

ботки является интегрированная среда разработки (IDE), в виду того, что в нем уже заложена определенная фундаментальная база для работы и создания игр. В качестве удобной и распространенной IDE является Android Studio.

Android Studio – интегрированная среда разработки производства Google, с помощью которой разработчикам становятся доступны инструменты для создания приложений на платформе Android. Android Studio создавалась на базе IDE IntelliJ IDEA. В среде есть такие языки программирования, как Java, Kotlin, C++.

Kotlin – статически типизированный, объектно-ориентированный язык программирования для разработки мобильных приложений. Язык является более лаконичным и типобезопасный, чем Java, и более простым.

Чтобы правильно и рационально расширить функциональные возможности приложения применяются шаблоны проектирования: «MVP». Паттерн MVP (Model-View-Presenter) позволяет отделить логику приложения от визуальной части (представления). Он является архитектурным, то есть он задает общую архитектуру приложения.

Выделяются определённые участники шаблона, MVP состоит из трех компонентов: модели (Model), уровня представления (View) и прослойкой (Presenter).

Model – уровень данных, который общается с базой данных и сетью. View – уровень отображения. Это будет Activity, Fragment или Custom View. Presenter – прослойка между View и Model. View передает происходящие события, а Presenter обрабатывает их, при необходимости обращается к Model и возвращает View данные на отрисовку.

Применительно к Android есть еще одна важная часть – Contract. Это интерфейс, который описывает все взаимодействия между вышеперечисленными компонентами.

А. В. Семенов, Е. А. Левчук
(БГУ, Минск)

Науч. рук. **Е. А. Левчук**, канд. техн. наук, доцент

РАЗРАБОТКА ТРЕБОВАНИЙ И ПРОЕКТИРОВАНИЕ ПРОТОТИПА ПРИЛОЖЕНИЯ

По степени интерактивности виртуальную коммуникацию можно разделить на синхронную и асинхронную. При синхронной коммуни-

кации взаимодействие происходит в режиме реального времени. Асинхронная предполагает возможность получить ответ с задержкой по времени. Обе формы виртуального общения имеют право на существование и выполняют свои специфические функции, однако, смотря на современную жизнь, можно сделать очевидный вывод, что увеличивается именно доля синхронной коммуникации, к которой относятся мессенджеры.

Объектом исследования является кроссплатформенный фреймворк React Native и использование его возможностей для создания мобильного приложения. React Native поддерживает такие платформы как Android, Android TV, iOS, macOS, Apple tvOS, Web, Windows и UWP, позволяя разработчикам использовать возможности библиотеки React вне браузера для создания мобильных приложений, имеющих полный доступ к системным API платформам [1].

Функциональные требования определяют поведение системы: отправку сообщений в личных беседах; отправку голосовых сообщений; создание групповых бесед; отправку изображений; online/offline статус; статус сообщений; ответы на сообщения.

Нефункциональные требования описывают характер поведения системы: простота в использовании для любого пользователя; интуитивно понятный интерфейс; удовлетворение потребностей пользователя.

В ходе проектирования были составлены диаграммы вариантов использования, которые описывают набор действий системы по сотрудничеству с одним или несколькими внешними пользователями системы (участниками). Каждый вариант использования должен обеспечивать некоторый конечный результат для участников или других заинтересованных сторон системы [2].

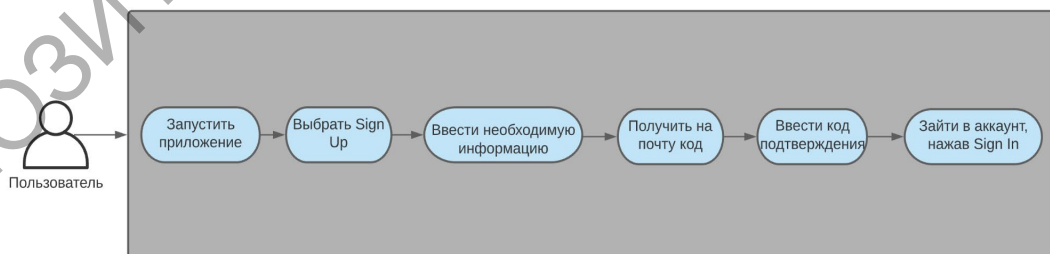


Рисунок 1 – Диаграмма вариантов использования «Восстановление пароля»

Описывается поведение системы с точки зрения пользователя. В результате спроектированы диаграммы для входа, регистрации, написания сообщений, выхода, восстановления паролей (рисунок 1).

Для описания динамических аспектов системы были разработаны диаграммы действий, отражающие алгоритмы последовательности действий, происходящих в системе. Диаграммы позволят определить, как поведет себя пользователь, поставив перед собой некоторую цель.

Ниже приведены примеры диаграммы деятельности для регистрации пользователя и отправки сообщений (рисунки 2 и 3).

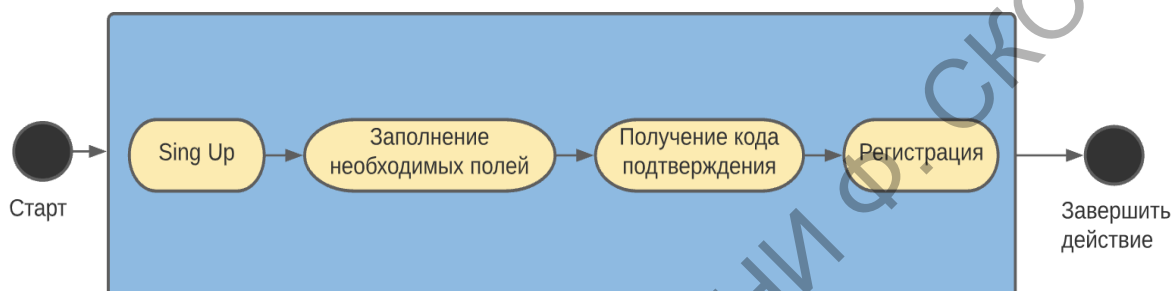


Рисунок 2 – Диаграмма деятельности «Регистрация»

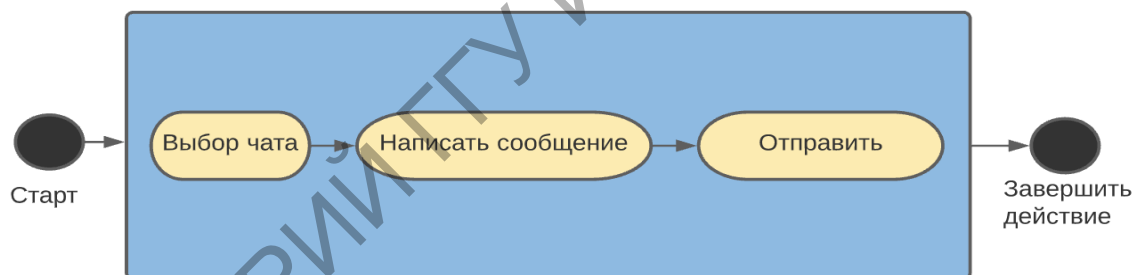


Рисунок 3 – Диаграмма деятельности «Отправка сообщения»

Приложение будет иметь 7 экранов: экран для входа, экран для регистрации, экран для получения кода подтверждения, экран сброса пароля, экран восстановления пароля, экран всех чатов и экран внутри конкретного чата. Примеры прототипов экрана представлены на рисунке 4.

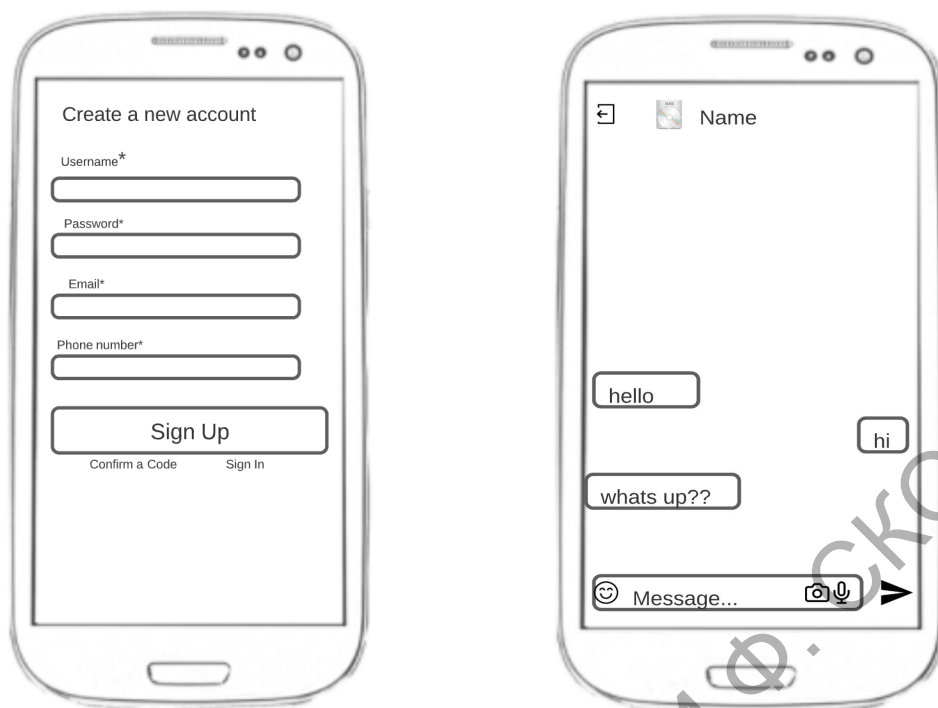


Рисунок 4 – Прототипы экрана «Регистрация» и «Чат»

Литература

1. Documentation for React Native [Electronic resource]. – URL: <https://reactnative.dev>. – Date of access: 29.01.2022.
2. Ларман, К. Применение UML 2.0 и шаблонов проектирования / К. Ларман. Пер. с англ. – М.: Вильямс, 2016. – 736 с.

Т. С. Семенченя

(ГГТУ имени П. О. Сухого, Гомель)

Науч. рук. **К. С. Курочка**, канд. техн. наук, доцент

ПОИСК КТ-ИЗОБРАЖЕНИЙ ПОЯСНИЧНОГО ОТДЕЛА ПОЗВОНОЧНИКА ЧЕЛОВЕКА С ПРИЗНАКАМИ ДЕГЕНЕРАТИВНЫХ ИЗМЕНЕНИЙ

Патологические заболевания и травмы являются одними из наиболее часто встречающихся заболеваний поясничного отдела позвоночника человека. Число заболеваний элементов позвоночника у человека очень велико, и каждый случай носит индивидуальный характер, соответствующий тому или иному виду патологического изменения или травмы [1].