1. **Слайд(Введение)**

Здравствуйте, я студент 3 курса ликино-дулевского политехнического колледжа колледжа, буду защищать мобильный проект, который создавался на платформе Xamarin используя Xamarin.forms. основная цель приложения обеспечить комфортное прохождение тестирования, и соответсвенно просмотр результатов. Разработка велась на языке программирование с#, хранение данных было реализовано в облачной базе данных firebase. Перейдем к следующем слайду.

1. **Слайд(Интерфейс 1 ЧАСТЬ)**

На данном слайде показаны следующие страницы – Страница авторизации, главная, где пользователь может увидеть категории выбранной темы, также просмотреть учебный материал, Результаты, и страница редактирование вопроса, перейдем к следующей части интерфейса

1. **Слайд(Интерфейс 2 ЧАСТЬ)**

Тут изображены следующие: страница категории, где отображается список тестов, страница прохождения теста, просмотр уже пройденных тестов, и просмотр чужихрезультатов тестирование, который доступен только пользователям с ролью «Admin»,

1. **Слайд(База данных)**

В мобильно приложении используется не реляционная база данных firebase, в которой

Данные хранятся в формате JSON и синхронизируются в режиме реального времени с каждым подключенным клиентом

На данной слайде показаны 4 основным таблицы в базе данных это – категории, вопросы, тесты и пользователи, в общем база данных имеем 8 таблиц.

Перейдем к подключению бд.

1. **Слайд(ПодключениеБд**

Подключение к базе данных осуществляется при помощи nu-get библиотеки Firebase.Databasе.

Программный код для подключения выглядит следующим образом, то есть просто вписываем ссылку на саму базу данных.

Все запросы к базе данных осуществляются асинхронно, в данном пример запроса, выводим всю информацию из коллекции Users.

Перейдем к структуре проекта.

1. **Слайд(ПрограммнаяЧасть/Структура)**

Структура проекта имеет следующие папки, Сustom controls - классы нестандартных обьектов, Font – шрифты используемые в интерфейсе, Models – модели данных нужные для работы с базой данных, тоесть написания запросов, Services – классы хранящие в себе запросы к определенным коллекция базы данных.

Views – Графическое представление программы(Страницы и всплывающие окна)

ViewsModel – классы в которых прописаны свойства, команды, методы, коллекции к определенной странице.

Cтруктура приложения основывается на шаблоне проектирование MVVM(MODEL-VIEW-VIEWMODEL), структурированный для разделения логики программы от элементов управления пользовательского интерфейса**.**

1. **Cлайд(ПрограммнаяЧасть/Навигация)**Навигация по приложению осуществляется при помощи оболочки Shell. Shell оболочка нацеленная на упрощение создания мобильных приложений и включает в себя следующий функционал – боковое меню, вкладки, навигацию, поиск.

Основная навигация по приложению осуществляется при помощи нижнего меню в приложении.

1. **Слайд(Функционал/Демонстрация)**

В приложении был реализован следующие функционал: …

И тут уже я просто показываю как это все работает.

1. **Спасибо за внимание**