Здравствуйте уважаемый председатель, члены итоговой аттестационной комиссии и все присутствующие. Вашему вниманию предоставляется моя выпускная квалификационная работа на тему: «Разработка мобильного приложения с возможностью отслеживания количества выполненных задач и учета не выполненных». В рамках данной работы было разработано мобильное приложение.

В современных условиях выбранная тема имеет высокую актуальность, поскольку, используя мобильные устройства мы совсем не привязаны к рабочему месту и без особого труда можем получить доступ к нужным данным в пути, в цеху, на совещании и где–либо еще, потратив на это минимальное количество денег и сил.

Целью выпускной квалификационной работы является разработка мобильного приложения, с помощью программного обеспечения Android Studio, для поддержания стабильности работы сотрудников, путем учета выполненных рабочих поручений.

В предоставленной работе выработан дальнейший алгоритм создания мобильного приложения, который состоит из трех главных шагов, подключающих в себе подэтапы. В целом процесс разработки мобильного приложения состоит из подготовительного шага, в котором продумывается полный цикл разработки приложения и его составляющего контента и соответственно этап разработки, на котором создается мобильное приложение, и проверяется работоспособность.

На данный момент Google прекратила поддержку инструментов для разработки в среде Android IDE. Соответственно было принято решение создавать приложение в среде разработки Android Studio.

ОС Android управляет активностями, опираясь на их состояние. Этот факт облегчает операционной системе определение активности, которые уже не используются и которые можно завершить, что позволит освободить память и ресурсы смартфона для следующих активностей.

Данные состояния можно разделить на группы следующим образом:

*Выполняется или работает*. Активности определяются работающими, если они находятся в самом верху стека активностей.

*Приостановлена*. Когда смартфон находится в состоянии сна или активность еще используется, но частично закрыта новой, не полноценной активностью, то она начинает считаться приостановленной. Приостановленные активности сохраняют информацию о состоянии и используемых элементах, также остаются закрепленными в оконном менеджере. Данное состояние имеет второй по счету приоритет в Android.

*В фоне/остановлена*. Активности считаются остановленными или пребывающими в фоновом режиме если они перекрыты другими. Остановленные активности стараются сохранить данные о своем текущем состоянии и состоянии дочерних компонентов так долго как это возможно, но при этом они имеют самый низкий приоритет и могут быть завершены ОС в любое время для создания лучшей работоспособности активностей, имеющих более высокий приоритет;

*Перезапущена*. Активности, имеющие состояние от приостановлена до остановлена могут быть убраны из оперативной памяти ОС и если юзеру снова необходима данная активность, то она может быть рестартнута. Состояние данной активности будет восстановлено, и она снова будет показана пользователю.

Также имеет место быть усложнение управление жизненным циклом приложения из-за изменения конфигурации.

На основе выявленных требований к мобильному приложению было найдено необходимое количество экранов мобильного приложения. Для каждого экрана определены важные пункты пользовательского интерфейса.

Все пункты графического интерфейса в приложении для Android созданы с использованием объектов View и ViewGroup.

В операционной системе Android хранение данных приложения происходит путем использования баз данных SQLite. Чтобы работать с текущей базой для платформы Xamarin имеется способ через использование библиотеки SQLite.NET, которая в свою очередь взаимодействует с технологией Object–Relational Mapping.

При помощи ORM, включенной в библиотеку SQLite.NET, нет необходимости разработки схем баз данных для простых объектов, благодаря чему значительно ускоряется разработка приложения. Чтобы хранить объекты в базе данных, можно просто описать их параметры, используя пару атрибутов, описанных в самой библиотеке SQLite.NET.

Адаптеры синхронизации – хорошее решение для синхронизации локальных данных с данными сервера, если в приложении уже реализован аутентификатор и провайдеры контента.

Так как для разрабатываемого приложения наибольшим приоритетом обладает регулярность обновления информации, а время жизни устройства на батарее не играет никакой роли (мобильные устройство большую часть времени находятся на зарядке). Для планирования запуска синхронизации данных был выбран класс AlarmManager, обладающий самым предсказуемым временем запуска и не требующий для своей работы наличия дополнительных сервисов.

Для осуществления процесса синхронизации в приложении реализованы два сервиса:

*UpdateMyTryService*, осуществляет синхронизацию списка существующих тренировок;

*UploadMyANDFriendsTryService*; осуществляет синхронизацию общих тренировок.

Оба этих сервиса являются наследниками базового класса IntentService. Класс IntentService автоматически создает очередь намерений, через которые был вызван сервис, что упрощает работу с многопоточностью.

Подводя итог можно сказать о том, что поставленные цели выполнены. К преимуществам данного приложения можно отнести:

* Отсутствие агентов приложения
* Идемпотентность приложения
* Простота и расширяемость
* Понятная архитектура для дальнейшей интеграции сервисов

Но здесь еще можно выделить и недостатки: такие как:

* Низкая надежность
* Отсутствие кроссплатформенности
* Урезанный (хоть и достаточный) начальный функционал

В целом созданное мобильное приложение Effektus готово к применению, однако на данном этапе его создание не закончилось. Ежегодно, а то и чаще, для ОС Android появляется больше новых возможностей. Со временем нужно проводить доработки и добавлять новые возможности.

Также в ближайших планах на будущее есть мысли о том, чтобы изменить полностью графический интерфейс приложения, используя не стандартные методы, предлагаемые Android Studio, а уже свои решения. Так же создать парочку возможностей, которые касаются безопасности. К примеру, возможность корректировать данные в созданных аккаунтах. Есть желание разработать возможность использования пароля не только вводом просто цифр и цифр с буквами, но и с применением графического ключа, отпечатка пальца или faceid. Все эти возможности позволяют приложению обрести новый уровень, новую уникальность, а для пользователя предоставят более совершенную безопасность персональных данных. Перечисленные и многие другие доработки будут разработаны в будущем.

На этом мой доклад окончен. Благодарю за внимание.