

### 3. В РАБОТА С БАЗОЙ ДАННЫХ

#### 3.1. Реализация архитектурного паттерна репозиторий

Реализация архитектурного паттерна "репозиторий" в фотоагентстве поможет упростить работу с базой данных и обеспечить более гибкую структуру приложения. Паттерн репозиторий позволяет абстрагировать работу с данными от остальной части приложения, что делает код более модульным, легко поддерживаемым и расширяемым.

В контексте фотоагентства репозитории могут быть использованы для работы с различными моделями данных, такими как фотосессии, фотографии, клиенты, оборудование и т.д. Каждый репозиторий будет предоставлять методы для выполнения операций CRUD (create, read, update, delete) с соответствующей моделью.

Примеры методов, которые могут быть реализованы в репозитории для модели PhotoSession:

1. `getAllPhotoSessions()`: получить список всех фотосессий из базы данных.
2. `getPhotoSessionById(id)`: получить конкретную фотосессию по ее идентификатору.
3. `createPhotoSession(photoSession)`: создать новую фотосессию в базе данных.
4. `updatePhotoSession(photoSession)`: обновить информацию о фотосессии в базе данных.
5. `deletePhotoSession(id)`: удалить фотосессию из базы данных.

При использовании паттерна репозиторий важно создать отдельный класс для каждой модели данных и инкапсулировать в них всю логику доступа к данным. Это позволит легко заменять и расширять методы работы с данными без изменения других частей приложения.

Кроме того, репозитории могут быть использованы для выполнения сложных запросов к базе данных, фильтрации данных и других операций, что

делает их очень полезными инструментами для работы с данными в фотоагентстве.

### **3.2. Использование EntityFramework для работы с базой данных**

При использовании Entity Framework в сочетании с паттерном репозиторий для работы с базой данных на сайте фотоагентства, вы можете создать контекст данных (DbContext), который будет представлять базу данных и таблицы в виде набора сущностей. В контексте данных определяются DbSet'ы для каждой сущности, таких как фотосессии, фотографии, клиенты, оборудование и другие.

Далее, создаются репозитории для каждой сущности, которые будут содержать методы для выполнения операций CRUD (Create, Read, Update, Delete). Репозитории будут использовать Entity Framework для доступа к данным через контекст данных.

Таким образом, паттерн репозиторий позволяет абстрагировать логику доступа к данным от остальной части приложения и упрощает тестирование и поддержку кода. Entity Framework же обеспечивает удобный способ работы с базой данных через объектно-ориентированный подход. Вместе они обеспечивают эффективное взаимодействие с базой данных в контексте фотоагентства, обеспечивая надежную и гибкую архитектуру данных для приложения.

### **3.3. Миграция данных**

При использовании Entity Framework для работы с базой данных на сайте фотоагентства также можете воспользоваться механизмом миграции данных. Миграции позволяют вносить изменения в структуру базы данных и обновлять её без потери данных.

Для использования миграций с Entity Framework, вам необходимо создать и применить миграции. Для этого вы можете воспользоваться инструментом командной строки, таким как Package Manager Console в Visual Studio.

Сначала необходимо создать миграцию, которая представляет собой скрипт изменений в базе данных. Для этого используется команда Add-Migration в Package

Manager Console. В миграции определяются изменения, такие как добавление новых таблиц, изменение структуры существующих таблиц и т.д.

Затем созданную миграцию можно применить к базе данных с помощью команды Update-Database. Эта команда применяет изменения, описанные в миграции, к базе данных.

Механизм миграций позволяет управлять изменениями в базе данных в процессе разработки и обновления приложения, обеспечивая согласованность структуры базы данных с моделью данных вашего фотоагентства.

#### *Заключение:*

*Использование Entity Framework с механизмом миграции данных представляет собой эффективный инструмент для управления базой данных на сайте фотоагентства. Этот подход обеспечивает удобное внесение и применение изменений в структуру базы данных, сохраняя целостность данных и обеспечивая согласованность с моделью данных приложения.*

*Применение паттерна репозиторий в сочетании с Entity Framework позволяет абстрагировать логику доступа к данным в фотоагентстве, делая код более модульным, легко поддерживаемым и расширяемым. Этот подход обеспечивает надежное взаимодействие с базой данных и повышает гибкость приложения.*

*Механизм миграций данных в Entity Framework предоставляет удобный инструмент для управления изменениями в базе данных фотоагентства. Он обеспечивает безопасное и последовательное внесение изменений в структуру базы данных, что помогает поддерживать её актуальность и соответствие модели данных приложения.*

Платон (427-347 гг. до н. э.) происходил из семьи афинской землевладельческой знати, получил хорошее образование, много путешествовал, затем создал свою философскую школу в Афинах, с которой были связаны последние 20 лет его жизни. Платона считают "отцом" политической науки. Его воззрения существенно менялись на протяжении его долгой жизни. Если в ранних его работах "Апология Сократа", "Протагор", "Критон" доминируют взгляды и метод Сократа - рационализм, то в более поздних - "Государство", "Политика", "Парменид", "Законы" прослеживаются религиозно-мифологические мотивы. Главным в учении Платона является его воззрение о двух мирах — мире идей и мире явлений. Мир идей — это истинное бытие, вечно неизменный божественный проект изменнического человеческого мира, мир явлений — это нечто мнимое, подвижное, искаженная копия мира идей.

Отрицательный тип государства выступает в четырех формах: как тимократия, олигархия, демократия и тирания. В сравнении с идеальным государством, каждая из этих форм есть последовательное ухудшение или извращение формы идеальной. В отрицательных формах государства вместо единомыслия налицо раздор, вместо справедливого распределения обязанностей — насилие и насильственное принуждение, вместо стремления правителей и воинов-стражей к высшим целям общежития - стремление к власти ради низких целей, вместо отречения от материальных интересов - алчность.

Первой во времени из этих отрицательных форм выступила, по Платону, тимократия, т. е. власть, основанная на господстве честолюбцев. В тимократии первоначально сохранились черты совершенного строя; здесь правители пользуются почетом, воины свободны от земледельческих и ремесленных работ и от всех забот материальных, трапезы общие, упражнения в воинском искусстве и гимнастика процветают. Однако со временем охотники до драгоценных металлов начинают в тайне собирать и хранить золото и серебро в стенах своих жилищ, и, при участии в этом жен, образ жизни меняется на роскошный. Так

начинается переход от тимократии к олигархии - господством немногих над большинством. Это правление, основывающееся на переписи и на оценке имущества, так, что в нем властвуют богатые, а бедные не имеют участия в правлении. В олигархическом государстве не выполняется основной закон жизни общества. По Платону, закон этот в том, чтобы каждый член общества "делал свое" и притом "только свое". Напротив, в олигархии, во-первых, часть членов общества занимаются каждый самыми различными делами - и земледелием, и ремеслами, и войной.

Во-вторых, в олигархии право человека, на полную распродажу накопленного им самим имущества приводит к тому, что такой человек превращается в совершенно бесполезного члена общества, он в нем лишь бедняк и беспомощный человек.

Дальнейшее развитие олигархии приводит, по Платону, к последовательному развитию ее в еще худшую форму государственного устройства, но правление в обществе, в котором противоположность между богатыми и бедными обостряется еще сильнее, чем при олигархии. Рост возмущения бедных против богатых приводит к восстанию. Если восстание заканчивается победой бедняков, то они часть богачей уничтожают, другую часть изгоняют, а государственную власть и функции управления разделяют между всеми оставшимися членами общества. Это и есть демократия.

Наихудшей формой отклонения от идеального государства Платон признал тиранию. Это власть одного над всеми в обществе. Возникает эта власть, подобно предыдущим формам, как выражение предшествующей ей демократической формы правления.

Тиран вырастает ни из чего более, как из корня, называемого представительством. В первые дни и в первое время он "улыбается и обнимает всех, с кем встречается, не называет себя тираном, обещает многое в частном и общем, освобождает от долгов, народу и близким к себе раздает земли и притворяется милостивым и кротким в отношении ко всем". Тирану необходимо непрерывно затевать войну, чтобы простой народ чествовал потребность в

вожде. Так как постоянная война возбуждает против тирана ненависть и так как граждане, способствовавшие его возвышению, будут мужественно осуждать оборот, который приняли события, то тиран, если захочет удержать власть, вынужден будет исподволь уничтожать своих осудителей, "пока не останется у него ни друзей, ни врагов, от которых можно было бы ожидать какой-нибудь пользы".