Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра ИВТиИБ

Отчёт защищён с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_ А.А.Третьяков.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчёт по лабораторной работе №3

«Командная разработка вариантов архитектуры программной реализации проекта и вариантов интерфейса»

по дисциплине «Управление IT-проектами»

ЛР 09.04.01.02.05.08.000 О

Студенты группы 8ИВТ-11 Е.А.Богер, С.М. Емелин, А.Ю.Мысин

Руководитель работы к.т.н. А.А.Третьяков

Барнаул 2022

**Цели и задачи работы:** получение навыков командной разработки вариантов архитектуры программной реализации проекта и вариантов интерфейса.

**Задание к работе:**

Команде самостоятельно выполнить задачу согласно методике в соответствии с индивидуальным вариантом.

**Методика выполнения работы:**

1. Разработать варианты общей архитектуры системы.

2. Разработать варианты пользовательского интерфейса.

3. Представить варианты на утверждение заказчику.

4. Уточнить план разработки и тестирования ПО, представить преподавателю график его исполнения членами команды по рубежным точкам в виде диаграммы Ганта.

**Ход работы**

При разработке программного обеспечения появилась необходимость хранить большое количество данных, которые содержат следующие категории:

* Список автомобилей;
* Клиенты;
* Персонал (Администраторы и Менеджеры);
* Перечень дополнительных услуг;
* Перечень заказов;
* История переписки клиентов с персоналом.

Требуется реализовать в приложении возможность добавлять, редактировать и удалять данные, а также проводить фильтрацию данных.

Рассмотрим несколько вариантов общей архитектуры для данного ИТ-проекта.

* Архитектура "файл-сервер";
* Двухзвенная архитектура "клиент-сервер";
* Многозвенная архитектура "клиент-сервер";
* Архитектура Веб-приложений.

**Архитектура "файл-сервер"**

Файл-серверные приложения – приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных.

* Функции сервера: хранения данных и кода программы.
* Функции клиента: обработка данных происходит исключительно на стороне клиента.

Достоинства такой архитектуры:

* многопользовательский режим работы с данными;
* удобство централизованного управления доступом;
* низкая стоимость разработки;
* высокая скорость разработки;
* невысокая стоимость обновления и изменения ПО.

Недостатки:

* проблемы многопользовательской работы с данными: последовательный доступ, отсутствие гарантии целостности;
* низкая производительность (зависит от производительности сети, сервера, клиента);
* плохая возможность подключения новых клиентов;
* ненадежность системы.

**Двухзвенная архитектура "клиент-сервер"**

Клиент-сервер (Client-server) – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

Преимуществами данной архитектуры являются:

* возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети;
* все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов, а также на сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;
* поддержка многопользовательской работы;
* гарантия целостности данных.

Недостатки:

* неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;
* администрирование данной системы требует квалифицированного профессионала;
* высокая стоимость оборудования;
* бизнес логика приложений осталась в клиентском ПО.

**Многозвенная архитектура "клиент-сервер"**

Многоуровневая архитектура клиент-сервер (Multitier architecture) – разновидность архитектуры клиент-сервер, в которой функция обработки данных вынесена на один или несколько отдельных серверов. Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов.

Плюсами данной архитектуры являются:

* клиентское ПО не нуждается в администрировании;
* масштабируемость;
* конфигурируемость – изолированность уровней друг от друга позволяет быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней;
* высокая безопасность;
* высокая надежность;
* низкие требования к скорости канала (сети) между терминалами и сервером приложений;
* низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости.

Минусы:

* растет сложность серверной части и, как следствие, затраты на администрирование и обслуживание;
* более высокая сложность создания приложений;
* сложнее в разворачивании и администрировании;
* высокие требования к производительности серверов приложений и сервера базы данных, а, значит, и высокая стоимость серверного оборудования;
* высокие требования к скорости канала (сети) между сервером базы данных и серверами приложений.

**Архитектура Web-приложений**

Под **web-приложением** понимается прикладная программа, разработанная по архитектуре "клиент-сервер", использующая в качестве клиента web-браузер и работающая на стороне web-сервера.

Можно выделить следующие основные особенности веб-архитектуры:

* отсутствие необходимости использовать дополнительное ПО на стороне клиента – это позволяет автоматически реализовать клиентскую часть на всех платформах;
* возможность подключения практически неограниченного количества клиентов;
* благодаря единственному месту хранения данных и наличия системы управления базами данных обеспечиваются минимальные требования для поддержания целостности данных;
* доступность при работоспособности сервера и каналов связи;
* недоступность при отсутствии работоспособности сервера или каналов связи;
* достаточно низкая скорость Веб сервера и каналов передачи данных;
* относительно объема данных – архитектура Веб систем не имеет существенных ограничений.

Web-приложения в сравнении с локальными приложениями имеют следующие преимущества:

* простота доступа к приложению – любой человек, имеющий компьютер, подключенный к сети Интернет, может использовать web- приложение;
* простота развертывания (установки) – в отличие от локальных приложений web-приложения после завершения разработки не требуют установки на компьютерах пользователей. Достаточно только сообщить им URL-адрес приложения. При изменении приложения все пользователи сразу начинают работать с измененной версией;
* наличие большого количества обученных пользователей;
* высокий уровень развития и надежности сетевых соединений и web-технологий.

Однако web-приложения имеют и свои недостатки, перечисленные ниже.

* "Слабосвязапная" архитектура web-сети – отсутствие поддержки состояния сеанса работы и задержка при перезагрузке каждой страницы. Каждая перегрузка (или обновление страницы) вызывает заметную задержку, вызванную необходимостью установить НТТР- соединение, обработать запрос на сервере, передать по сети ответное HTTP-сообщение и перезагрузить страницу браузером. Это создает скачки и прерывистый режим работы пользователя.
* Ограниченный набор элементов управления для проектирования форм приложения. Текущая версия языка HTML поддерживает только ограниченный набор элементов управления (текстовые элементы, радиокнопки, флажки, выпадающие списки и командные кнопки).

**Варианты пользовательского интерфейса:**

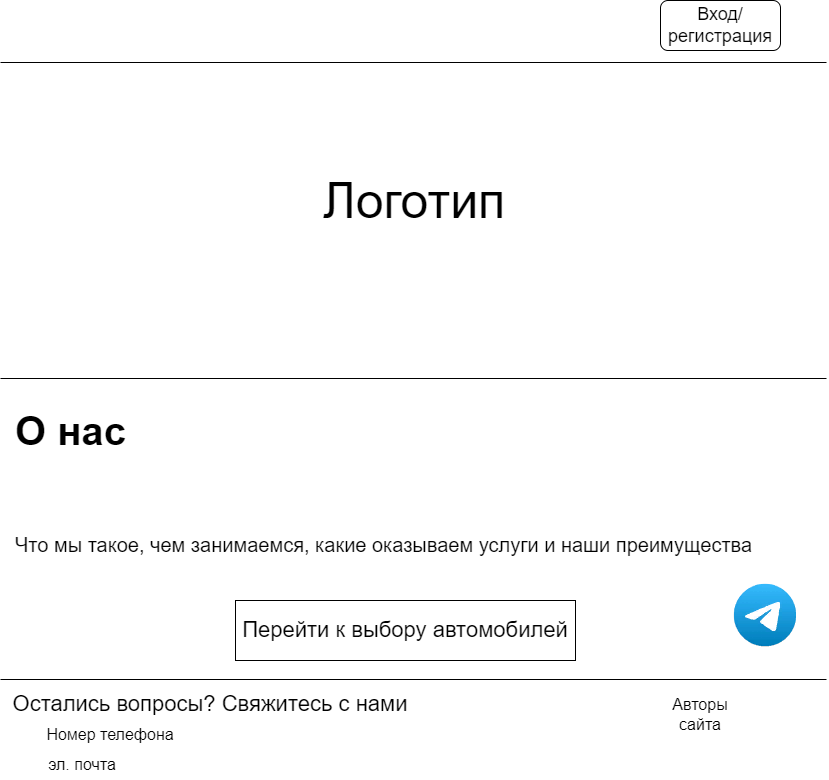
****

Рисунок 1 – Главная страница

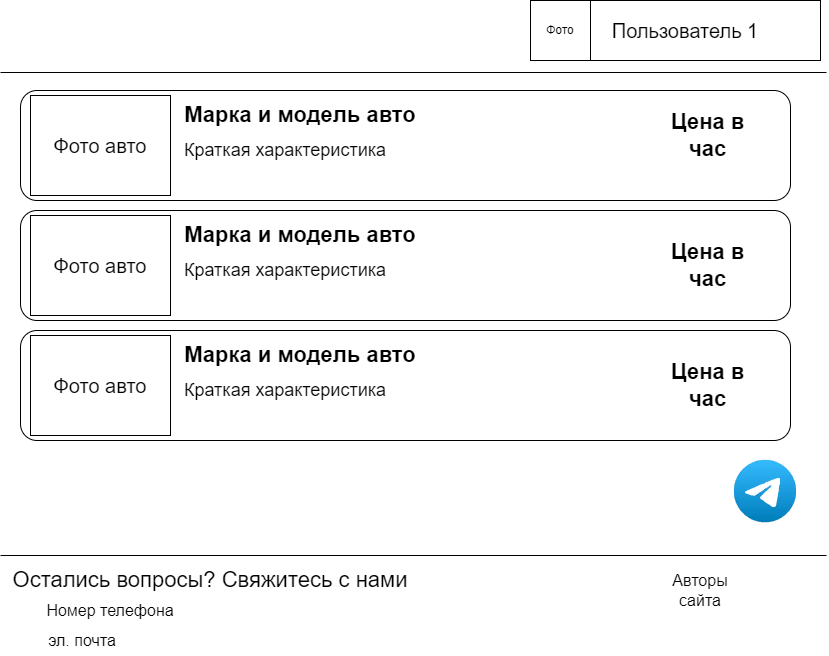


Рисунок 2 – Страница выбора автомобилей

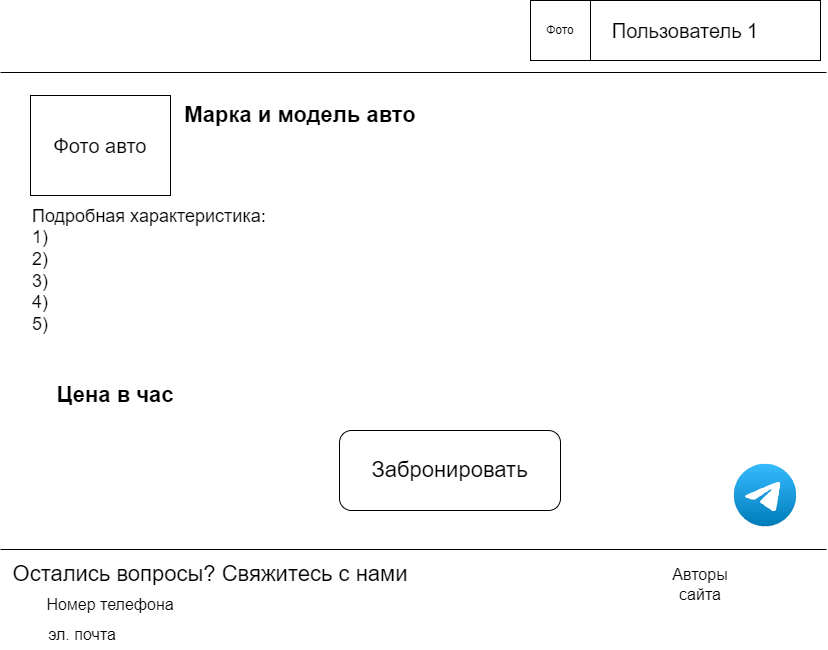


Рисунок 3 – Страница автомобиля

**График исполнения плана разработки и тестирования ПО членами команды по рубежным точкам в виде диаграммы Ганта:**

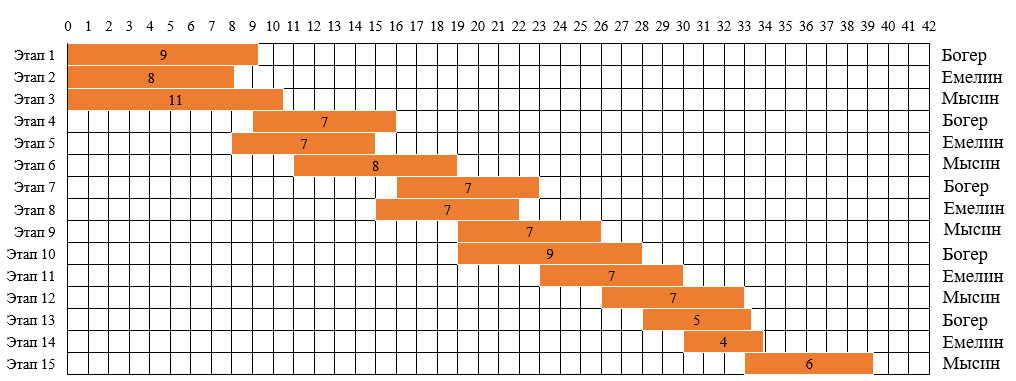


Рисунок 4 - Диаграмма Ганта

**Вывод:** В ходе проделанной работы были созданы наброски вариантов пользовательского интерфейса, составлена диаграмма Ганта. В качестве архитектуры исполнения была выбрана архитектура Web-приложения.