

1. Укажите полезные источники по laravel и другим темам

<https://learn.javascript.ru> - для изучения JS

<https://laravel.com/docs/8.x/installation> - документации для laravel

<https://laracasts.com/series?curated> - курсы по laravel

<https://developer.mozilla.org/ru/> - документация по css/js/html и т.д.

2. Управление проектом/задачами/исполнителями

Управление в целом не получилось по причинам:

- отсутствие навыков управления у лида
- отсутствие средств мотивации участников команды
- вникать в проблемы других участников для выставление задач
- временное отсутствие члена команды - что привело к рассинхрону по статусу выполнения задач

Получилось:

- Назначать задачи по умениям участников команды

3. Сбор, анализ и формализация требований

Очень много времени было потрачено на то чтобы собрать требования и формализовать их и составить по ним сценарий и макеты. (Может повлияли особенности системы или не сразу взаимопонимание между преподавателем и командой)

4. Прототипирование

Особых проблем не было. За исключением поиска хорошего материалов и примеров на php.

5. Моделирование структуры данных

Особых проблем не было. Долго думали над тем как правильно организовать хранение истории о переводах.

6. Макетирование интерфейса

Макеты получились сразу хорошими. Получилось это скорее всего из-за опыта Марии в данном вопросе и тем что она занималась написанием сценариев и хорошо поняла смысл системы.

7. Моделирование бизнес-процессов

Бизнес процессы были описаны диаграммами состояния и сценариями. Проблемы были с пониманием диаграмм состояния. А также с некорректными формулировками в сценарии которые потом со временем исправлялись.

8. Моделирование состояния ключевых сущностей

Вроде у нас нет состояния у ключевых сущностей.. Или я неправильно понял вопрос..

9. Архитектурные решения.

Архитектура проекта должна реализовывалась с помощью паттерна MVC, а также созданием REST API для других клиентов. В целом получилось, но сама по себе MVC - сложная в ГРАМОТНОЙ реализации архитектура и требует проработки длительной и предварительного проектирования. Так что не уверен, что у нас получился хороший MVC.

10. Конструирование программы

Незнание языка php и особенностей фреймворка laravel приводило иногда к сомнительным решениям и “костылям”. Скорее всего из-за того что большая часть информации о php проектах - сомнительного качества. Может стоит перейти на какой-то более строгий фреймворк. (Например Spring). Либо использовать стек, который используется на предмете “Основы web-разработки”.

11. Совместное владение кодом

Проблем с ведением репозитория на гитхабе не было. Так как все создавали для каждой фичи - свою ветку и над каждой фичей работал один человек. Также велась одна ветка в которую один человек все мерджил.

12. Тестирование

В плане реализации - ничего не получилось. Хромала архитектурная часть, частый изменения в проекте который меняли многое и нехватка времени под конец. Так что изучить получилось только в теории.

13. Отладка и исправление ошибок

В отладке помогали отладочные функции типа ddd или dd. Также в PhpStorm можно настроить отладку для пошагового выполнения кода, но нам особо это не понадобилось.

14. Развертывание конечного приложения

Особых проблем не было. Все замечания с которыми столкнулись - написали в вики. Антон Викторович доставлял зависимости и необходимые библиотеки. Вообще у нас был настроен CI/CD на TeamCity который делал эту рутинную работу за нас при каждом пуше коммитов в ветку develop. Так что мы вообще после первой настройки в сентябре CI/CD на сервер сам больше не заходило - проект обновлялся и разворачивался сам.

15. Время - как ресурс: что получилось

В основном время уходило на компилятор, ну и личные дела. Также было сложно скооперироваться в одно и тоже время (вне пар) чтобы что-то или посидеть и подумать над какой-нить задачей или проблемой всей командой.

16. Разное

Может стоит пересмотреть задачу, как-то уточнить требования. Сформировать ее заранее и, чтобы все преподаватели четко видели результат! То есть знали - что должно получиться! В этот раз было все субъективно в плане того, что кто-то хочет так, другой говорит - что это плохо и т.д.

Также было бы неплохо сделать так, чтобы каждый преподаватель именно вел работу студентов и он же ее оценивал! А не перекидывать это все на тимлида, который не может из-за отсутствия опыта это все сделать!

Про идею с реализацией ORM - говорил лично. Можно подумать! Либо что-то подобное. Я все-таки за то, чтобы мы объединяли прошлые знания и может не научились чему-то новому, но хотя бы поняли какую-то сферу более глубоко, применяя знания, которые были получены на других предметах!