**2** СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Изучив теоретические аспекты разрабатываемой системы и выработав список требований необходимых для разработки системы, разбиваем систему на функциональные блоки Это необходимо для обеспечения гибкой архитектуры. Такой подход позволяет изменять или заменять модули без изменения всей системы в целом.

В разрабатываемом веб-приложении можно выделить следующие блоки:

* блок маршрутизации;
* блок авторизации;
* блок управления;
* блок валидации данных;
* блок модели данных;
* блок построения запроса;
* блок базы данных;
* блок загрузки первичных данных;
* блок контроля версий;
* блок формирования ответа.

Структурная схема, иллюстрирующая перечисленные блоки и связи между ними приведена на чертеже ГУИР.400201.601 C1.

Каждый модуль выполняет свою задачу. Чтобы система работала каждый модуль взаимодействует с другими модулями путем обмена данными, используя различные форматы и протоколы.

Рассмотрим функциональные блоки веб-приложения.

*Блок маршрутизации* представляет собой механизм для определения пути прохождения запроса от клиентской части приложения, которое выполняет запрос к данному API. Также данный блок позволяет структурировать группы запросов, чтобы определить какие из запросов нуждаются в авторизации, а какие не являются приватными. К данным групп запросов также возможно применение промежуточных слоев, которые внедряют дополнительную обработку частей запроса.

*Блок авторизации* используется теми маршрутами, которые включены в группы приватных маршрутов. Данный блок необходим для проверки прав на выполнения конкретного запроса. При наличии у пользователя достаточных прав на выполнения определенного запроса, данный блок позволит запросу выполниться, в противном случае будет вызвано исключение или переадресация на другой, доступный пользователю маршрут.

*Блок управления* является главным блоком всего проекта, который используется для взаимодействия остальных блоков. Данный блок, при поступлении запроса с наличием каких-либо данных, отправляет их на проверку в *Блок валидации*. При необходимости работы с базой данных, *Блок управления* взаимодействует с ней через *Блок модели данных.* При сборе всех необходимых данных, соответствующий запросу клиентку, данный блок посылает их в *Блок формирования ответа* для дальнейшей обработки.

*Блок валидации данных* предоставляет возможность оценить корректность и валидность данных, которые пришли от клиента, используя различные методы проверки. Для этого при получении данных для последующей валидации также принимаются правила для их валидации. В результате работы данного блока, возвращается статус валидации

*Блок модели данных* предоставляет удобный интерфейс работы с объектами данных. С его помощью данные, хранящиеся в базе данных структурируются в необходимые конструкции для дальнейшей обработки. Также блок позволяет преобразовывать типы данных, объединять данные из разных таблиц и производить фильтрацию. Перед сохранением данных в базу данных, данный блок производит дополнительные преобразования и отсылает человеко понятные запросы в *Блок построения запроса.*

*Блок построения запроса* является трансформатором человеко понятных запросов в запросы, необходимые интерфейсу базы данных. При этом происходит объединение нескольких простых запросов в один сложный, для обеспечения наибольшей производительности и наименьшему времени отклика.

*Блок базы данных* является непосредственным инструментов для взаимодействия с базой данных. В данном проекте используется реляционная база данных MySQL, а также Redis, для хранения необходимых для модуля поиска данных. Данный блок непосредственно выполняет, пришедшие из *Блока построения запроса* запросов и посылает ответ в *Блок модели данных* для дальнейших преобразований и обработки.

*Блок загрузки первичный данных* необходим для инициализации базы данных приложения первичными данными, которые будут необходимы для формирования базы статический данных.

*Блок контроля версий* используется для версионирования изменений, которые проводятся с таблицами и данными в базе данных. Он необходит для поддержания всех копий данного продукта в оптимальном состоянии.

*Блок формирования ответа* является точкой выхода работы приложения. После успешной работы приложения *Блок управления* передает данные данному блоку для дальнейшего преобразования в формат и структуру, которую запрашивал пользователь.

Данное приложение реализовано на PHP фреймворке Laravel, основой которого является шаблон проектирования MVC, контроллером из данного шаблона является *Блок управления*, представлением является *Блок формирования ответа*, а моделью – *Блок модели данных*.