Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 2

По дисциплине «Разработка и анализ требований»

На тему «Анализ требований. Декомпозиция. Приоритезация»

Выполнил:

Студент 2 курса 9 группы

Павлович Ян Андреевич

Преподаватель: асс. Ромыш А.С.

2025, Минск

**Содержание**

1. Введение
2. Основные роли
3. Карта пользовательских историй
4. Теоретическая часть
5. Заключение

**Введение**

Автосервисы сталкиваются с проблемами в организации работы, что приводит к долгому ожиданию клиентов и неэффективному распределению ресурсов. Разрабатываемое приложение автоматизирует запись клиентов, координацию сотрудников и оплату услуг, повышая удобство и скорость обслуживания. Данный документ представляет анализ пользовательских историй (User Story Mapping) для оптимального проектирования системы.

**Основные роли в системе**

Клиент – записывается на обслуживание, отслеживает статус, оплачивает услуги.

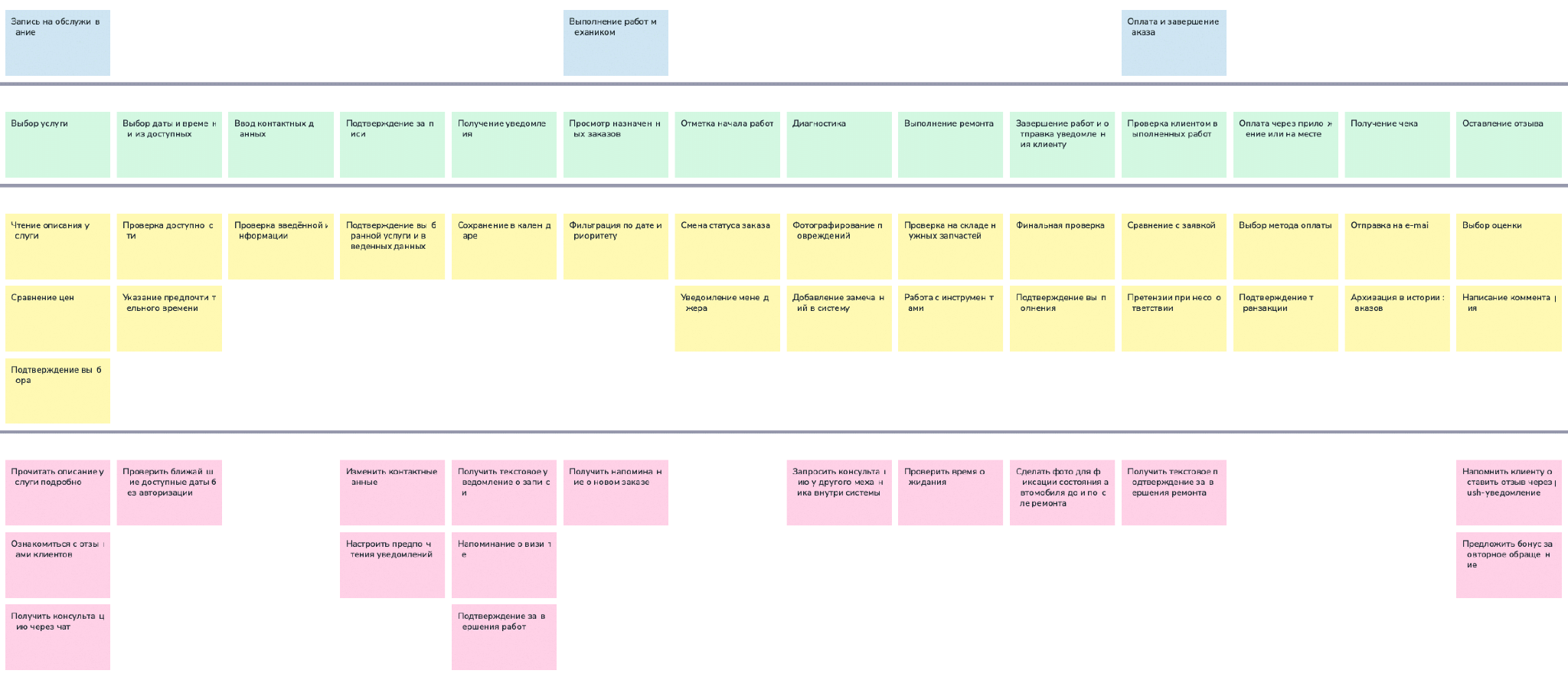
Администратор – управляет сервисами, настраивает график, следит за пользователями.

Менеджер – принимает заявки, информирует клиентов, отвечает на вопросы.

Механик – видит список заказов, выполняет работу, отмечает выполненные услуги.

**Карта пользовательских историй**

User Story Mapping для автосервиса



1. Действия

- Запись на обслуживание

- Выполнение работ механиком

- Оплата и завершение заказа

2. Шаги (основные подзадачи, необходимые для выполнения действий)

Запись на обслуживание

- Выбор услуги (ТО, замена масла, диагностика и т. д.)

- Выбор даты и времени из доступных слотов

- Ввод контактных данных

- Подтверждение записи

- Получение уведомления

Выполнение работ механиком

- Просмотр назначенных заказов

- Отметка начала работ

- Диагностика (добавление замечаний, если нужно)

- Выполнение ремонта

- Завершение работ и отправка уведомления клиенту

Оплата и завершение заказа

- Проверка клиентом выполненных работ

- Оплата через приложение или на месте

- Получение чека

- Оставление отзыва

3. Детали (мелкие конкретные действия)

Запись на обслуживание

- Выбор услуги → Чтение описания услуги → Сравнение цен → Подтверждение выбора

- Выбор даты и времени → Проверка доступности → Указание предпочтительного времени

- Ввод контактных данных → Проверка введенной информации → Подтверждение

- Получение уведомления → Сохранение в календаре

Выполнение работ механиком

- Просмотр назначенных заказов → Фильтрация по дате и приоритету

- Отметка начала работ → Смена статуса заказа → Уведомление менеджера

- Диагностика → Фотографирование повреждений → Добавление замечаний в систему

- Выполнение ремонта → Проверка на складе нужных запчастей → Работа с инструментами

- Завершение работ → Финальная проверка → Подтверждение выполнения

Оплата и завершение заказа

- Проверка выполненных работ → Сравнение с заявкой → Претензии при несоответствии

- Оплата через приложение → Выбор метода оплаты → Подтверждение транзакции

- Получение чека → Отправка на e-mail → Архивация в истории заказов

- Оставление отзыва → Выбор оценки → Написание комментария

4. Поддерживающие взаимодействия (Supporting Interactions)

Запись на обслуживание

- Получить консультацию перед записью (чат с менеджером)

- Ознакомиться с отзывами других клиентов

- Проверить ближайшие доступные даты без авторизации

- Получить напоминание о визите через SMS или push-уведомление

Выполнение работ механиком

- Получить напоминание о новом заказе (push-уведомление для механика)

- Проверить наличие запчастей перед началом работ

- Запросить консультацию у другого механика внутри системы

- Сделать фото для фиксации состояния автомобиля до и после ремонта

Оплата и завершение заказа

- Получить текстовое подтверждение завершения ремонта

- Запросить электронный чек на e-mail или мессенджер

- Напомнить клиенту оставить отзыв через push-уведомление

- Предложить бонус за повторное обращение (программа лояльности)

**Теоретическая часть**

Что означает акроним INVEST?

I – Independent (Независимость). Истории могут быть реализованы в любом порядке.

N – Negotiable (Обсуждаемость). Можно менять детали без ущерба для общей логики.

V – Valuable (Ценность). История приносит пользу пользователю или бизнесу.

E – Estimable (Оцениваемость). Можно оценить сложность реализации.

S – Small (Компактность). История не слишком большая.

T – Testable (Тестируемость). Можно проверить выполнение требований.

Для чего используется User Story Mapping?

- Определение структуры и функционала продукта.

- Выявление важных функций для MVP.

- Разбиение проекта на релизы.

- Улучшение коммуникации в команде.

Методы приоритезации требований

1. MoSCoW

Must – обязательно.

Should – желательно.

Could – возможно, если останется время.

Won't – в этот раз не будет реализовано.

2. RICE – Reach (охват), Impact (влияние), Confidence (уверенность), Effort (затраты).

3. WSJF – сначала реализуются короткие и важные задачи.

**Заключение**

Создание карты пользовательских историй позволяет наглядно представить взаимодействие пользователей с системой, определить ключевые функциональные требования и приоритетные задачи. Это способствует эффективному планированию разработки и улучшению пользовательского опыта. Использование методов приоритизации (MoSCoW, RICE, WSJF) помогает выделить наиболее значимые функции для MVP и дальнейших релизов. В результате разрабатываемое программное средство обеспечит удобство работы автосервиса и повысит качество обслуживания клиентов.