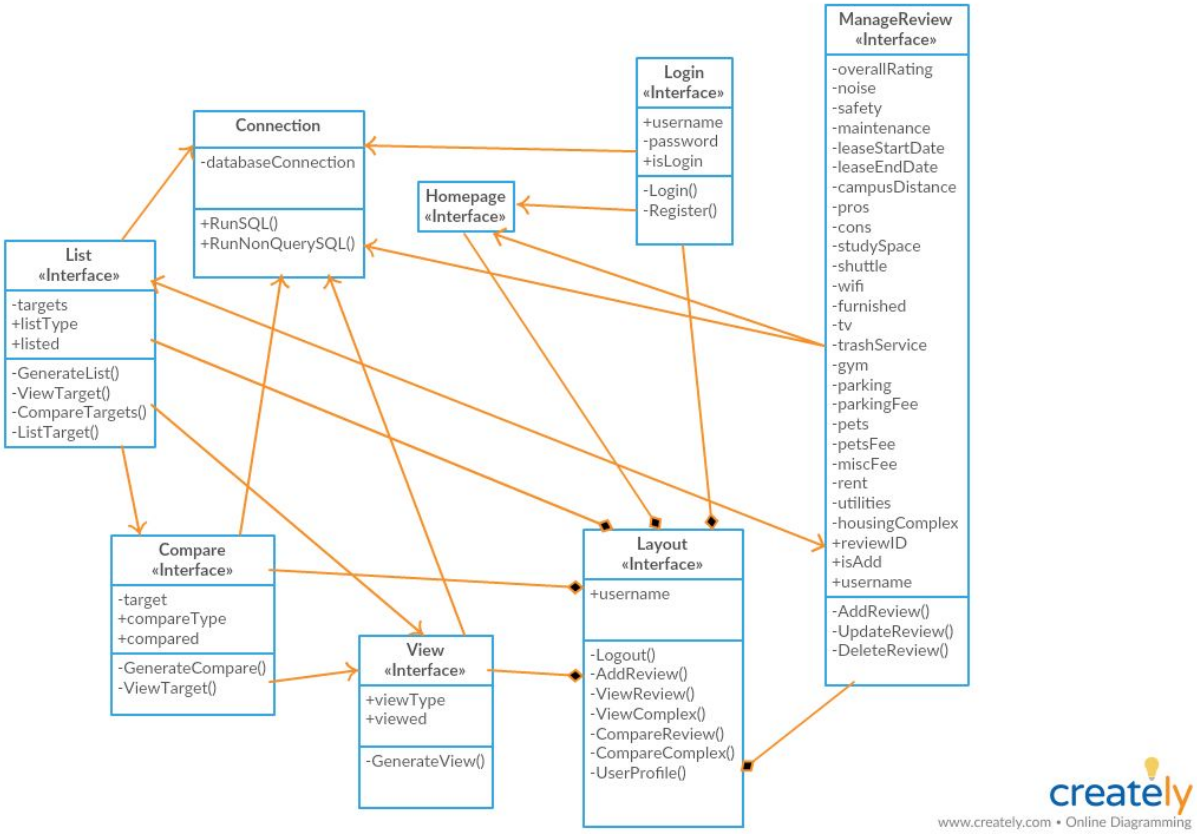
Table of Contents

1. Overall Code Discussion
2. Class Diagram
3. Class Descriptions/Data Types/Operations

1. **Overall Code Description**

In creating this code, the two largest references were the class diagram and the descriptions, data types, and operations done by each class. This was the most integral to our coding. For our actual code, we used C# with the ASP.NET framework. Thus we were able to closely mirror the object oriented design principles we utilized in the planning portions of the project. The common trend amongst most of the classes is that the functions labeled as Generate(Classname) became the Page_Load function as defined by ASP.Net. These then contain a switch statement that switches based on a GET variable called Page which is what was initially called listType, compareType, viewType, isLogin, isAdd, etc. All of these variables that existed to define what would be displayed were consolidated into this GET variable. A large advantage of the ASP.Net Framework was the close ties that events had with server-side functions. The vast majority of functions defined in our class diagram should be recognizable to a reader, but will most often be along the lines of btn(Function)_click as defined by the framework. None of our code should be unfamiliar to anyone looking at our class diagram or descriptions, but is tweaked in such a way as to maximize our chosen language and framework without sacrificing the object oriented principle we had in mind during design.

2. **Class Diagram**



3. Class Descriptions/Data Types/Operations:

Class	Type/Signature	Description
List: List's main goal to is to form lists of information to display.	-targets: string[]	The review ids or complex names associated with the selected list elements.
	+listType: string	The type of list to be displayed. Will either be "UserReviews", "AllReviews", "AllComplexes", or "ComplexReviews"
	+listed: string	The userID or complex name associated with the current listType
	-GenerateList()	Will generate and display the list based on current listType and listed.
	-ViewTarget(targets: string[])	Will redirect to the view page and will pass the current selection to set what will be viewed.

	-CompareTargets(targets: string[])	Will redirect to the compare page and will pass the current selection to set what will be compared.
	-ListTarget(targets: string[])	Will redirect to the list page and will pass the current selection to set what will be listed.
Compare: Compare's main goal is to compare multiple pieces of information.	-target: string	The review id or complex name associated with the selected compared.
	+compareType: string	The type of item to be compared. Will either be "Complexes" or "Reviews"
	+compared: string	The review IDs or complex names associated with the current compareType
	-GenerateCompare()	Will generate and display the compared items based on current compareType and compared.
	-ViewTarget(target: string)	Will redirect to the view page and will pass the current selection to set what will be viewed.
View: View's main goal is to display a single piece of information.	+viewType: string	The type of item being viewed, will either be, "Complex" or "Review"
	+viewed: string	The review ID or complex name of the item being viewed.
	-GenerateView()	Will generate and display the currently viewed item based on current viewType and viewed.
Layout: Layout's main goal is to provide an overarching user interface and fluid transitions between	+username: string	The username of the currently logged in user, if applicable
	-Logout()	Logs out the current user specified in username
	-AddReview()	Initiates the add review process, redirects to ManageReview.
	-ViewReview()	Initiates the view review process, redirects to List.

functions of the system.	-ViewComplex()	Initiates the view complex process, redirects to List.
	-CompareReview()	Initiates the compare review process, redirects to List.
	-CompareComplex()	Initiates the compare complex process, redirects to List.
	-UserProfile()	Initiates the process to view current users reviews, redirects to list.
Manage Review: ManageReview's main goal is to add, update, and delete reviews. It performs the majority of the non select calls to the review storage.	-overallRating: int	All of these variables come from the form displayed to add or edit a review. They represent basic attributes about the housing complex being reviewed. Their names should be self explanatory when viewed in the context of reviewing a housing complex.
	-noise: int	
	-safety: int	
	-maintenance: int	
	-leaseStartDate: Date	
	-leaseEndDate: Date	
	-campusDistance: float	
	-pros: string	
	-cons: string	
	-studySpace: bool	
	-shuttle: bool	
	-wifi: bool	
	-furnished: bool	
	-tv: bool	
	-trashService: bool	
	-gym: bool	
	-parking: bool	
	-parkingFee: float	
	-pets: bool	

	-petsFee: float	
	-miscFee: float	
	-rent: float	
	-utilities: float	
	-housingComplex: string	
	+isAdd: bool	Indicates whether the form is being displayed to add or edit a review
	+reviewID: int	The id of the review being edited if applicable.
	+username: string	The current logged in user if applicable
	-AddReview()	Adds the review currently in the displayed form to the database.
	-UpdateReview()	Updates the database entry for the form currently being displayed.
	-DeleteReview()	Deletes the database entry for the form currently being displayed.
Login: Login's main goal is to provide functionality for logging into or registering to the software.	+username: string	The username of the person attempting to or who has successfully logged in.
	-password: string	The password of the person attempting to login.
	+isLogin: bool	Indicates whether a user is logging in or registering.
	-Login()	Attempts to login a user based on the credentials in the displayed form and checks them with the database.
	-Register()	Attempts to create a database entry for the user trying to register.
Connection: Connection's main goal is to provide a	-databaseConnection: string	The string used to connect to the database
	+RunSQL(sql: string, parameters: SQLParameters[]): DataTable	Executes SQL from the passed in string and parameters, returns the selected

central point for all database calls to originate from.		datatable from the database.
	+RunNonQuerySQL(sql: string, parameters: SQLParameters[]): int	Executes SQL from the passed in string and parameters. Used for non-select statements. Returns the amount of rows affected.