**Доклад**

**Слайд 1.**

Здравствуйте, уважаемые председатель и члены государственной аттестационной комиссии. Разрешите представить вам выпускную квалификационную работу, посвященную разработке программного модуля для проведения финансовых операций на POS-терминале. Шифр – ПМ ФО.

Разрабатываемый программный модуль обеспечивает повышение эффективности проведения финансовых операций и расширение функционала POS-терминалов. Цель и задачи выпускной квалификационной работы представлены на слайде № 1.

**Слайд 2.**

На слайде 2 представлено исследование предметной области. По результатам исследования был определен ряд проблем, с которыми сталкиваются как разработчики, так и пользователи POS-терминалов.

**Слайд 3.**

Слайд 3 содержит результаты анализа существующих программных и аппаратных решений. Из таблицы, представленной на слайде, наглядно видно, что существующие решения не обеспечивают всех потребностей.

**Слайд 4.**

Разработка была начата с выбора языка программирования. Были рассмотрены следующие языки программирования: С, C++, C#, Java, Kotlin. Проанализировав результаты сравнения по представленным критериям на слайде 4, были выбраны языки Java и Kotlin.

Для выбранных языков программирования был рассмотрен ряд сред разработки: Eclipse, Microsoft Visual Studio, IntelliJ IDEA, Android Studio. По приведенным в таблице результатам была выбрана наиболее подходящая среда Android Studio.

**Слайд 5.**

На слайде 5 представлена схема данных программного модуля. Для использования ПМ ФО его необходимо сконфигурировать. Конфигурация происходит с помощью XML файлов.

Для проведения финансовой операции программному модулю необходимо получить тип операции, сумму или номер счета.

Также запрос на операцию может прийти от кассы в виде сообщения протокола ТТК.

После получения всех необходимых данных программный модуль будет ждать данные карты с трех различных считывателей.

Далее происходит проведение EMV процессинга. Данные операции до отправки на сервер сохраняются в базу данных. После этого происходит процесс обмена сообщениями с сервером банка, сохранение в базу данных информации о решении по операции и формирование чека.

**Слайд 6.**

На слайде 6 представлен алгоритм работы программного модуля. Для начала, необходимо провести инициализацию. Далее пользователю предлагается выбрать тип финансовой операции и ввести необходимые данные.

После терминал начинает ждать банковскую карту. Если с картой все в порядке, происходит считывание данных. Если на карте есть несколько поддерживаемых терминалом приложений, то пользователю предлагается выбрать одно их них.

После этого выполняется EMV процессинг. Далее выполняется проверка держателя карты. Если проверка не пройдена, операция завершается с ошибкой. Если проверка пройдена, то в зависимости от того, онлайн операция или офлайн, происходит сборка сообщения для банка, или терминал решает сам, одобрить операцию или отклонить.

После проведения операции, пользователю сообщается о результате и происходит печать чека.

**Слайд 7.**

На 7-ом слайде представлены экранные формы пользовательского интерфейса.

Экранная форма «Обычные операции» предоставляет доступ к самым частым операциям: Продажа и просмотр журнала.

Сервисная экранная форма предоставляет дополнительные функции, такие как сверка с банком, проверка сервера банка, изменения параметров и др.

Журнал операций предоставляет доступ к проведенным операциям, для их просмотра, а также предоставляет функции отмены возврата или печати копии чека.

**Слайд 8.**

На слайде 9 представлены процесс отладки и тестирования программного модуля с помощью стандартного отладчика Android Studio и библиотеки для реализации автоматизированного тестирования Junit 5. На первой экранной форме показан процесс отладки с помощью системы брейкпоинтов, на второй отображен набор автоматизированных тестов.

**Слайд 9.**

На слайде 9 представлена апробация реализованного программного модуля.

**Слайд 10.**

Показаны результаты работы.

Доклад окончен. Спасибо за внимание.