

Техническое задание на разработку портала

27/07/2016

1 Описание проекта

Рабочее название проекта - WorkWell. Проект представляет собой платформу, состоящую из нескольких сервисов. Главная цель проекта - позволять пользователям бронировать места в коворкингах, которые принадлежат владельцу.

1.1 Цели платформы для клиента

Клиент (пользователь платформы) может регистрироваться в системе (различные абонементы). Он может переписываться с другими участниками (чат), он может принимать участие в специальных мероприятиях, создавать и откликаться на объявления, получать бонусы и бронировать помещения/столы.

1.2 Цели сайта для заказчика

Проект дает заказчику возможность управлять своим контентом. Он может создавать новые здания с интерактивными планами, управлять данными планами, получать статистику загрузки помещений.

2 Технические уточнения

2.1 Термины

- Блок - некий визуальный элемент, выделяющийся либо графически (в виде рамок, очертаний), либо по смыслу (совокупность похожих элементов)
- Компонент - часть содержания, имеющего закрытое визуальное представление. Одна страница сайта может состоять из нескольких компонентов.
- Фронтэнд - для пользователя видимая оболочка веб-страницы
- Бэкэнд - невидимые для пользователя математические алгоритмы
- CMS - все работы происходят на основе системы управления содержанием - CMS 1C Bitrix (1С Битрикс)

- Модуль - является описанием общего функционала, который не может быть классифицирован как привязанный к определенной странице. Он может встречаться на любой странице в любом месте. Модуль может состоять из нескольких компонентов. Также модуль может содержать в себе части логики фронтэнда и бэкэнда.
- Хэдер - верхняя часть сайта, обладающая определенной структурой, которая видна сквозняком на всех или почти всех страницах сайта. Также используется обозначение "шапка".
- Футер - нижняя часть сайта. Функционал аналогичен хэдеру. Также используется обозначение "подвал".
- Администраторская панель ("админка") - часть сайта, скрытая для обычных пользователей, позволяющая редактировать содержимое сайта
- Публичная часть ("паблик") - часть сайта, доступная для просмотра обычным пользователям.
- Слайдер - блок с фотографиями и (опционально) текстом, которые можно листать (влево/вправо). Обычно размещается в верхней части страницы и занимает всю ширину экрана. Также возможны более узкие слайдеры.

2.2 Технические требования к сайту

Сайт располагается на хостинге или выделенном сервере (virtual dedicated server), в зависимости от предпочтений заказчика.

3 Архитектура проекта

3.1 Подход API first

Для данного проекта используется подход "API first". Это означает, что разрабатывается RESTful API, которая позволяет получить доступ к данным вне зависимости от приложения. Таким образом мы разграничиваем обработку данных и вывод данных клиенту.

3.2 Структура

Проект имеет следующую структуру:

- Основное приложение
- API-надстройка
- Фронтэнд-приложение (может быть одно или несколько)
- Мобильное приложение

Имеется основная установка платформы, которая фигурирует как администрируемая база данных (на основе Битрикс: Управление Сайтом). Для создания API используется дополнительный фреймворк (так как методами Битрикса создать API невозможно, а писать чистый PHP очень долго) - это может быть ZEND, Yii 2 или Symfony (на обоснованный выбор разработчика).

3.3 Модули проекта

Модули проекта - это сайт или приложение с определенным набором функционала и как правило, обладающий собственным названием. Перечень модулей:

- Личный кабинет пользователя
- Мобильное приложение
- Лендинг
- Оплата

3.4 Компоненты модулей

3.4.1 Личный кабинет

Модуль состоит из следующих компонентов (могут пересекаться с понятием "страница но не обязательно).

- Дашборд
- Акции
- Друзья
- Сообщества
- Места (брони)
- Мероприятия

3.5 Общее

Проект изначально должен соблюдать правила объектно ориентированного программирования. Т.е. любой пользователь должен быть объектом класса "пользователь". Это распространяется на все сущности. Все действия системы должны быть методами своих классов. Псевдокод:

```
#Class description
```

```
class User < ActiveRecord::Base

def initialize(name, phone, mail)
  @name = name
  @phone = phone
  @mail = mail
end
```

```
# Register method
```

```
def register(user_params)
  User.create(user_params)
```

```
end
```

```
# Allow only whitelisted params
```

```
private
```

```
  def user_params
```

```
    params.require(:user).permit(:name, :phone, :mail)
```

```
  end
```

Программирование должно осуществляться по подходу TDD - test driven development. Это означает, что перед написанием функционала необходимо написать тест (который заведомо провалится, так как функционала нет). Далее пишется минимальный функционал, который позволяет пройти данный тест.

Псевдокод теста:

```
require 'user'
```

```
Rspec.describe User, "#register" do
```

```
  context "when user is completely new" do
```

```
    it "creates a new entry with valid data" do
```

```
      user = User.new
```

```
      user.name = "Vasya Pupkin"
```

```
      user.phone = "89032278874"
```

```
      user.mail = "vasya@pupkin.ru"
```

```
      user.register
```

```
      expect(User.count).to change{User.count}.by(1)
```

```
    end
```

```
    it "does not create a new entry with invalid data" do
```

```
      user = User.new
```

```
      user.name = "Vasya Pupkin"
```

```
      user.phone = "j1lkh3i1u2y98"
```

```
      user.mail = "someinvalidtext"
```

```
      user.register
```

```
      expect(User.count).not_to change{User.count}.by(1)
```

```
    end
```

```
  end
```

```
end
```

Т.е. в данном случае мы тестируем создание нового пользователя с валидными данными (первый случай) и с невалидными данными. В первом случае мы ожидаем увеличение общего кол-ва пользователей на 1, во втором случае мы не ожидаем увеличений.

3.6 Модели

Перечень:

- Пользователь
- Стейтмент

- Комментарий
- Страна
- Город
- Здание
- Этаж
- Отсек
- Акция
- Мероприятие

3.6.1 Структура моделей

3.6.1.1 Пользователь Пользователь имеет связку has-and-belongs-to-many :through с другими пользователями. Это означает, что есть дополнительная таблица, которая ведет учет пользователей, которые находятся у изначального пользователя в друзьях и фолловерах.

Пример

Original-user.friends - выдает всех друзей оригинального пользователя.

Original-user.followers - выдает всех фолловеров оригинального пользователя.

Пользователь может писать стейтмент, т.е. у одного пользователя может быть несколько стейтментов, но у одного стейтмента только один пользователь.

3.6.1.2 Стейтмент Может иметь один или несколько комментариев. Принадлежит в свою очередь одной категории - пользовательский стейтмент, стейтмент в разделе акция или мероприятие.

Стейтментам можно ставить лайк. Таким образом стейтмент появляется в ленте всех тех людей, у которых есть в друзьях тот, кто поставил лайк. Также, стейтмент, составленный человеком, который есть у пользователя в друзьях или фолловерах, появится в ленте пользователя.

3.6.1.3 Гео-данные У страны может быть много городов. У города может быть много зданий. У здания может быть много этажей, у этажа может быть много отсеков.

3.6.1.4 Отсек Отсек может иметь категорию - комната, стол+стул, переговорка, конференц-зал и прочее.

Отсек можно бронировать через соответствующую форму. При этом он закрепляется за его забронировавшим пользователем и у пользователя списываются credits. После бронирования на карте отсек помечается красным, если выбрано соответствующее время.

4 Модули

4.1 Оплата

Необходимо встроить прием онлайн-оплаты через агрегатор "CloudPayments". При успешной оплате формируется заказ и клиент перенаправляется на соответствующую success-страницу.

4.2 Интерактивная карта

Сердце данного приложения.

5 Страницы

5.1 Личный кабинет

5.2 Типовые страницы

- Контакты (impressum) - необходимо подгружать из настроек проекта
- О проекте - текстовая страница
- Соглашение пользователя - текстовая страница
- Конфиденциальность - текстовая страница
- Новости - управляется соответствующим блоком в администраторской панели (либо отдельный соответствующий модуль битрикса, либо инфоблок - на усмотрение разработчика)
- Часто задаваемые вопросы - управляется блоком в админке (инфоблок)
- Обратная связь - контактная форма, присылает сообщение на адрес из настроек магазина (проработать возможность подключить битрикс-модуль ТП, т.е. что бы ответы из модуля приходили клиенту на почту)
- Регистрация пользователя - клиент должен ввести мэйл и пароль. На почту ему приходит подтверждение о регистрации.
- Забытый пароль - страница позволяет ввести мэйл и восстановить пароль