

## 15. Техническое решение

Опираясь на пункты **FR005, NFR01, NFR04, NFR05, NFR19, NFR20** в приложении 2 и 3 для реализации всех четырех программных продуктов мы используем следующий стек технологий:

- **LAMP;**
- **HTML, JavaScript, CSS;**
- **Bootstrap;**
- **Laravel Framework;**
- **SASS;**
- **Webpack;**
- **БЭМ;**
- **Vue JS;**
- **Git.**

### Используемые технологии и подходы

#### Бэкенд

- Руководствуемся принципами разработки (**KISS, DRY, SOLID**).
- Язык PHP с применением паттернов проектирования.
- **Laravel Framework.**
- Реляционная база данных **MySQL**:
  - репликации;
  - применение индексов;
  - транзакции с указанием уровня изоляции, блокировки;
  - построение сложных запросов.
  -
- Система контроля версий **Git** с моделью ветвления **master, stable, develop, task.**
- Организация релевантного морфологического поиска:
  - **Elasticsearch.**
  - «Родная» поддержка полнотекстового поиска для **MySQL**, с частичным использованием алгоритма «**Стеммер Портера**».
- Другие технологии:
  - менеджер пакетов и зависимостей **Composer;**
  - автоматизация тестирования основана на **Codeception;**
  - CI Server — **Gitlab;**
  - анализаторы кода **phpcpd, phpmd**, а также **phploc** для фиксирования метрик;
  - применяется проверка стандартов написания кода с помощью **phpcs;**
  - **Redis, Memcached** — кеширование и сервер очередей;
  - **Docker** — разработка, развертывание на серверах.



## Фронтенд

- Верстаем макеты с использованием технологий **HTML, CSS, JavaScript**:
  - принципы БЭМ и разделение интерфейса на абсолютно независимые компоненты;
  - css анимации и js интерактив;
  - адаптивность, в том числе за счет использования библиотеки **Bootstrap**;
  - кроссбраузерность **IE8+**;
  - семантика и валидность **html** разметки.
- Настраиваем сборку **frontend**:
  - компиляция препроцессоров **SASS**;
  - настройка гибкого, модульного JavaScript с **Webpack**, с возможностью писать на новом стандарте ES6, транслируя код в ES5 с помощью Babel;
  - автоматическая проверка стиля кода инструментами **Stylelint** и **ESLint**.
- Оптимизируем производительность проектов:
  - сокращение количества запросов к серверу, путем правильной сборки зависимостей;
  - постепенная загрузка не критичных компонентов страницы;
  - управление отображением содержимого в процессе загрузки страницы;
  - правильное использование различных форматов графики и ее оптимизация.
- Используем библиотеку **Vue JS** для реализации интерфейсов «на лету».

## Проектирование

- Построение схем бизнес-процессов **BPMN, EPC, IDEF0**.

## Коммуникации внутри команды

- **Slack, Trello**.
- Сервисы **Google**.

## Среда разработки

- **PHPStorm**.

