**Модели**

На основе описанных моделей была создана база данных, а по ней создана ER-диаграмма.

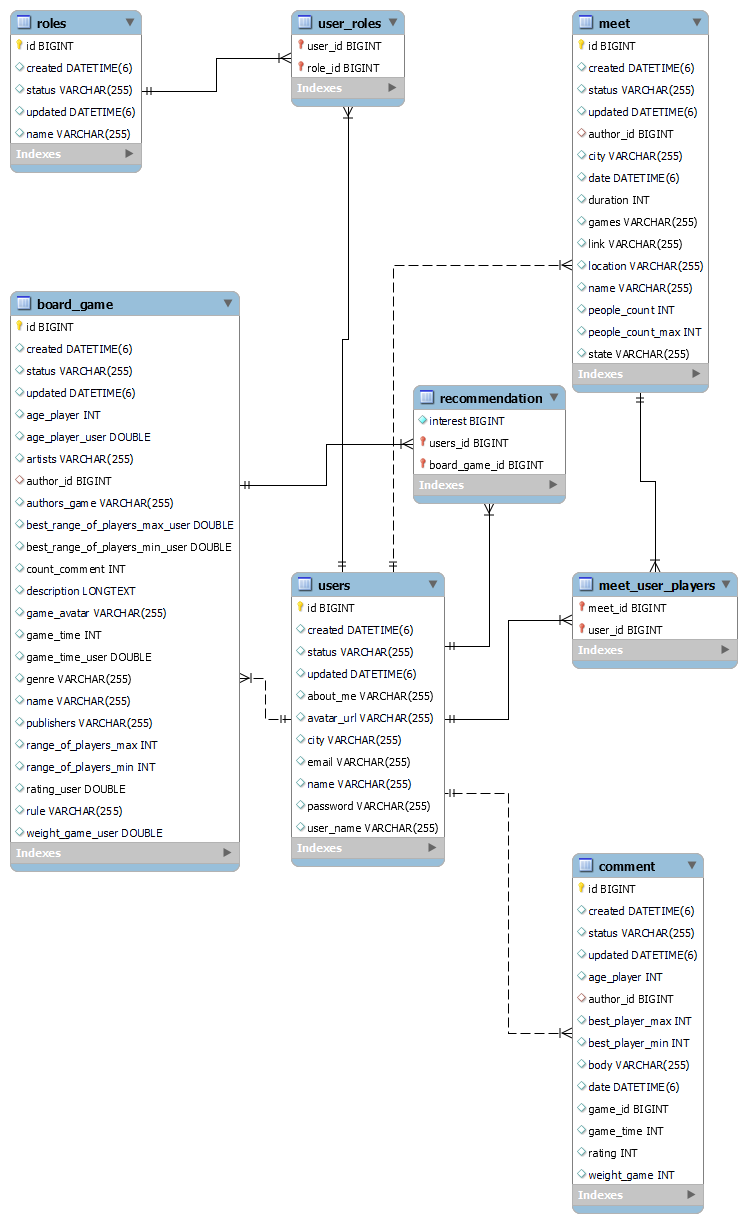


Рисунок 1 ER-диаграмма

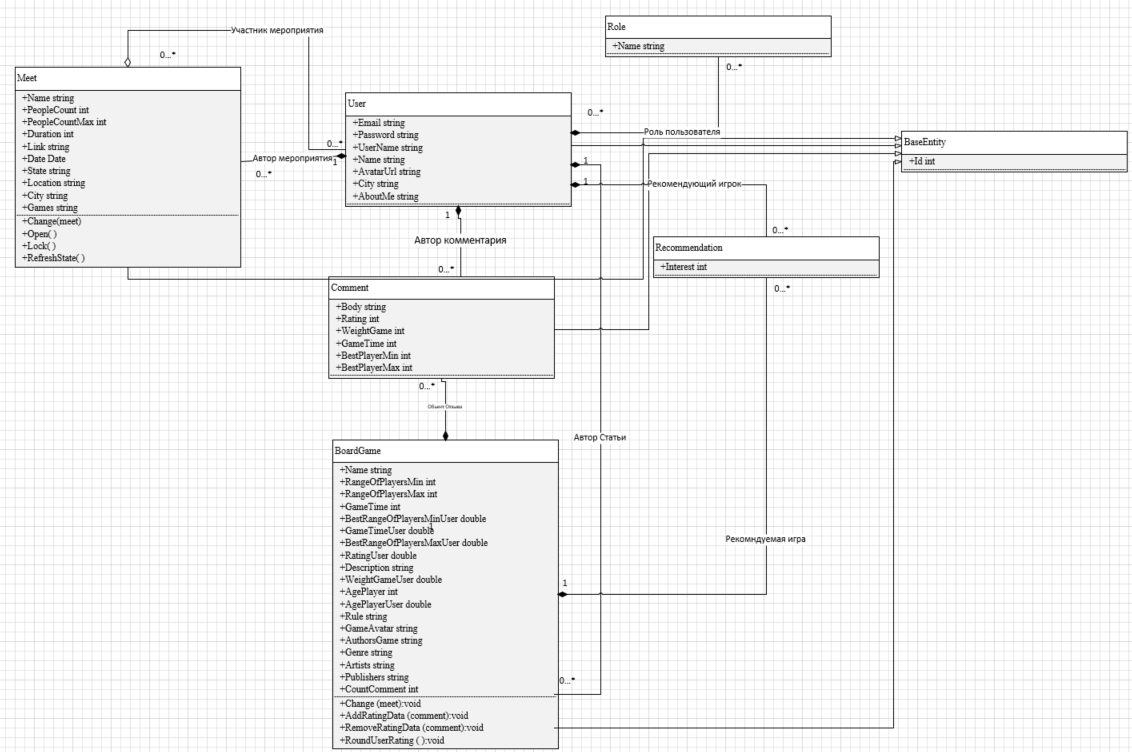


Рисунок 2 Схема моделей

В системе существует 6 моделей:

1. Meet,
2. User,
3. Comment,
4. BoardGame,
5. Role,
6. Recomendation,

Все эти модели кроме Recomendation наследуются от BaseEntity, которая содержит одно поле идентификатора

Meet это модель мероприятия, которая содержит информацию о месте, времени, количестве человек, игр и т.д. Так же модель связана с моделью пользователем, отношением один-ко-многим «Автор Мероприятия» и отношением многие-к-многим «Участник Мероприятия».

Класс модели мероприятия содержит следующие поля:

1. Name – хранит название мероприятие,
2. PeopleCount – хранит текущее количество участников мероприятия,
3. PeopleCountMax – хранит максимальное количество участников мероприятия,
4. Duration – хранит предположительную продолжительность мероприятия в минутах,
5. Link – хранит ссылку для связи между пользователями на сторонний сервис
6. Date – хранит дату и время проведения мероприятия,
7. State – хранит состояние мероприятия, то есть информацию о том начато ли мероприятие или еще ждет идет набор участников,
8. Location – хранит информацию о месте проведения мероприятия,
9. City – хранит информацию о городе проведения мероприятия,
10. Games – хранит список игр в которые будут играть на мероприятие,

так же класс мероприятия хранит следующие методы:

1. Change – Метод для изменения мероприятия,
2. Open – Метод для открытия мероприятия для набора пользователей,
3. Lock – Метод для закрытия мероприятия для набора пользователей,
4. RefreshState – Метод который обновляет состояние мероприятия

User это модель пользователя, которая содержит информацию о пользователи, такая как почта, хеш пароля, город проживания и т. д.

Класс модели пользователя содержит следующие поля:

1. Email – хранит электронную почту пользователя,
2. Password – хранит хеш пароля пользователя,
3. UserName – хранит имя используемое для входа в система,
4. Name – хранит ФИО пользователя,
5. AvatarUrl – хранит ссылку на фотографию хранящуюся на сервере,
6. City – хранит город в котором живет пользователь,
7. AboutMe – хранит информацию о пользователе, которую пользователь пишет о себе

модель пользователя не содержит методов.

Comment это модель комментария к настольной игре, которая содержит дату, оценки,текст комментария, так же эта модель связана с моделью пользователя отношением многие-ко-одному «Автор комментария» и связью многие-ко-одному «Объект отзыва» с моделью настольной игры.

Класс модели комментария содержит следующие поля:

1. Body – хранит текст комментария,
2. Rating – хранит общий рейтинг игры оставляемый пользователем,
3. WeightGame – хранит сложность игры по мнению пользователя,
4. GameTime – хранит время игры по мнению пользователя,
5. BestPlayerMin – хранит минимальное количество игроков для игры по мнению пользователя,
6. BestPlayerMax – хранит максимальное количество игроков для игры по мнению пользователя

Класс комментария не содержит методов.

BoardGame это модель настольной игры, содержащий описание настольной игры, информация о рейтинге игры и т. д. Модель настольных игр связанна с моделью пользователей связью многие-к-одному «Автор игры».

Класс модели настольной игры содержит следующие поля:

1. Name – хранит название настольной игры,
2. RangeOfPlayersMin – хранит минимальное количество игроков для игры определяемое автором,
3. RangeOfPlayersMax – хранит максимальное количество игроков для игры определяемое автором,
4. GameTime – хранит информацию о длительности игры, определяемое автором,
5. GameTimeUser – хранит информацию о длительности игры, определяемое пользователями,
6. BestRangeOfPlayersMinUser – хранит минимальное количество игроков для комфортной игры определяемое пользователями,
7. BestRangeOfPlayersMaxUser – хранит максимальное количество игроков для комфортной игры определяемое пользователями,
8. RatingUser – хранит рейтинг игры определяемый пользователями,
9. Description – хранит описание настольной игры,
10. WeightGameUser – хранит сложность игры, определяемое пользователями,
11. AgePlayer – хранит информацию о рекомендуемом возрасте игроков определяемое автором,
12. AgePlayerUser – хранит информацию о рекомендуемом возрасте игроков определяемое пользователями,
13. Rule – хранит ссылку на файл с правилами на сервере,
14. GameAvatar – хранит фотографию настольной игры с правилами на сервере,
15. AuthorsGame – хранит имена авторов игры,
16. Genre – хранит жанр игры,
17. Artists – хранит имена художников игры,
18. Publishers – хранит название издательства выпускающего игру,
19. CountComment – хранит количество оставленных комментариев.

Класс модели настольной игры содержит следующие методы:

1. Change – метод изменяющей объект мероприятия,
2. AddRatingData – метод изменяет рейтинг игры при добавление комментария,
3. RemoveRatingData – метод изменяет рейтинг игры при удаления комментария,
4. RoundUserRating — метод округляющий значения рейтинга для вывода в пользовательский интерфейс.

Role это модель роли используемая авторизации различных пользователей она содержит лишь одно поле Name, содержащее название роли, и связана с моделью пользователя многие-ко-многим «Роли Пользователя»

Recommendation это модель рекомендации используемая в рекомендательной алгоритме, и содержащая одно поле Interest которое показывает интерес пользователя к конкретной игре. Модель рекомендации связана с моделью пользователя и настольной игры связями многие-к-одному «Рекомендующий игрок», «Рекомендуемая игра» соответственно.

**Слой репозиториев**

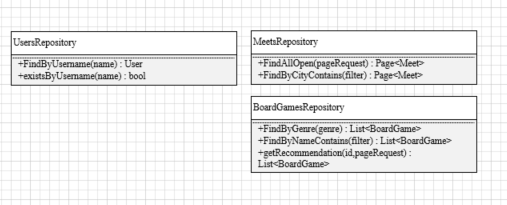
****

Рисунок 3 Слой репозиториев

В системе содержатся 6 репозиториев все они работают с соответствующими моделями. Репозитории должны обеспечивать доступ к моделям в базе данных их в системе 6:

1. BoardGameRepository,
2. UserRepository.
3. MeetRepository,
4. RecommendationRepository,
5. CommentRepository,
6. RoleRepository,

BoardGameRepository содержит 3 метода:

1. FindByGenre – метод по переданному жанру выдает список настольных игр,
2. FinbByNameContains – метод по переданной строке имени ищет все настольные игры в название которых содержится переданная строка,
3. getRecommendation – метод выдает рекомендованные пользователю игры.

UsersRepository содержит 2 метода:

1. FindByUsername – метод отдает пользователя с определенным полем username
2. existsByUsername – метод указывает если в базе данных пользователь с определенным полем username

MeetRepository содержит 2 метода:

1. FindAllOpen – метод выдает все мероприятия состояние которых позволяет вступить в них пользователям
2. FindByCityContains – метод выдает мероприятия в которых название города содержит переданную подстроку.

RecommendationRepository содержит метод findByUserAndBoardGame который выдает строку рекомендации по идентификаторам пользователя и настольной игры.

CommentRepository и RoleRepository не содержат логики кроме как логики получения моделей по идентификатору.

**Слой сервисов**

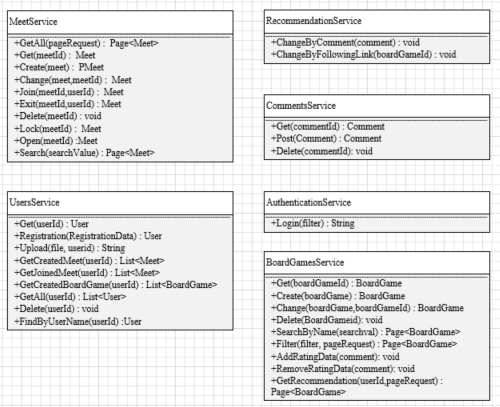
****

Рисунок 4 Слой сервисов

В системе содержится 6 cервисов 5 из них предоставляют логику работы с сущностями системы такие как:

1. BoardGameService,
2. CommentService,
3. MeetService,
4. RecommendationService,
5. UserService,

а также AuthenticationService предоставляет логику работы с авторизацией в системы.

BoardGameService содержит 9 методов:

1. Get – метод выдающий настольную игру по идентификатору,
2. Create – метод который создает объект настольной игры,
3. Change – метод который изменяет объект настольной игры,
4. Delete – метод который удаляет настольную по идентификатору,
5. SearchByName – метод который выдает страницу настольных игр в именах, которых есть передаваемая подстрока,
6. Filter – метод который выдает страницу настольных игр отфильтрованных и отсортированных по переданным правилам
7. AddRatingData – метод изменяющийся рейтинг игры относительно добавленного комментария,
8. RemoveRatingData – метод изменяющийся рейтинг игры относительно удаляемого комментария,
9. GetRecommendation – метод выдает страницу рекомендованных игр для определенного пользователя.

CommentService содержит 3 метода:

1. Get – метод, который выдает комментарий по идентификатору,
2. Post – метод, который создает объект комментария,
3. Delete – метод, который удаляет объект комментария по определенному идентификатору.

MeetService содержит 10 методов:

1. GetAll – метод, который выдает страницу с мероприятиями,
2. Get – метод, который выдает мероприятию по идентификатору,
3. Create – метод, который создает объект мероприятия,
4. Change – метод, который изменяет объект мероприятия,
5. Join – метод, который включает игрока в определенное мероприятие,
6. Exit – метод, который исключает игрока из определенного мероприятия,
7. Delete – метод, который удаляет мероприятие с определенным идентификатором
8. Lock – метод, который блокирует набор на мероприятие,
9. Open – метод, который открывает набор на мероприятие,
10. Search – метод, который выдает страницу мероприятий содержащих в название переданную подстроку,

RecommendationService содержит 2 метода:

1. ChangeByComment – метод, изменяющий интерес игрока к настольной игре при комментирование,
2. ChangeByFollowingLink – метод, изменяющий интерес игрока к настольной игре при просмотре настольной игры,

UserService содержит 9 методов:

1. Get – метод, который возвращает объект пользователя по идентификатору,
2. Registration – метод, который регистрирует объект пользователя в системе,
3. Upload – метод, который загружает аватар пользователя на сервер,
4. GetCreatedMeet – метод, возвращающий список созданных мероприятий пользователя,
5. GetJoinedMeet – метод, возвращает список мероприятий в которые вступил пользователь,
6. GetCreatedBoardGame – метод, возвращает список созданных настольных игр ползователем,
7. GetAll – метод, который возвращает список всех пользователей в системе,
8. Delete – метод, который удаляет пользователя из системы,
9. FindByUserName – метод, который возвращает пользователя с определенным пользовательским именем.

AuthenticationService содержит метод Login, который авторизирует пользователя в системе и выдает токен авторизации.

**Слой контроллеров**

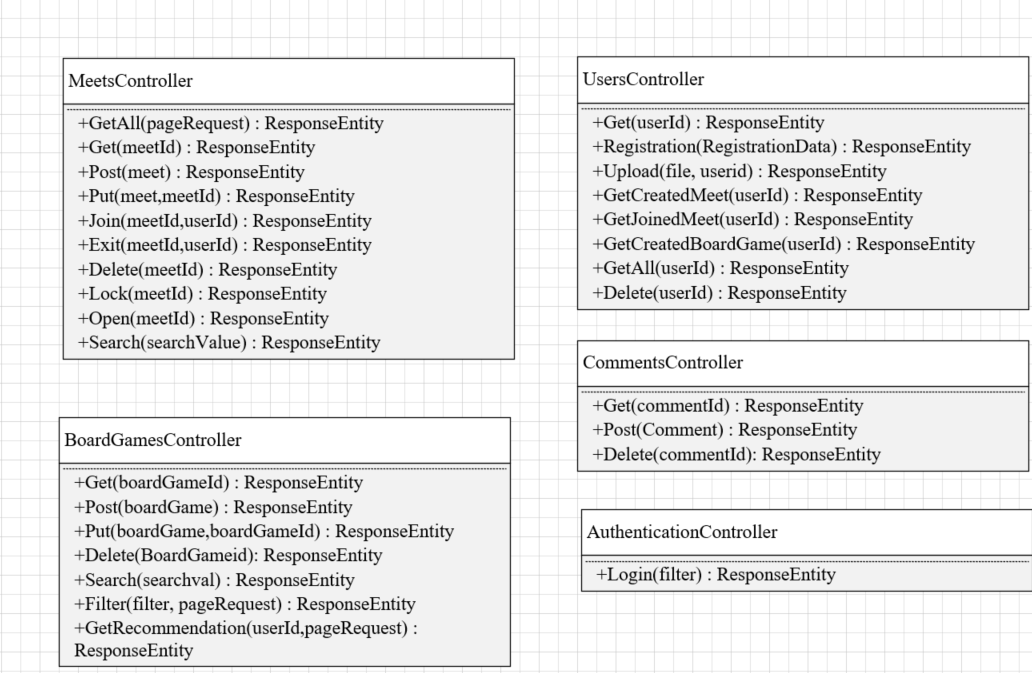


Рисунок 5 Слой контроллеров

В системе содержится 5 контроллеров:

1. BoardGameController,
2. CommentController,
3. MeetController,
4. UserController,
5. AuthenticationController,

все из них обрабатывают запросы на сервер и выдает объект ответа в HTTP-ответе

BoardGameController содержит 7 методов:

1. Get – метод, который возвращает HTTP-ответ с настольной игрой по идентификатору,
2. Post– метод который создает объект настольной игры и возвращает HTTP-ответ с созданной настольной игрой,
3. Put – метод который изменяет объект настольной игры и возвращает HTTP-ответ с измененной настольной игрой,
4. Delete – метод который удаляет настольную по идентификатору и выдает HTTP-ответ,
5. Search – метод который выдает HTTP-ответ со страницей настольных игр в именах, которых есть передаваемая подстрока,
6. Filter – метод который выдает HTTP-ответ со страницей настольных игр отфильтрованных и отсортированных по переданным правилам
7. GetRecommendation – метод выдает HTTP-ответ с страницей рекомендованных игр для определенного пользователя.

CommentService содержит 3 метода:

1. Get – метод, который выдает HTTP-ответ с комментарием по идентификатору,
2. Post – метод, который создает объект комментария и возвращает HTTP-ответ с созданным комментарием,
3. Delete – метод, который удаляет объект комментария по определенному идентификатору и выдает HTTP-ответ.

MeetService содержит 10 методов:

1. GetAll – метод, который выдает страницу с мероприятиями,
2. Get – метод, который выдает мероприятию по идентификатору,
3. Post– метод, который создает объект мероприятия и возвращает HTTP-ответ с созданным мероприятием,
4. Put– метод, который изменяет объект мероприятия и возвращает HTTP-ответ с измененным мероприятием,
5. Join – метод, который включает игрока в определенное мероприятие и возвращает HTTP-ответ,
6. Exit – метод, который исключает игрока из определенного мероприятия и возвращает HTTP-ответ,
7. Delete – метод, который удаляет мероприятие с определенным идентификатором и возвращает HTTP-ответ,
8. Lock – метод, который блокирует набор на мероприятие и возвращает HTTP-ответ c заблокированным мероприятием,
9. Open – метод, который открывает набор на мероприятие и HTTP-ответ c открытым мероприятием,
10. Search – метод, который выдает HTTP-ответ со страницей мероприятий содержащих в название переданную подстроку,

UserService содержит 9 методов:

1. Get – метод, который возвращает HTTP-ответ с объектом пользователя по идентификатору,
2. Registration – метод, который регистрирует объект пользователя в системе возвращает HTTP-ответ с зарегистрированным пользователем,
3. Upload – метод, который загружает аватар пользователя на сервер и возвращает HTTP-ответ с пользователем у которого изменен аватар,
4. GetCreatedMeet – метод, возвращающий HTTP-ответ со списком созданных мероприятий пользователя,
5. GetJoinedMeet – метод, возвращает HTTP-ответ со списком мероприятий в которые вступил пользователь,
6. GetCreatedBoardGame – метод, возвращает HTTP-ответ со списком созданных настольных игр пользователем,
7. GetAll – метод, который возвращает HTTP-ответ со списком всех пользователей в системе,
8. Delete – метод, который удаляет пользователя из системы и возвращает HTTP-ответ,

AuthenticationService содержит метод Login, который авторизирует пользователя в системе и выдает токен авторизации.