

- верситета им. Н.А. Некрасова. Серия: Педагогика. Психология. Социальная работа. Ювенология. Социокинетика. 2014. Т. 20. № 1. С. 28-32.
9. Скударёва Г.Н., Беликова, Е.А. Общественно-ориентированное образование как новая социально-педагогическая реальность // Современное общественно-ориентированное образование: диалог концепций. Сборник научных трудов по материалам Международной научно-практической конференции. 2015. С. 254-261.
 10. Скударёва Г.Н., Брагина Е.С. Информационно-коммуникативные технологии во взаимодействии учителя с родительским сообществом. Современное гуманитарное образование и иностранный язык // Сборник научных тезисов по материалам международной научно-практической конференции. 2016. С. 46.;
 11. Скударёва Г.Н., Павлова О.Г. Цифровое образование: от теоретического осмысления к реализации социального заказа // Современные здоровьесберегающие технологии. 2017. № 4. С. 150-157..
 12. Скударёва Г.Н., Осинина Т.Н. Социальный заказ на непрерывное педагогическое образование // Проблемы современного педагогического образования. Сер.: Педагогика и психология. Ялта: РИО ГПА, 2017. Вып. 57-1.Ч. 1. С. 200-209.
 13. Федеральный закон "Об образовании в Российской Федерации". М.: Проспект, 2013. - 160с.

РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ AR-ПРИЛОЖЕНИЯ ДЛЯ КОЛЛЕКТИВНОГО ОБУЧЕНИЯ

С.А. Ткачёв, В.С. Старшинов

(г. Томск, Национальный исследовательский Томский политехнический университет)

e-mail: vss21@tpu.ru

REALIZATION OF SERVER PART OF AR-APPLICATION FOR COLLECTIVE TRAINING

S.A.Tkachev, V.S.Starshinov

(Tomsk, National Research Tomsk Polytechnic University)

Abstract. The creation of virtual educational systems in the modern world is mandatory, and it is also possible to expand the reach of the audience and increase flexibility in the way educational services are provided. A necessary layer, an artificial information space between the subject and the object perceived by him, augmented reality. The idea of the application is to develop a toolkit that will allow for the collective training of group personnel, who must interact together, without having special programming skills for creating 3D models and their further visualization between all participants. In addition, the system may be included in the domain (in museums, libraries), when planning the architecture of buildings and in some cases. In this project, the authors implemented the server component of the application.

Keywords: AR, 3D, Unity, Server, ORM, ASP.NET Core

Введение. Создание виртуальных образовательных систем в современном мире является необходимостью, так как позволяет существенно расширить аудиторию и повысить гибкость в способах предоставления образовательных услуг. Необходима некоторая прослойка, искусственное информационное пространство между субъектом и воспринимаемым им объектом, дополненная реальность. Идея приложения состоит в разработке инструментария, который позволит обеспечить коллективное обучение некоторой группы людей, которые должны взаимодействовать вместе, при этом не обладая специальными навыками программирования для создания 3D моделей и их дальнейшей визуализации между всеми участниками. Также данная система может применяться в целях культурного досуга (в музеях, библиотеках), при планировании архитектуры зданий и в прочих случаях. В данном проекте авторами была реализована серверная составляющая приложения [1].

Общая схема архитектуры платформы. Приложение имеет трехзвенную архитектуру и состоит из клиентской части, веб-сервиса, конфигуратора проекта и хранилища файлов. Общая архитектура приложения представлена на рисунке 1 [2].

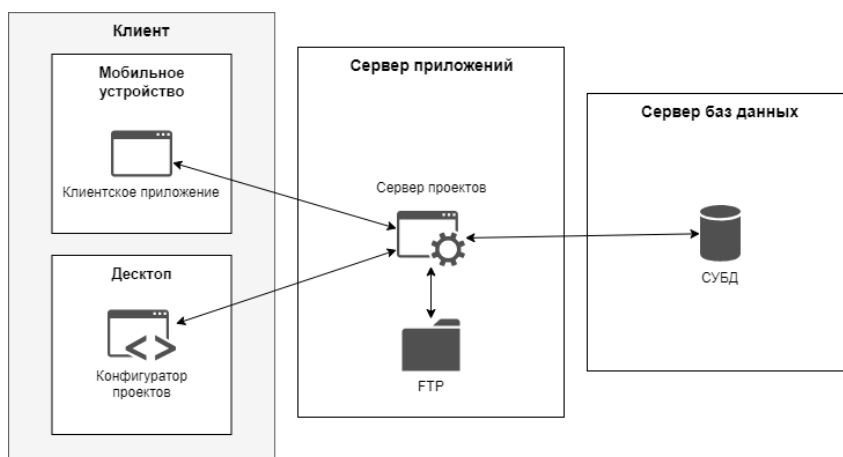


Рис. 1. Общая архитектура приложения

Сервер проектов – это веб-приложение, предназначенное для передачи данных между клиентом и сервером. Для разработки веб-приложения будут использоваться среда разработки Visual Studio и фреймворк ASP.NET Core.

Хранилище файлов предназначено для осуществления хранения файлов проектов на удаленном сервере. Доступ к файлам будет осуществляться по протоколу FTP.

Сервер баз данных так же представляет программную платформу, на которой размещена база данных, хранящая всю необходимую информацию. В качестве СУБД будет использоваться MS SQL Server [2].

Архитектура сервера проектов. Сервер проектов представляет собой веб-приложение Web API, применяющее стиль REST. Данный стиль подразумевает реализацию CRUD-операций.

На рисунке 2 представлена диаграмма пакетов, которая описывает архитектуру сервера проектов.

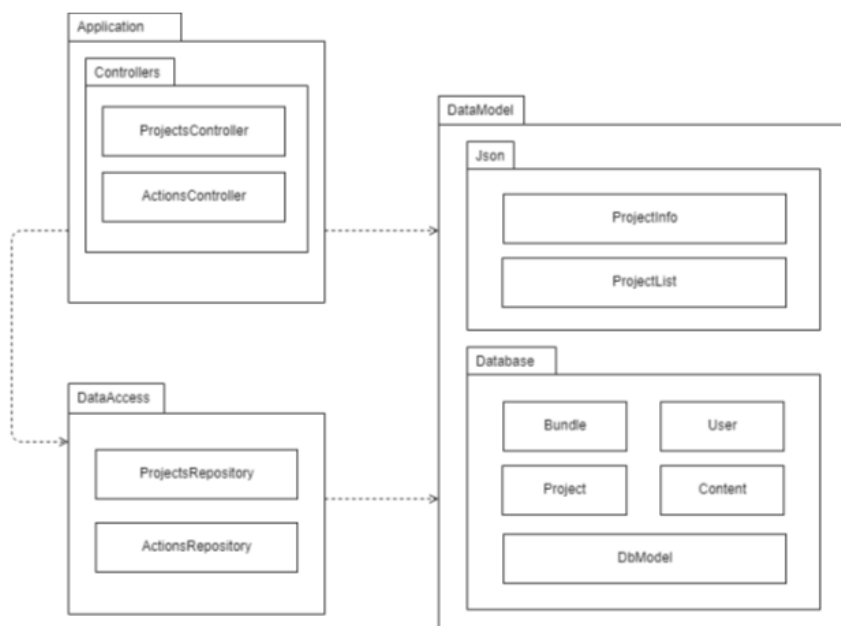


Рис. 2. Диаграмма пакетов приложения

На данной диаграмме представлены три пакета:

- Application – проект, содержащий контроллеры;
- DataModel – подключаемая библиотека, содержащая классы сущностей;
- DataAccess – подключаемая библиотека, содержащая классы хранилищ данных.

Пакет Application включает в себя пакет:

- Controllers – пакет, содержащий классы контроллеров.

Взаимодействие сервера проектов с базой данных. Для того, чтобы осуществить взаимодействие между сервером проектов и базой данных без необходимости написания SQL-кода, нужно использовать технологию ORM. С помощью данной технологии осуществляется работа с данными как с классами, а не как с таблицами. Кроме этого, данная технология позволяет преобразовывать данные классов в данные, хранящиеся в таблицах.

В качестве решения данной задачи была выбрана технология ADO.NET Entity Framework Core (рис. 3). Она предоставляет возможность взаимодействия с объектами посредством LINQ to Entities. Существует несколько подходов к решению, но выбор был сделан в пользу подхода Database First. С помощью данного подхода по готовой базе данных можно сгенерировать модель Entity Data Model, которая будет играть роль «виртуальной базы данных». Кроме генерации модели Entity Data Model так же генерируются классы, соответствующие таблицам в базе данных [3].

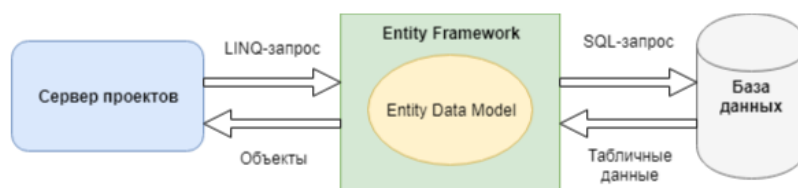


Рис. 3. Схема взаимодействия сервера проектов и базы данных через Entity Framework

Взаимодействие клиента с сервером проектов. Для того, чтобы осуществить взаимодействие между клиентом и сервером проектов, нужно использовать HTTP-методы:

- PUT – отвечает за редактирование данных;
- GET – отвечает за получение данных;
- POST – отвечает за отправку данных;
- DELETE – отвечает за удаление данных.

С помощью данных запросов можно отправлять и получать данные, необходимые для работы клиента. Данные, которые используются для обмена информацией, представлены в формате JSON (рис. 4).



Рис. 4. Пример структуры POST-запроса

Реализация архитектуры сервера проектов. После того, как база данных была размещена на сервере, следующим этапом была реализация архитектуры сервера проектов, которая представлена на рисунке 5 [4].

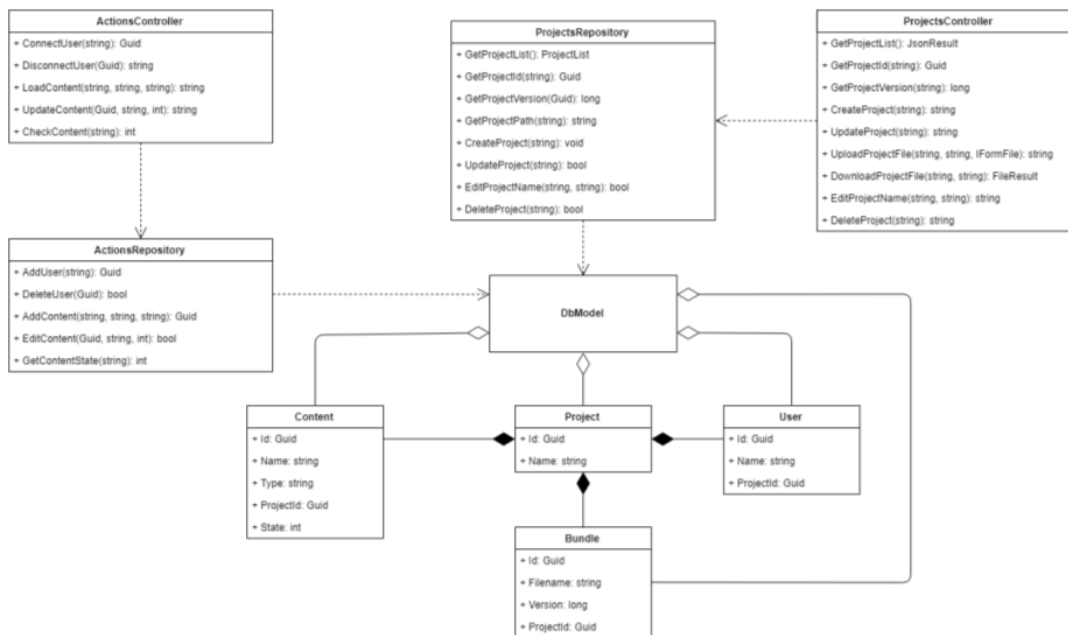


Рис. 5. Архитектура сервера проектов

На данной диаграмме классов представлены следующие классы:

- * **ProjectsController** – контроллер, предоставляющий методы для работы с проектами:
 - **GetProjectList()** – метод, отвечающий за получение всех проектов с сервера.
 - **GetProjectId(string)** – метод, отвечающий за получение идентификатора проекта с сервера.
 - **GetProjectVersion(Guid)** – метод, отвечающий за получение версии проекта с сервера.
 - **CreateProject(string)** – метод, отвечающий за создание проекта на сервере.
 - **UpdateProject(string)** – метод, отвечающий за обновление проекта на сервере.
 - **UploadProjectFile(string, string, IFormFile)** – метод, отвечающий за отправку файла проекта на сервер.
 - **DownloadProjectFile(string, string)** – метод, отвечающий за получение файла проекта с сервера.
 - **EditProjectName(string, string)** – метод, отвечающий за изменение названия проекта на сервере.
 - **DeleteProject(string)** – метод, отвечающий за удаление проекта с сервера.
- * **ActionsController** – контроллер, предоставляющий методы для многопользовательского взаимодействия.
 - **ConnectUser(string)** – метод, отвечающий за подключение пользователей к проекту.
 - **DisconnectUser(Guid)** – метод, отвечающий за отключение пользователей от проекта.
 - **LoadContent(string, string, string)** – метод, отвечающий за загрузку контента проекта.
 - **UpdateContent(Guid, string, int)** – метод, отвечающий за обновление состояния контента проекта.

- CheckContent(string) – метод, отвечающий за проверку состояния контента проекта.
- * DbModel – класс, отвечающий за хранение объектов классов Bundle, Project, Content и User.
- * Bundle – класс, отвечающий за хранение информации о файле проекта.
 - Id – свойство, отвечающее за хранение идентификатора файла проекта.
 - Filename – свойство, отвечающее за хранение пути к файлу проекта на сервере.
 - Version – свойство, отвечающее за хранение версии файла проекта.
 - ProjectId – свойство, отвечающее за хранение идентификатора проекта, к которому относится файл.
- * Project – класс, отвечающий за хранение информации о проекте.
 - Id – свойство, отвечающее за хранение идентификатора проекта.
 - Name – свойство, отвечающее за хранение названия проекта.
- * Content – класс, отвечающий за хранение информации о контенте проекта.
 - Id – свойство, отвечающее за хранение идентификатора контента проекта.
 - Name – свойство, отвечающее за хранение названия контента проекта.
 - Type – свойство, отвечающее за хранение типа контента проекта.
 - ProjectId – свойство, отвечающее за хранение идентификатора проекта, к которому относится контент.
 - State – свойство, отвечающее за хранение состояния контента проекта.
- * User – класс, отвечающий за хранение информации о пользователях проекта.
 - Id – свойство, отвечающее за хранение идентификатора пользователя проекта.
 - Name – свойство, отвечающее за хранение имени пользователя проекта.
 - ProjectId – свойство, отвечающее за хранение идентификатора проекта, к которому подключается пользователь.

Физическая схема базы данных представлена на рисунке 6.

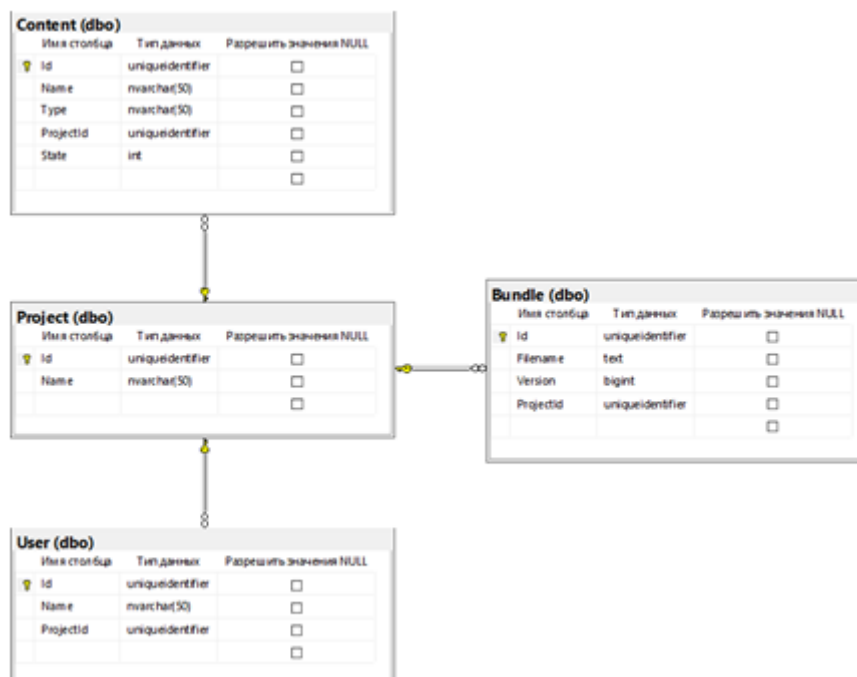


Рис. 6. Физическая схема базы данных

Физическая схема базы данных состоит из 4 таблиц:

- Таблица Content содержит информацию о контенте и хранит состояние анимации.
- Таблица Bundle содержит название упакованного контента, привязанного к проекту.
- Таблица Project содержит имя проекта.
- Таблица User содержит информацию о пользователях, которые подключаются к проекту.

Реализация логики данной таблицы предусмотрена следующим функционалом: таблицы Project и Bundle обеспечивают хранение упакованного контента AssetBundle, а точнее ссылок на данные файлы, которые располагаются на FTP-сервере. Многопользовательское взаимодействие осуществляют таблицы Content, User и Project, которые привязывают пользователя к конкретному проекту, а также контент к конкретному проекту.

Заключение. В процессе выполнения работы было проведено проектирование и реализация серверной части платформы для коллегиального обучения, то есть были спроектирована архитектура серверной части, на основании которой была реализована физическая база данных, хранилище и веб-сервис. Веб-сервис был развернут с применением Microsoft Azure.

ЛИТЕРАТУРА

1. Дополненная реальность — что сегодня предлагают разработчики? // Хабр [Электронный ресурс]. – URL: <https://habr.com/company/madrobots/blog/407723/> (Дата обращения 4.09.2018).
2. AR-жизнь: применение и перспективы дополненной реальности // DTF [Электронный ресурс]. – URL: <https://dtf.ru/gamedev/7800-ar-zhizn-primenenie-i-perspektivy-dopolnennoy-realnosti>. (Дата обращения 4.09.2018).
3. ORM или как забыть о проектировании БД // Хабр [Электронный ресурс]. – URL: <https://habr.com/post/237889/>. (Дата обращения 6.09.2018).
4. Vuforia: немного магии в нашей реальности // Хабр [Электронный ресурс]. – URL: <https://habr.com/post/198862/>. (Дата обращения 6.09.2018).

РАЗРАБОТКА КОМПЛЕКСА МЕРОПРИЯТИЙ И ПРОГРАММ ДЛЯ СОЗДАНИЯ НОВОЙ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

А. В. Хаперская

(г. Томск, Томский политехнический университет)

e-mail: khape@mail.ru

THE DEVELOPMENT OF ACTIVITIES AND PROGRAMS FOR THE CREATION OF A NEW EDUCATIONAL ENVIRONMENT ON THE BASIS OF ARTIFICIAL INTELLIGENCE

V. Khaperskaya

(Tomsk, Tomsk Polytechnic University)

Abstract. The most popular type of information and educational environment are online courses that contain cloud structures. This does not take into account the specifics of working with big data – the possibility of parallelization and duplication of information flows, especially when using neural network algorithms for data analysis in the learning process. This makes it impossible to deploy structures, and does not exclude