МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Проектирование и архитектура программных систем»

Лабораторная работа № 2

Выполнил:

студент гр. ПИбд -31

Черноморченко А.В.

Проверил:

Скалкин А. М

Ульяновск (2020).

Лабораторная работа №2 Вар. 14.

Цель работы: фиксирование и документирование этапа проектирования решения поставленной задачи в соответствии с вариантом.

Требования:

• Определить на текущий момент проектирования:

1. цель архитектуры (начальное определение задач архитектуры, определение потребителей архитектуры, определение ограничений)

2. основные сценарии (бизнес-критические и имеющие большое влияние)

3. общее представление приложения (определение типа приложения, ограничений развертывания, значащих архитектурных стилей проектирования и выбор подходящих технологий)

4. потенциальные проблемы (потенциальные и критически важные)

5. выбор показателей качества.

6. сквозная функциональность.

7. варианты решений (базовая или пилотная архитектура) •

Предоставить графическое представление архитектуры

Фиксирование и документирование этапа проектирования решения поставленной задачи.

Требования:

* Определить на данный момент проектирования:

1. цель архитектуры (начальное определение задач архитектуры, определение потребителей архитектуры, определение ограничений)

* Клиент-серверная модель системы, позволяющая скрыть детали реализации от пользователя и представляющая собой удобную форму взаимодействия между клиентом и сервером.
* Понятность системы. Быстрая и эффективная отладка для восстановления работоспособности.

2. Определение потребителей архитектуры:

2.1. Разработчики

2.2. Специалисты поддержки

3. Определение ограничений

3.1. Необходимо наличие постоянного интернет-соединения для функционирования каждой части системы.

3.2. Сложный процесс настройки безопасности базы данных в СУБД MySQL.

3.3. Сложность одновременной обработки большого количества запросов веб-сервером Apache.

4. Основные сценарии (бизнес-критические и имеющие большое влияние)

4.1. Бизнес-критические

* Регистрация пользователя
* Авторизация пользователя в системе с клиентского приложения
* Процесс выбора товаров и оформление заказа
* Процесс оплаты

4.2. Имеющие большое влияние

* Автоматическое оповещение пользователя об изменении статуса заказа

5. Общее представление приложения

5.1. Тип

* Клиент-серверное приложение с тонкими клиентами на разных платформах: Android, Web.

5.2. Ограничения развертывания

* ОС сервера должна поддерживать php 7.4.9, Laravel API (Windows/Linux).
* Сервер с Laravel API разворачивается в закрытой сети и имеет открытый порт для общения с клиентами, БД доступна только по запросам от Laravel API, поддерживает механизм BasicAuth(логин/пароль).
* Для работы Android приложения необходим Android 8+
* Для работы WEB приложения необходимы браузеры, поддерживающие HTML5, JS

5.3. Выбор технологий разработки

* Эмулятор Android устройства Nexus 4 (Android 8+).
* Язык программирования: Php 7.4.9 (IDE: PhpStorm), Dart (IDE: Android Studio), JS (IDE: WebStorm), язык разметки для Web-приложения: HTML5.
* СУБД: MySQL.
* Фреймворк, реализующий возможность поддержки API контроллеров: Laravel API.
* Blackfire Profiler (инструмент, позволяющий осуществлять сбор данных о серверных ресурсах и деятельности пользователей).

6. Потенциальные проблемы (потенциальные и критически важные)

6.1. Возможные сбои интернет-соединения, приводящие к возникновению ошибок в работе системы.

6.2. Возможный сбой в платежной системе при возникновении сетевых неполадок.

6.3. Возникновение DDoS-атак.

7. Выбор показателей качества

7.1. Безопасность

* Информация о действиях в мобильном и веб-приложении записывается в специальные файлы (логи), таким образом можно отследить деятельность злоумышленника.
* Взаимодействие с базой данных происходит не напрямую, а через API запросы.
* Зашифрованный протокол передачи данных (HTTPS).

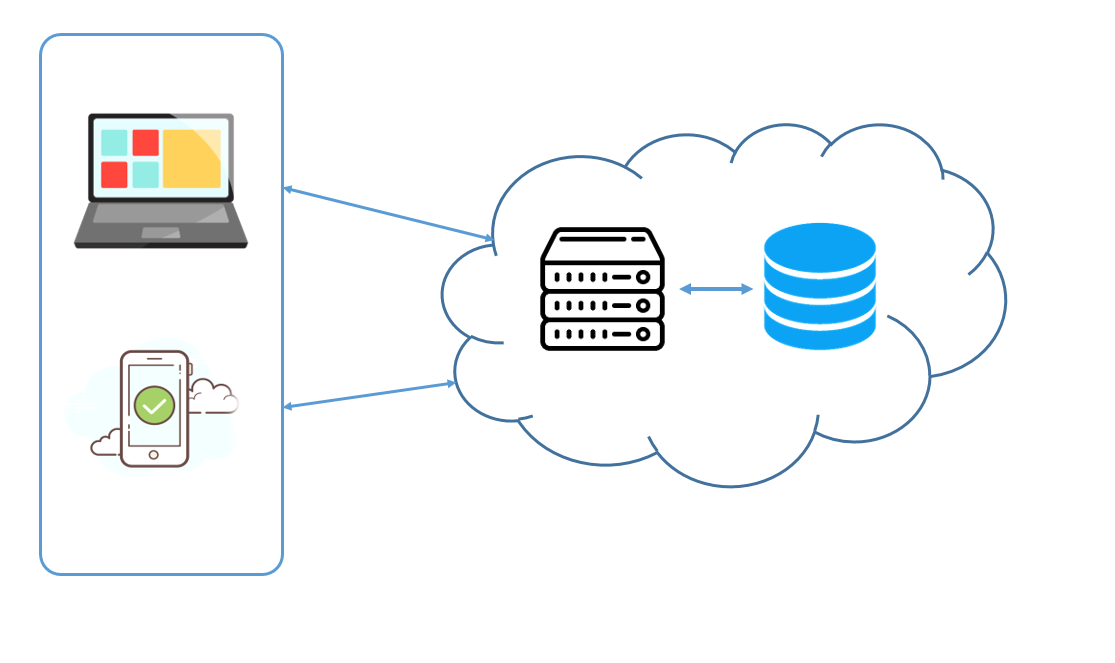
7.2. Быстродействие отклика сервера на запрос клиентского приложения

7.3. Лёгкость установки мобильного приложения

8. Сквозная функциональность.

* Авторизация пользователей для разграничения доступа
* Протоколирование. Все действия в мобильном и веб-приложении записываются в специальные файлы (логи), благодаря выбранному инструменту Blackfire Profiler очень удобно настраивать нужную информацию для отслеживания в удобной для просмотра форме.

9. Варианты решений: (см. ниже)



API-запросы

API-запросы

Вывод: благодаря данной лабораторной работе я научилась фиксировать и документировать этапы проектирования.