

Задача 3

На основании требований, собранных в задании 2, напиши задание для разработчика. Перечисли в задании функциональность и логику работы приложения, а также то, в каком порядке ее нужно реализовать, чтобы клиенты могли оставлять чаевые онлайн. Описывать внутреннее устройство платежной системы банка не нужно.

Решения присылайте в поле для ответа в виде ссылки на онлайн документ.

Решение:

1. Общая информация о ИС

1) **Название:** Автоматизация процесса получения чаевых в кафе

2) **Цель:** Увеличить долю оплаты чаевых через приложение до 90% в течение 6 месяцев после запуска приложения.

3) **Задачи:**

- Обеспечить комфортный и безопасный процесс отправки чаевых для клиентов.
- Обеспечить удобство и простоту использования приложения для официантов, менеджеров и администраторов.
- Интегрировать приложение с платежными системами для приема платежей онлайн.
- Обеспечить безопасность данных и соответствие требованиям законодательства.

2. Предполагаемый технологический стек

Фронтенд:

1) Android:

- Язык программирования Kotlin.
- Фреймворк:** Jetpack Compose

2) iOS:

- Язык программирования Swift
- Фреймворк SwiftUI

Бэкенд:

- 1) Язык программирования: Python
- 2) Фреймворк: Django
- 3) База данных SQL
- 4) Кеширование: Redis
- 5) Сетевые протоколы: HTTPS
- 6) Шифрование данных: AES
- 7) Защита от DDoS-атак: Cloudflare
- 8) Соблюдение стандартов: PCI DSS

Выбор технологического стека основывается на следующих критериях:

- 1) Функциональность - поддержка всех необходимых функций для реализации проекта.
- 2) Производительность - высокая скорость работы и масштабируемость системы.
- 3) Безопасность - надежная защита данных и соответствие требованиям законодательства.
- 4) Сообщество и поддержка - активное сообщество разработчиков и доступность документации.
- 5) Стоимость - оптимальное соотношение цены и качества.

3. Описание выполняемых бизнес-процессов

3.1. Регистрация и авторизация

1) Клиент:

- Регистрация с использованием номера телефона или электронной почты.
- Подтверждение регистрации через SMS или email.
- Авторизация с использованием пароля и номера телефона или электронной почты.
- Восстановление пароля через SMS или email.

2) Официант:

- Регистрация официанта осуществляется администратором через панель управления.
- Авторизация с использованием пароля и номера телефона или электронной почты.
- Восстановление пароля через SMS или email.

3) Администратор и менеджер:**

- Регистрация с использованием номера телефона или электронной почты.
- Подтверждение регистрации через SMS или email.
- Авторизация с использованием пароля и номера телефона или электронной почты.
- Восстановление пароля через SMS или email.

3.2. Создание и оплата счета

1) Официант:

- Создание счета для клиента с указанием номера столика и трех вариантов суммы чаевых на выбор.

2) Клиент:

- Просмотр счета в приложении.
- Выбор суммы чаевых из предложенных вариантов или отказ от оплаты чаевых.
- Выбор способа оплаты.
- Оплата счета одним нажатием.
- Повторная попытка оплаты в случае неудачи.

3) Система:

- Проверка корректности данных счета.
- Передача данных оплаты в платежную систему.
- Обработка результата оплаты.
- Отправка уведомлений клиенту и официанту о результатах оплаты.

3.3. Управление приложением

1)Администратор и менеджер:

- Просмотр общей статистики по чаевым.

- Просмотр статистики по чаевым с возможностью фильтрации по дате и официанту.
- Генерация отчетов по статистике чаевых в удобном формате.
- Добавление, удаление и редактирование информации об официантах.
- Настройка системы уведомлений.

4. Описание интерфейсов (пользовательских и интеграционных) и алгоритмов обработки объектов/данных, используемые справочники и формы отчётов

4.1. Пользовательские интерфейсы

1) Клиент:

- Экран регистрации/авторизации.
- Экран просмотра счета.
- Экран выбора суммы чаевых и способа оплаты.
- Экран успешной оплаты.
- Экран профиля с возможностью просмотра истории платежей и оставления отзыва.

2) Официант:

- Экран авторизации.
- Экран создания счета.
- Экран уведомлений о полученных чаевых.

3) Администратор и менеджер:

- Панель управления с возможностью просмотра статистики, управления официантами и настройки уведомлений.

4.2. Интеграционные интерфейсы

1) Платежные системы:

- API для приема платежей онлайн. С функцией создания платежных сессий, подтверждение платежей, обработка возвратов.

2) Уведомления:

- API для отправки push-уведомлений клиентам и официантам.

4.3. Алгоритмы обработки объектов/данных

1) Регистрация и авторизация:

- Проверка корректности данных регистрации/авторизации.
- Отправка кода подтверждения через SMS или email.
- Проверка кода подтверждения.
- Создание/аутентификация пользователя.

2) Создание и оплата счета:

- Проверка корректности данных счета.
- Передача данных оплаты в платежную систему.
- Обработка результата оплаты.
- Отправка уведомлений клиенту и официанту.

3) Управление приложением:

- Просмотр и фильтрация статистики по чаевым.
- Генерация отчетов.
- Добавление, удаление и редактирование информации об официантах.
- Настройка системы уведомлений.

4.4. Используемые справочники и формы отчётов

1) Справочники:

- Справочник официантов - номер телефона, ФИО.
- Справочник столиков - номер столика.

2) Формы отчётов:

- Отчет по общей статистике чаевых.
- Отчет по статистике чаевых с фильтрацией по дате и официанту.

5. Роли пользователей и доступные им функции, статусные модели объектов учёта

5.1. Роли пользователей

1) Клиент:

- Регистрация и авторизация.
- Просмотр счета.
- Выбор суммы чаевых и способа оплаты.
- Оплата счета.

- Просмотр истории платежей.
- Оставление отзыва.

2) Официант:

- Авторизация.
- Создание счета.
- Просмотр уведомлений о полученных чаевых.

3) Администратор:

- Авторизация.
- Просмотр статистики по чаевым.
- Управление списком официантов.
- Настройка системы уведомлений.

4) Менеджер:

- Авторизация.
- Просмотр статистики по чаевым.
- Управление списком официантов.
- Настройка системы уведомлений.

5.2. Статусные модели объектов учёта

1) Счет:

- Создан.
- Оплачен.
- Отменен.

2) Оплата:

- Ожидает оплаты.
- Оплачен.
- Отклонен.

6. Нефункциональные требования

1) Производительность:

- Система должна обрабатывать не менее 20 запросов в секунду при пиковых нагрузках.
- Приложение должно загружаться быстро, не более 5 секунд на среднестатистическом устройстве.

2) Доступность:

- Система должна быть доступна не менее 99,9% времени в течение года.

3) Безопасность:

- Использование шифрования данных при передаче и хранении.
- Использование безопасных протоколов для передачи данных.
- Ограничение доступа к персональным данным и платежной информации.
- Соблюдение требований законодательства в области защиты персональных данных и безопасности платежей.

4) Удобство использования:

- Минималистичный и легкий дизайн интерфейса.
- Интуитивно понятный и доступный для пользователей с разным уровнем технической подготовки.

5) Обслуживаемость:

- Легкость обслуживания и поддержки приложения.
- Возможность быстрого выявления и устранения неисправностей.

Источники информации:

1) Статья " Составляем понятное ТЗ для разработчика и заказчика" (ссылка на сайт <https://tproger.ru/articles/sostavljaem-ponjatnoe-tz-dlja-razrabotchika-i-zakazchika>).

2) Стандарты безопасности платежей (ссылка на сайт <https://robokassa.com/content/pci-dss.html>).