

*Дворовкин Андрей Григорьевич
студент магистратуры,
Физико-технологический институт
Калужский государственный университет им. К.Э. Циолковского,
Россия, г. Калуга
e-mail: andrey-dvorovkin@mail.ru*

*Научный руководитель: Тимошина Н.В.,
кандидат педагогических наук, доцент
Калужский государственный университет им. К.Э. Циолковского,
Россия, г. Калуга*

ИСПОЛЬЗУЕМЫЕ СРЕДСТВА ДЛЯ РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСПИСАНИЕ, ДЛЯ УЧЕБНЫХ УЧРЕЖДЕНИЙ

***Аннотация:** В статье рассмотрены сведения для разработки автоматизированной информационной системы расписание, для учебно-образовательных учреждений, в частности определены инструменты, а также приведены их понятия.*

***Ключевые слова:** автоматизированная информационная система, образовательные учреждения, расписание занятий.*

*Dvorovkin Andrey Grigorievich
Master student,
Institute of physics and technology
Kaluga state University. K. E. Tsiolkovsky,
Russia, Kaluga*

*Scientific adviser: Timoshina N. V.,
candidate of pedagogical sciences, associate professor
Kaluga state University. K. E. Tsiolkovsky,
Russia, Kaluga*

THE TOOLS USED FOR DEVELOPING AN AUTOMATED INFORMATION SYSTEM SCHEDULE FOR EDUCATIONAL INSTITUTIONS

***Abstract:** The article deals with information for the development of an automated information system schedule, for educational institutions, in particular, the tools are defined, as well as their concepts.*

***Key words:** automated information system, educational institutions, class schedules.*

На первый взгляд, автоматизированная информационная система достаточно проста в своей функциональной части, однако если углубиться и сделать действительно хороший сервис, необходимо разработать следующие функциональные модули:

- Веб-интерфейс для пользователей системы
- Веб-интерфейс для администраторов системы с разграничением прав доступа;
- Интеграция с социальными сетями;
- Файлообменник.

Веб-интерфейс системы разрабатывался на языке программирования PHP – это мощный, серверный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML.

В настоящее время разработка практически любого веб-приложения не обходится без использования сторонних библиотек и фреймворков.

В качестве серверного языка программирования для разработки системы был выбран язык PHP и фреймворк Laravel.

Фреймворк – это набор различных библиотек (инструментов) для быстрой реализации повседневных (рутинных) задач. Главная цель фреймворка – предоставить программисту удобную среду (экосистему) для проекта с большим и хорошо расширяемым функционалом.

Важно понимать, что фреймворк не является средой разработки или системой управления веб-сайтом (CMS). Фреймворк – это набор готовых библиотек. Самое важное в фреймворке – это предоставление изначально правильной и расширяемой структуры приложения.

Laravel предоставляет следующие возможности для разработки:

1. Паттерн проектирования MVC

Паттерн Model-View-Controller (модель – представление – контроллер) является важнейшей частью веб-приложений. Такая архитектура позволяет разделить приложение на три отдельных компонента таким образом, чтобы

изменение одного из этих компонентов оказывало минимальное влияние на работу двух остальных [1, с. 73].

В такой архитектуре каждый компонент выполняет свои функции.

Модель (Model) предоставляет данные, работает с базой данных, реагирует на запросы, изменяя свое состояние, но не содержит никакой информации о том, каким образом данные визуализируются. Другими словами, вся бизнес-логика приложения описана в моделях.

Представление отвечает за отображение информации (визуализацию).

Контроллер обеспечивает связь между пользователем и системой. Он получает запрос от пользователя, контролирует ввод данных, получает данные из необходимой ему модели, преобразует данные в необходимую форму и передает их представлению, которое в свою очередь выполняет свои функции (отображает данные).

Работу такой архитектуры можно представить следующим образом:

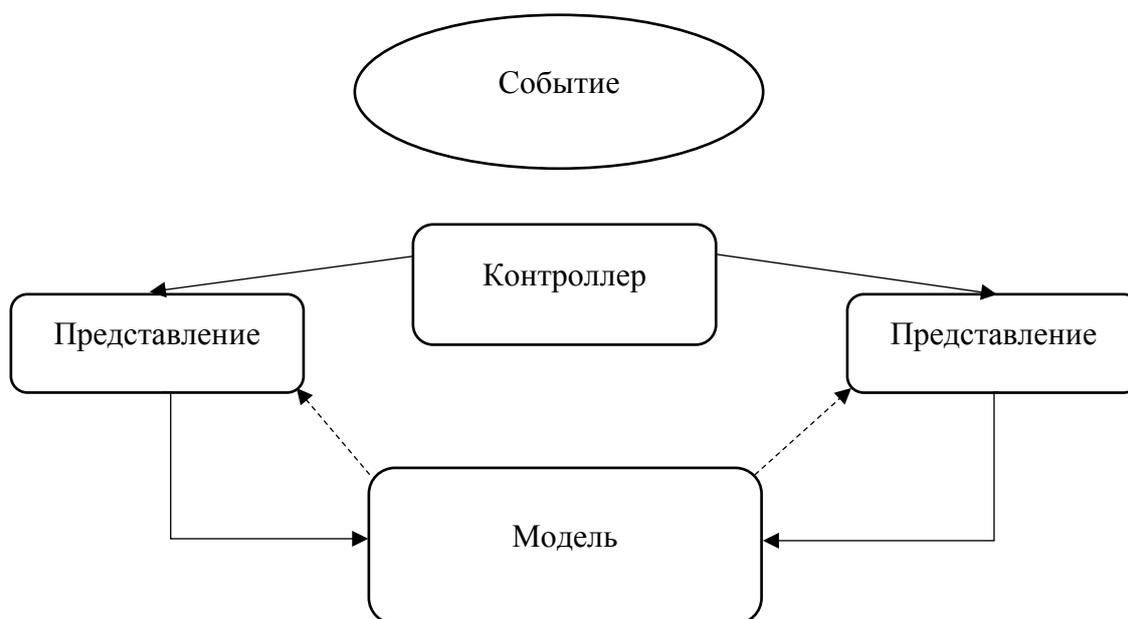


Рисунок 1. Архитектура

Такой подход к проектированию архитектуры приложения позволяет гибко настраивать модели независимо от их визуализации и наоборот.

2. Собственная система связей и Eloquent ORM

Как известно, в любой базе данных между сущностями можно создавать связи. При разработке на Laravel связи также, конечно же, существуют, но несколько в другом виде.

Вся забота о нормализации данных в БД лежит на самом приложении, а не на системе управления базой данных (СУБД). То есть, связи описываются как часть модели, а в самой базе данных (MySQL или др.) никаких связей не создается.

Наряду со стандартными связями существующих СУБД, таких как «многие ко многим», «один ко многим» и «один к одному» в Laravel существует особый тип связи, который особо востребован в веб-приложениях: полиморфная связь. Такая связь позволяет модели быть связанной с более, чем одной моделью.

Например, в разрабатываемой CRM системе есть такая модель, как «тэги». Тэги могут существовать почти у любой другой сущности системы: сделки, компании, персоны, проекты и т.д. Более того, у одной сущности может быть несколько тэгов. Если бы связь между сущностями организовывалась бы как «многие ко многим», то для каждого отношения пришлось бы создавать отдельную таблицу, что в итоге породило бы 10-15 таблиц с одинаковой структурой. Полиморфная связь позволяет хранить такие данные в одной таблице и обеспечивает к ним доступ через стандартное описание отношений [2, с. 137].

ORM (Object-relational mapping) – это технология программирования, которая позволяет связывать базы данных с концепциями объектно-ориентированного программирования (ООП), «создавая виртуальную объектную базу данных». Другими словами, ORM позволяет переводить данные из БД в объекты языка программирования. При использовании ORM отпадает необходимость написания запросов непосредственно на языке SQL (за

исключением сложных запросов). Все запросы формируются с помощью «обёрток» над языком SQL, а данные возвращаются сразу в объектах класса.

3. Система миграций

Миграции – это один из видов контроля версий базы данных приложения. С помощью миграций можно выполнять любые операции над базой данных: добавление и удаление таблиц, столбцов, манипуляции с данными (удаление/копирование/добавление), создание индексов и многое другое. Другими словами, миграции – это инструмент, позволяющий управлять структурой базы данных с помощью специальных команд фреймворка. Плюсы такого подхода к работе с базой данных заключаются в том, что каждая миграция создает «новую версию базы данных», к которой можно вернуться в любой момент. Также, миграции позволяют быстро изменить структуру БД на сервере приложения. Обычно, разработка приложения осуществляется на локальной машине, а необходимые изменения отправляются на основной сервер. Если изменения касаются только файлов, то проблем не возникает. С базами данных дела обстоят несколько иначе: при изменении структуры БД необходимо вручную изменять структуру и на основном сервере. Миграции решают эту проблему: все, что необходимо сделать – это запустить миграции на основном сервере – они переведут базу данных в новую версию, при этом данные, хранящиеся в БД, останутся без изменений.

4. Мягкое удаление

Мягкое удаление (или псевдоудаление) позволяет не удалять данные из БД, а лишь пометить их удаленными. Это позволяет защититься от случайного или умышленного удаления данных. Реализовано это достаточно просто: в каждую таблицу, для которой необходимо реализовать «мягкое удаление», добавляется поле с названием «created_at», которое хранит дату удаления, если запись удалена, либо значение NULL, если запись не удалена. При этом, при выборке записей, Eloquent ORM автоматически добавляет условие на выборку только неудалённых записей, если это необходимо.

5. Локализация

Laravel дает возможность создавать несколько языков интерфейса приложения. Все что необходимо для реализации мультиязычности – это описание всех строковых значений, отображаемых в интерфейсе приложения в соответствующих конфигурационных файлах.

6. Валидация

Laravel предоставляет удобный интерфейс проверки данных, полученных от пользователя. Для этого для каждой модели определяются правила, которым эта модель должна соответствовать. Перед сохранением объекта модели в базу данных сначала будет произведена проверка того, соответствуют ли данные модели ее правилам. Laravel содержит большое количество стандартных правил валидации: проверка обязательных полей, проверка длины введенного текста и т. д. Вдобавок к существующим правилам Laravel предоставляет возможность описывать собственные правила. Кроме всего прочего, валидатор Laravel производит фильтрацию полученных данных, обеспечивая высокую безопасность приложения.

7. Поддержка composer

Composer – это менеджер зависимостей для проектов, разработанных на PHP. Он позволяет производить установку необходимых пакетов (библиотек), при этом автоматически добавляет их в автозагрузчик классов. Также, composer позволяет управлять версиями пакетов и обновлять их при необходимости. В официальном репозитории composer зарегистрировано почти 100 тысяч различных пакетов, все они бесплатны для использования.

8. Кэширование данных

В Laravel реализована работы с самыми популярными сервисами кэширования данных, такими как Memcached и Redis. Кэширование необходимо для того, чтобы сохранять промежуточные данные с целью уменьшения нагрузки на БД. Кэширование необходимо в случаях, когда к базе данных делаются сложные запросы: в таком случае запрос к базе данных выполняется один раз, а данные сохраняются в кэш и используются до момента потери их

актуальности, после чего снова обновляются из базы данных и используются снова.

Вышеприведенные возможности фреймворка Laravel дают нам возможность сосредоточиться на бизнес-логике приложения, оставив решение рутинных задач самому фреймворку.

PHP в связке с фреймворком выступает серверным языком программирования веб-оболочки системы. На клиентской части используется язык разметки HTML – стандартизированный язык разметки документов во Всемирной паутине. Также, для придания веб-оболочке интерактивности, используется язык программирования JavaScript (JS). За визуальное оформление интерфейса отвечают таблицы каскадных стилей – CSS

В качестве веб-сервера приложения будет использоваться портативная серверная платформа под названием «OpenServer». Этот программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов.

И, конечно, нельзя не сказать об очень важном компоненте портала – базе данных. В веб-разработке очень популярной СУБД является MySQL ввиду тесной её интеграции с PHP и поддержкой различными библиотеками для работы с базами данных, таких как PDO.

Для управления процессом разработки и контролем версий приложения была использована система контроля версий под названием Git. Система контроля версий позволяет сохранять промежуточное состояние кода разрабатываемого приложения. Это особенно актуально в больших системах, где важно в любой момент времени иметь возможность вернуться на определенной момент разработки или зафиксировать текущие изменения.

Автоматизированная информационная система расписаний является сложным механизмом, что подтверждается использованием большого количества самых различных технологий в её разработке. Каждый из компонентов системы разрабатывается с использованием различных средств,

языков программирования, фреймворков, а в связке все эти звенья решают задачу удобного предоставления расписания занятий для студентов.

Список литературы:

1. Эрик Фримен, Элизабет Фримен, Кэти Сьерра, Берт Бейтс. Паттерны проектирования. 2017. С. 73-74.
2. Илюшечкин В. М. Основы использования и проектирования баз данных. Учебник. М.: Юрайт. 2015. 137 с.