

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Сатпаев университет

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Қаби Айбек Бауыржанұлы

Разработка информационной системы «Поиск банковских карт»

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – «Информационные системы»

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Сатпаев университет

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

КБОиХИ, канд.техн.наук,

ассистент-профессор

_____ Н.А. Сейлова

«__» _____ 2020 г.

ДИПЛОМНАЯ РАБОТА

На тему: Разработка информационной системы «Поиск банковских карт»

Специальность 5В070300 – «Информационные системы»

Выполнил

Қаби А.Б.

Научный руководитель,

К.т.н., ассоц.профессор,

доцент

_____ Б.И.Жумагалиев

«__» _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Сатпаев университет

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой

КБОиХИ, канд.техн.наук,

ассистент-профессор

_____ Н.А. Сейлова

«__» _____ 2020 г.

ЗАДАНИЕ

На выполнение дипломной работы

Обучающемуся Қаби Айбек Бауыржанұлы

Тема: Разработка информационной системы «Поиск банковских карт»

Утверждена приказом Ректора Университета №762-б от 27.01.2020 г.

Срок сдачи законченной работы «25» мая 2020 г.

Исходные данные к дипломной работе: результат литературного обзора современного состояния по данной теме, результаты преддипломной практики.

Краткое содержание дипломной работы:

- а) виды банковских карт;
- б) разработка структуры приложения;
- в) разработка программного обеспечения приложения;
- г) описание приложения и настройка базы данных.

Перечень графического материала (с точным указанием обязательных чертежей): представлены ___ слайдов презентации работы.

Рекомендуемая основная литература: из 8 наименований.

ГРАФИК

подготовки дипломной работы

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Обзор технологий использования банковских карт	10.01.2020 – 08.02.2020	
Разработка базы данных	05.02.2020 – 10.03.2020	
Разработка информационной системы	11.03.2020 – 28.04.2020	

Подписи

Консультантов и нормоконтроллера на законченную дипломную работу с указанием относящихся к ним разделов работы

Наименование разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Разработка ПО	Ассистент М.А. Кабдуллин		
Нормоконтроллер	Тьютор М.Б. Бауыржан		

Научный руководитель _____ Жумагалиев Б.И.

Задание принял к исполнению обучающийся _____ Каби А.Б.

Дата «27» января 2020 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Сатпаев университет

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

Дипломная работа

Қаби Айбек Бауыржанұлы

5B070300 – Информационные системы

Тема: Разработка информационной системы «Поиск банковских карт»

Дипломная работа представляет собой выпускную квалификационную работу по специальности 5B070300 – «Информационные системы». Пояснительная записка состоит из введения, 3 глав, заключения, списка использованных источников и приложения.

Автор дипломной работы поставленные задачи полностью выполнил и показал владение современными технологиями в предметной области.

Дипломная работа выполнена на достаточном профессиональном уровне и содержит все необходимые сведения для такого рода работ. К замечаниям следует отнести незначительные стилистические ошибки.

Считаю, что дипломная соответствует требованиям, предъявляемым к выпускным квалификационным работам по специальности 5B070300 – «Информационные системы». Автор работы Қаби Айбек Бауыржанұлы заслуживает присвоения академической степени бакалавра.

Научный руководитель

Ассоц.проф., канд.техн.наук

Жумагалиев Б.И.

« ___ » _____ 2020 ж.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Каби Айбек

Название: Разработка информационной системы «Поиск банковских карт»

Координатор: Биржан Жумагалиев

Коэффициент подобия 1:15,5

Коэффициент подобия 2:8,8

Замена букв:2

Интервалы:1

Микропробелы:0

Белые знаки:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....
.....

Дата

Подпись заведующего кафедрой /

начальника структурного подразделения

Окончательное решение в отношении допуска к защите, включая обоснование:

.....
.....
.....
.....
.....
.....

Дата

.....
*Подпись заведующего кафедрой /
начальника структурного подразделения*

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Каби Айбек

Название: Разработка информационной системы «Поиск банковских карт»

Координатор: Биржан Жумагалиев

Коэффициент подобия 1: 15,5

Коэффициент подобия 2: 8,8

Замена букв: 2

Интервалы: 1

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....
Дата

.....
Подпись Научного руководителя

DATA RAPORTU: 2020-05-27 07:06:17

TYTUŁ:

Разработка информационной системы «Поиск банковских карт»

AUTOR/ZY:

Каби Айбек

PROMOTOR:

Биржан Жумагалиев

JEDNOSTKA ORGANIZACYJNA:

ИКиИТ

DATA WGRANIA DOKUMENTU:

2020-05-27 07:05:22

LICZBA SPRAWDZEŃ:

1



POMINIĘTE STRONY WWW:



Метрика подобиев

Należy pamiętać, że wysokie wartości Współczynników nie oznaczają automatycznie plagiatu. Raport powinien zostać przeanalizowany przez kompetentną / upoważnioną osobę. Wyniki są uważane za wymagające szczegółowej analizy, jeśli WP 1 wynosi ponad 50%, a WP 2 ponad 5%.



% wyrazów odnalezionych w ramach 5-wyrazowych fraz bez BAP (Baza Aktów Prawnych)

25

Długość frazy dla WP 2



% wyrazów odnalezionych w ramach 25-wyrazowych fraz bez BAP

2757

Liczba słów



% tekstu objęta cudzośłowem

20800

Liczba znaków

Alerty

W tej sekcji znajdują się statystyki występowania w tekście zabiegów edytorskich, które mogą mieć na celu zaburzenie wyników analizy. Niewidoczne dla osoby zapoznającej się z treścią pracy na wydruku lub w pliku, wpływają na frazy porównywane podczas analizy tekstu (poprzez celowe błędy pisowni) w celu ukrycia zapożyczeń lub obniżenia wyników w Raporcie podobieństwa. Należy ocenić, czy zaznaczone wystąpienia wynikają z uzasadnionego formatowania tekstu (nadwrażliwość systemu), czy są celową manipulacją.

Znaki z innego alfabetu	2	pokaż w tekście
liczba znaków z innych alfabetów - mogą imitować litery z alfabetu właściwego dla języka pracy powodując błędy pisowni wyrazów w tekście, należy zweryfikować zasadność użycia		
Rozstrzelenia	1	pokaż w tekście
liczba zastosowań zwiększenia odległości pomiędzy literami - należy sprawdzić czy nie imitują spacji łącząc wyrazy		
Mikrospacje	0	pokaż w tekście
liczba spacji o zerowej długości - należy sprawdzić czy nie wywołują nieprawidłowego podziału wyrazów w tekście		
Białe znaki	0	pokaż w tekście
liczba znaków o białym kolorze czcionki - należy sprawdzić czy nie zastępują spacji powodując złączenie wyrazów (na Raporcie kolor liter jest zmieniany na czarny w celu ich uwidocznienia)		

Aktywne listy podobieństw

Uwagi wymagają szczególnie fragmenty, które zostały włączone do WP 2 (zaznaczone pogrubieniem). Użyj linku "Pokaż w tekście" i zobacz, czy są to krótkie frazy rozproszone w dokumencie (przypadkowe podobieństwa), skupione wokół siebie (parafraza) lub obszerne fragmenty bez wskazania źródła (tzw. "kryptocytaty").

10 najdłuższych fragmentów (11,14 %)

Dziesięć najdłuższych fragmentów znalezionych we wszystkich dostępnych zasobach.

LP	TYTUŁ LUB ADRES URL ŹRÓDŁA (NAZWA BAZY)	AUTOR/ZY	LICZBA IDENTYCZNYCH SŁÓW	
1	https://ru.wikipedia.org/wiki/Kotlin		46	1,67 %
2	https://helpiks.org/5-90400.html		44	1,60 %
3	https://utm.md/wp-content/uploads/2019/06/5_Sectia_BD.pdf		35	1,27 %
4	https://utm.md/wp-content/uploads/2019/06/5_Sectia_BD.pdf		34	1,23 %
5	https://helpiks.org/5-90400.html		33	1,20 %
6	Поисковая система "Qaida" Satbayev University (ИКиИТ)	Елеусизов Талгатбек	26	0,94 %
7	https://www.banki.ru/wikibank/bankovskaya_karta/		25	0,91 %
8	https://helpiks.org/5-90400.html		24	0,87 %
9	https://helpiks.org/5-90400.html		23	0,83 %
10	https://ru.wikipedia.org/wiki/Kotlin		17	0,62 %

z bazy RefBooks (0,00 %)

Wszystkie fragmenty znalezione w bazie RefBooks, która zawiera ponad 3 miliony tekstów od redaktorów i autorów.

LP	TYTUŁ	AUTOR/ZY	LICZBA IDENTYCZNYCH SŁÓW (LICZBA FRAGMENTÓW)
----	-------	----------	--

NIE WYKRYTO ZAPOŻYCZEŃ

z bazy macierzystej (2,29 %)

Wszystkie fragmenty znalezione w bazie danych Twojej instytucji.

LP	TYTUŁ	AUTOR/ZY	DATA INDEKSOWANIA	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	Поисковая система "Qaida" Satbayev University (ИКИУИТ)	Елеусизов Талгатбек	2019-05-06	63 (5)	2,29 %

z Programu Wymiany Baz (0,00 %)

Wszystkie fragmenty znalezione w bazie danych innych instytucji.

LP	TYTUŁ NAZWA BAZY	AUTOR/ZY	DATA INDEKSOWANIA	LICZBA IDENTYCZNYCH SŁÓW (LICZBA FRAGMENTÓW)
----	---------------------	----------	-------------------	--

NIE WYKRYTO ZAPOŻYCZEŃ

z Internetu (13,20 %)

Wszystkie fragmenty znalezione w globalnych zasobach Internetu o otwartym dostępie.

LP	ADRES URL ŹRÓDŁA	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	https://helpiks.org/5-90400.html	124 (4)	4,50 %
2	https://ru.wikipedia.org/wiki/Kotlin	104 (6)	3,77 %
3	https://utm.md/wp-content/uploads/2019/06/5_Sectia_BD.pdf	69 (2)	2,50 %
4	https://www.banki.ru/wikibank/bankovskaya_karta/	33 (2)	1,20 %
5	https://infourok.ru/statya-robototehnika-vo-vneurochnoj-deyatelnosti-4188714.html	15 (1)	0,54 %
6	https://ru.m.wikipedia.org/wiki/Android_Studio	11 (1)	0,40 %
7	https://wiki2.net/NoSQL	8 (1)	0,29 %

АҢДАТПА

Бұл дипломдық жұмыс банктік карталарды іздеу бойынша ақпараттық жүйені жобалауға арналған. Ақпараттық жүйені құру барысында банктік карталарды іздеу мәселесін шешуге арналған мобильдік қосымша әзірленді.

Бұл қосымша банк картасын іздеу міндетін шешетін пайдаланушыларға арналған. Findcard қосымшасы Kotlin мобильдік қосымшаларын жасау ортасында іске асырылған. Деректер қорын басқару жүйесі ретінде Firebase онлайн МББЖ таңдалынды.

Ақпараттық жүйеде ыңғайлы және достық пайдаланушы интерфейсі бар, сондай-ақ банк карталарының барлық түрлерін іздеуге арналған.

АННОТАЦИЯ

Данная дипломная работа посвящена разработке информационной системы по поиску банковских карт. В составе информационной системы разработано мобильное приложение, предназначенное для пользователей, которым необходимо решить проблему по поиску банковских карт.

Данное приложение предназначено для пользователей, которым необходимо решить задачу по поиску банковской карты. Приложение Findcard реализовано в среде разработки мобильных приложений Kotlin. В качестве системы управления базой данных была выбрана онлайн СУБД Firebase.

Информационная система содержит удобный и дружелюбный пользовательский интерфейс, а также предназначено для поиска всех типов банковских карт.

ANNOTATION

This thesis is devoted to the development of an information system for searching Bank cards. As part of the information system, a mobile application has been developed for users who need to solve the problem of searching for bank cards.

This app is intended for users who need to solve the task of searching for a bank card. The Findcard app is implemented in the Kotlin mobile app development environment. The online database management system Firebase was chosen as the database management system.

The information system contains a convenient and friendly user interface, and is designed to search for all types of Bank cards.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	15
1 Обзор технологий использования банковских карт	16
1.1 Банковские услуги	16
1.2 Банковские карты	16
1.3 Постановка задачи	17
2 Разработка мобильного приложения	19
2.1 Требования по функционалу	19
2.2 «Findcard»	19
2.3 Выбор методов	19
2.3.1 Язык программирования Kotlin	20
2.3.2 Android Studio	20
2.3.3 Облачное СУБД Firebase	20
2.4 Диаграмма действий	21
2.5 Прототипирование	22
2.5.1 Проектирование макетов	23
2.5.2 Проектирование базовых страниц приложения	26
3 Описание программного обеспечения	30
3.1 Пользовательский интерфейс приложения	30
3.2 Организация базы данных	38
3.3 Мониторинг посещаемости	40
ЗАКЛЮЧЕНИЕ	42
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	43
Приложение А	44
Приложение Б	45

ВВЕДЕНИЕ

Банковские карты – предмет, который часто используется людьми в повседневной жизни. На сегодняшний день многие люди имеют не одну, а несколько банковских пластиковых карт. Подобного рода карточки являются инструментом, который осуществляет возможность доступа к своему личному счету в банке. Банковские карты используются для безналичных платежей, в том числе и через интернет, и также для снятия наличных или пополнения счета через банкомат или пункт выдачи наличных.

В век технологий людям очень удобно хранить средства в электронном виде, чем передвигаться с наличными купюрами или монетами. Такого рода карты стали неотъемлемой частью жизни многих людей в современном обществе.

Потеря банковских карт является большой проблемой для людей. Казалось бы, что в век инновационных технологий люди должны найти пути решения данной проблемы. Но, на сегодняшний день не нашлось способов ее решения. Решение данной ситуации очень актуально, так как люди тратят большое количество времени на поиск утерянных карт.

Основной задачей является разработка информационной системы, с помощью которой у людей появится возможность искать свои потерянные карты либо находить владельцев найденных карт.

1 Обзор технологий использования банковских карт

1.1 Банковские услуги

Банковская услуга – деятельность банка, осуществляемая в интересах клиента с целью удовлетворения его потребностей в банковском обслуживании. Услуга обычно ассоциируется с конечным результатом.

Услуга – действие, направленное на потребителя, то есть на банковского клиента. Особенность услуг банка является, в отличие от товаров, услуг предприятий, то, что клиент не только принимает услугу банка, но и участвует в процессе ее оказания

Банковская услуга - это форма возможного удовлетворения клиента в кредите, расчетно-кассовом обслуживании, покупке и хранении ценных бумаг, покупке или продаже валюты, технической, технологической помощи.

В условиях, когда банки предоставляют в основном одинаковые, традиционные услуги, качество этих услуг становится важным инструментом конкурентной борьбы банков за клиентов, а также предоставление услуг на новой технологической основе. Качество банковской услуги – совокупность ее свойств, элементов, обеспечивающих удовлетворение уже сформировавшихся и потенциальных потребностей клиентов.

1.2 Банковские карты

Банковская пластиковая карта - именной инструмент для безналичных расчётов, дающий возможность его держателю оплачивать покупку товаров или услуг, или получать наличные деньги в отделениях банков и банкоматах. Банковская карточка – одно из важнейших средств автоматизации расчётных операций в розничном секторе платёжного оборота.

На рисунке 1 показан пример банковской карты где мы можем увидеть магнитную полосу или встроенную микросхему, дающий ключ к специальному карточному счёту в банке. Также на карте имеются данные о сроке действия карты, имени и фамилии владельца, и номере карты.

Это универсальный платёжный инструмент, являющийся ключом доступа к управлению банковским счётом. Карточка позволяет управлять одним или несколькими счетами, выпускается на определённый срок, порядок её использования регламентируется взаимными обязательствами держателя карточки и банка-эмитента.



Рисунок 1 – Пример банковской карточки

Банковские карты делятся на кредитные и дебетовые в зависимости от возможности использования средств сверх остатка на банковском счёте клиента, а также чековые гарантийные карточки.

Кредитная карточка выпускается кредитным учреждением и предполагает наличие у его владельца, помимо счёта в этом учреждении, также возможности получения возобновляемого кредита. Первые небанковские кредитные карточки возникли в США в связи с бурным развитием потребительского кредитования. В 1914 некоторые магазины стали выдавать своим наиболее богатым постоянным клиентам спец. карточки. В последующие годы были придуманы такие элементы финансовой кредитной схемы, как минимальное ежемесячное погашение долга, период отсрочки, беспроцентное кредитование и так далее.

Дебетовая пластиковая карточка привязана к текущему счёту держателя карточки; по ней использование средств возможно только в пределах имеющейся на счёте суммы. Дебетовые карточки не предусматривают автоматического предоставления кредита. Карточки для банковских автоматов – разновидность дебетовых, дают возможность их владельцу получать наличные деньги в пределах остатка средств на своём счёте через специальные автоматические устройства, установленные в банках, магазинах, на вокзалах и т.д. Владелец такой карточки может получить деньги в любое время суток.

1.3 Постановка задачи

Мы живем в эру информационных технологий. В данное время большая часть информации и процессов проходит в цифровом виде. Банки и различные финансовые организации широко применяют это в своей деятельности. Они совершают платежи, переводы и различные выплаты в электронном формате.

Многие организации выплачивают заработную плату своим работникам в электронном виде. В связи с этим у большинства работающих людей имеется одна или более пластиковых банковских карт.

Часто бывают случаи, когда в различных торговых точках выдают сдачу монетами либо мелкими купюрами. Использование пластиковых карточек очень удобно для людей, так как им не приходится носить с собой купюры и монеты.

Нередко бывает такое, что человек теряет свою карточку. Он может оставить ее в магазинах, кафе либо в банкомате. После этого банковскую карту найти бывает проблематично. Для открытия новой карточки нужно посетить банк и внести дополнительные средства. Данный процесс может занять много времени. Альтернативного способа решения на данный момент не существует.

Для решения данной проблемы было принято решение создать информационную систему, при помощи которой появится возможность найти ранее утерянную банковскую карту. Данная информационная система будет реализована в качестве мобильного онлайн приложения. В связи с этим, для реализации приложения был выбран язык программирования Kotlin.

Главной целью данной дипломной работы является - разработка системы, которая могла бы осуществлять возможность людям находить свои потерянные банковские карты.

Для достижения хорошего конечного результата должны быть поставлены определенные задачи. В дипломной работе должны быть решены следующие основные задачи:

- разработка логической схемы информационной системы;
- выбор вида приложения и способа реализации;
- разработка программной части приложения;
- разработка дружественного пользовательского интерфейса.

Приложение должно быть предназначено для практического использования клиентов банков второго уровня.

2 Разработка мобильного приложения

2.1 Требования по функционалу

Данное программное обеспечение должно быть создано как мобильное приложение. ПО должно функционировать на устройствах с операционной системой android. Должна иметься своя БД с функцией добавления и удаления строк.

Данное приложение должно иметь следующие функции:

- поиск карты;
- добавление данных о карте;
- добавление данных о владельце карты;
- добавление данных о нашедшем человеке;
- систему поощрения за каждую найденную карту;
- разделение пользователя на нашедшего и на потерявшего.

2.2 «Findcard»

Для того, чтобы данная система была популярна, она должна быть индивидуальной. Одним из критериев индивидуальности является неповторимое название или имя приложения. В связи с этим было принято решение дать имя данной системе.

«Findcard» - информационная система по поиску банковских карт. Так как, основной задачей системы является поиск потерянных карт, название состоит из двух английских слов «find» и «card», что означают «поиск» и «карта» соответственно.

2.3 Выбор методов

Данная система будет разработана в качестве мобильного приложения. Это связано с тем, что большинство людей используют смартфоны, и им намного удобней пользоваться данным приложением через их мобильные телефоны.

Более 70% всех мобильных приложений написано для операционной системы Android[2]. Этот факт дает нам понять, что данная ОС широко распространена. В связи с этим «Информационная система по поиску банковских карт» будет разработана в качестве Андроид приложения.

Для разработки было решено использовать современный язык программирования Kotlin.

2.3.1 Язык программирования Kotlin

Kotlin (Котлин) — статически типизированный, объектно-ориентированный язык программирования, работающий поверх JavaVirtualMachine и разрабатываемый компанией JetBrains. Также компилируется в JavaScript. Язык назван в честь острова Котлин в Финском заливе, на котором расположен город Кронштадт.

Синтаксис языка использует элементы из Паскаля, PL/SQL, C++, Java, C#. Кроме объектно-ориентированного подхода, Kotlin также поддерживает процедурный стиль с использованием функций. Как в Си и C++ точка входа в программу — функция `main`, принимающая массив параметров командной строки. Программы на Kotlin также поддерживают perl- и shell-стиль интерполяции строк (переменные, включённые в строку, заменяются на их содержимое).

В разработке андроид приложений данный язык появился сравнительно недавно, но уже стал распространённым. Это связано с тем, что его ближайший конкурент Java развивается очень медленно и не имеет многих характеристик, которые имеет в себе Kotlin.

17 мая 2017, на ежегодной конференции, Google анонсировал что начнет поддержку языка программирования Kotlin, который используется в Android Studio, как официального языка программирования для платформы Android наряду с такими языками программирования как Java и C++ [4].

2.3.2 Android Studio

Android Studio — это интегрированная среда разработки для работы с платформами на Android. Данная среда разработки была анонсирована 16 мая 2013 года на конференции Google.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, - официальное средство разработки Android приложений [5]. Данная среда разработки доступна для Windows, OS X и Linux.

2.3.3 Облачное СУБД Firebase

Firebase — американская компания, которая предоставляет облачные услуги. Данная компания была основана в 2011 году Эндрю Ли и Джеймсом Тэмплином, а затем в 2014 году перешла во владения корпорации Google[6].

Основной сервис данной компании — облачная СУБД класса NoSQL, которая позволяет разработчикам приложений синхронизировать и хранить данные между несколькими пользователями. Поддержана способность интеграции с приложениями под операционными системами Android и iOS. Firebase это базы данных в режиме реального времени с которой вы можете

передавать данные непосредственно с клиента. Когда вы сохраняете данные на Firebase, изменения отправляются мгновенно для всех клиентов, веб и мобильных устройств, которые запросили их.

NoSQL (not only SQL) — термин, обозначающий подходы, которые направлены на реализацию СУБД, имеющих существенные отличия от моделей, которые используются в традиционных реляционных СУБД с доступом к данным средствами языка SQL. Применяется к базам данных, в которых делается попытка решить проблемы масштабируемости и доступности за счёт атомарности и согласованности данных.

2.4 Диаграмма действий

Диаграмма действий – UML-диаграмма, на которой показан возможный порядок действий от начала и до конца.

На рисунке 2 показана диаграмма со всевозможными действиями в информационной системе по поиску банковских карт «Findcard».

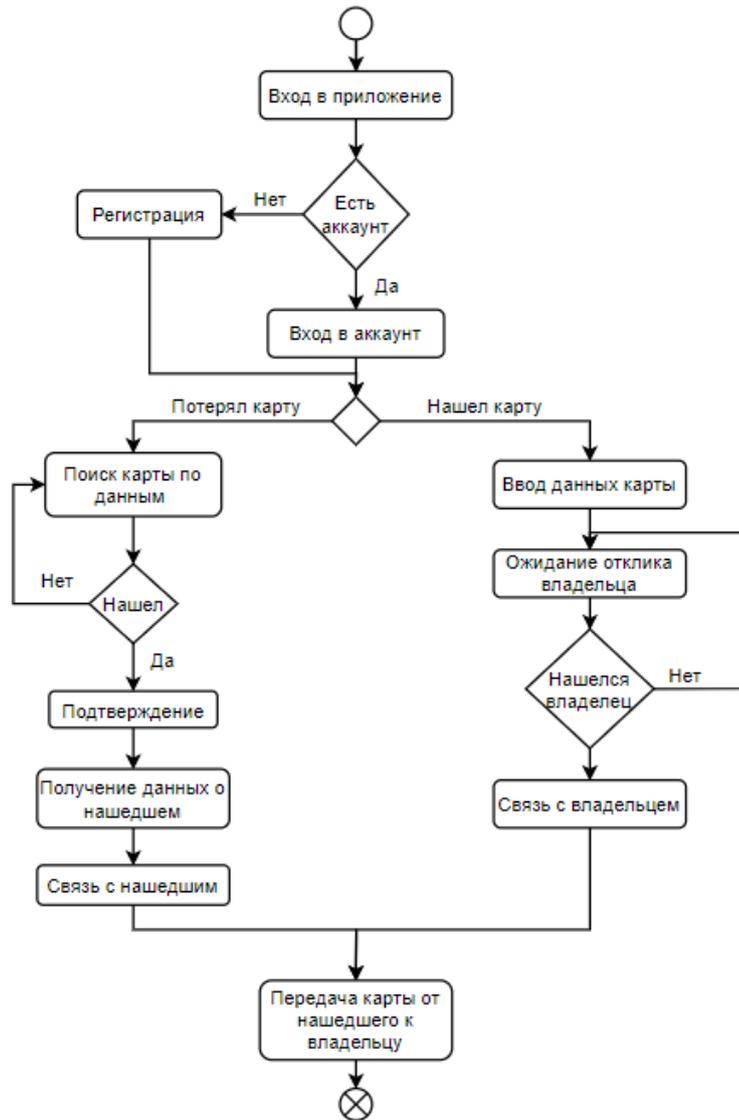


Рисунок 2 – Диаграмма действий

2.5 Прототипирование

Прототипирование — быстрая реализация базовой функциональности для анализа работы всего продукта в целом. На данном этапе малыми усилиями создается функционирующая система. Но, на этапе прототипирования данный продукт может работать с ошибками. На данном этапе будет возможность посмотреть детально устройство системы.

Прототипирование используется в приборостроении и программирование, а также и во многих других сферах технологии. Прототипирование является одним из самых важных этапов разработки приложения.

Прототип – модель или симуляция, представляющая финальный вариант продукта.

Процесс создания прототипа:

1. Определение первоначальных требований к системе.
2. Разработка начального варианта прототипа, который будет содержать в себе лишь пользовательский интерфейс системы.
3. Переработка прототипа учитывая полученные замечания и предложения.

2.5.1 Проектирование макетов

На рисунках 3 – 8 показаны первоначальные макеты страниц приложения. При помощи данных макетов были разработаны главные переходы между страниц. Также, при помощи данных макетов стало понятно какие обязанности и функции будет выполнять та или иная страница.

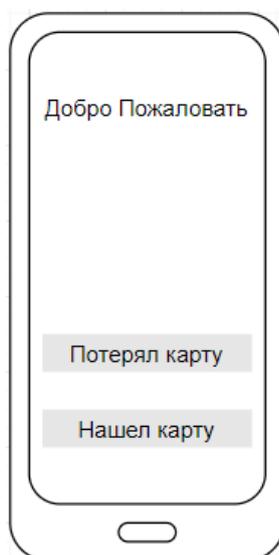


Рисунок 3 – Прототип главной страницы

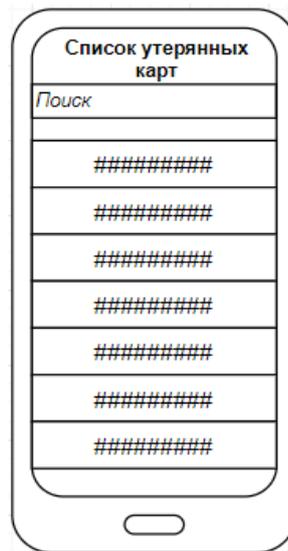


Рисунок 4 – Прототип страницы со списком карт

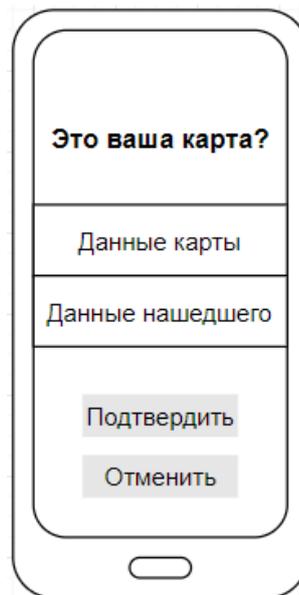


Рисунок 5 – Прототип страницы подтверждения карты



Рисунок 6 – Прототип страницы для выбора типа данных



Рисунок 7 – Прототип страницы для ввода данных о карте

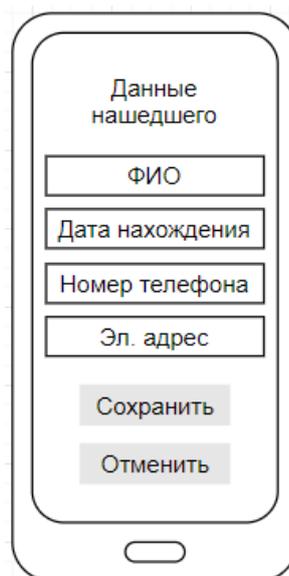


Рисунок 8 – Прототип страницы для ввода данных о нашедшем

2.5.2 Проектирование базовых страниц приложения

На рисунке 9 показана главная страница – первая страница при входе в приложение. На данной странице нужно выбрать свою категорию пользователя. Если вы потеряли карту, то вам нужно нажать на кнопку «Я ищу карту». В том случае если вы нашли карту, то вам нужно нажать на кнопку «Я нашел карту».

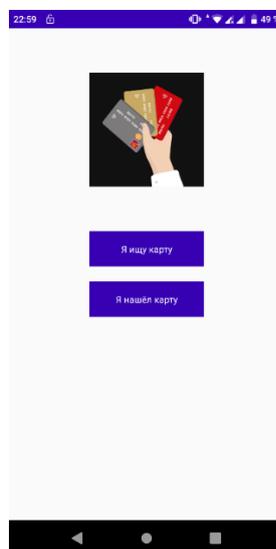


Рисунок 9 – Главная страница

Если вы выберете «Я ищу карту», то вам откроется новая страница со списком - «Найденные карты». Как показано на рисунке 10 в данном списке

имеются данные карт, которые были найдены посторонними людьми. На данной странице можно искать карты по разным критериям.

Как показано на рисунке 11 можно осуществлять поиск по номеру банковской карты. Также, согласно рисункам 12 и 13 можно искать по имени и по фамилии соответственно. Как показано на рисунке 14 если вы находите свою карту, вам предлагают подтвердить данный факт и высвечиваются данные нашедшего.

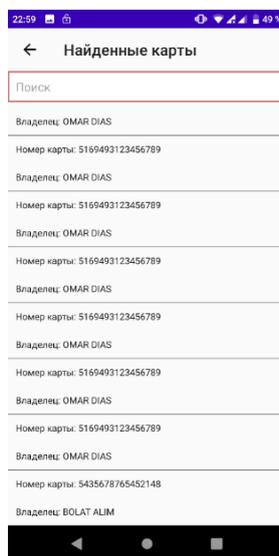


Рисунок 10 – Найденные карты

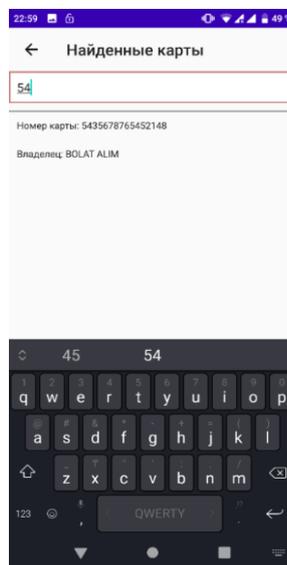


Рисунок 11 – Поиск по номеру карты

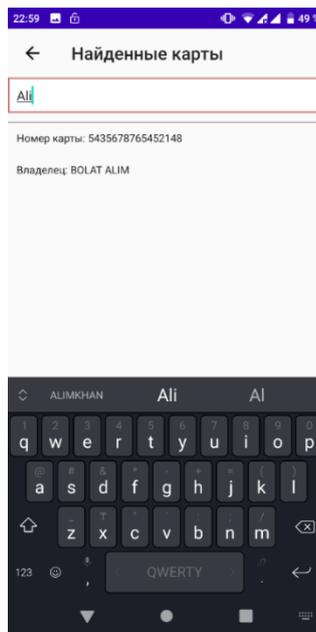


Рисунок 12 – Поиск по имени



Рисунок 13 – Поиск по фамилии

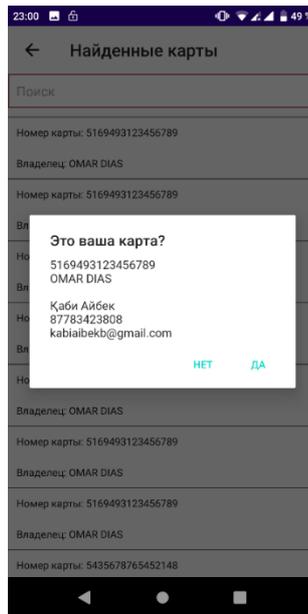


Рисунок 14 – Подтверждение карты

Если вы выберете «Нашел карту», то вам откроется окно «Добавить карту». Согласно рисунку 15 в данном окне вам нужно будет заполнить поля с данными о карте и вашими данными. После заполнения нужно нажать на кнопку «Добавить». После этого данные карты, которые вы ввели можно будет посмотреть в списке найденных карт.

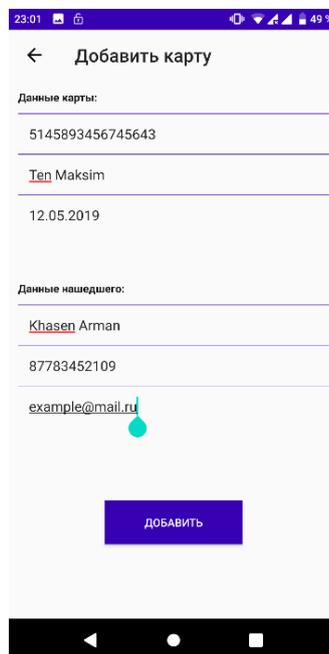


Рисунок 15 – Добавить карту

3 Описание программного обеспечения

3.1 Пользовательский интерфейс приложения

Для проекта «Информационная система по поиску банковских карт» было принято решение использовать графический интерфейс.

Графический интерфейс – вид интерфейса, где человек-пользователь взаимодействует с устройством при помощи различных графических элементов, то есть при помощи списков, таблиц, кнопок и изображений.

Как показано на рисунке 16 при входе в приложение пользователя встречает страница с формой для авторизации, где нужно ввести логин и пароль.

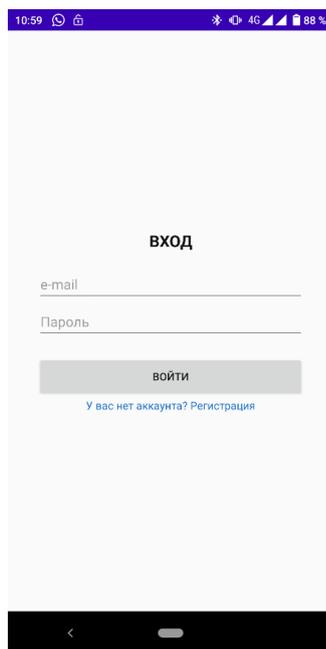


Рисунок 16 – Страница авторизации

В том случае если пользователь не имеет личного кабинета в приложении и желает зарегистрироваться, то он может нажать на «Регистрация» и перейти к форме заполнения данных о пользователе. Как показано на рисунке 17 нужно ввести фамилию и имя, номер телефона, почту и пароль. После этого по нажатию на кнопку «Зарегистрироваться» он переходит в основное меню приложения.

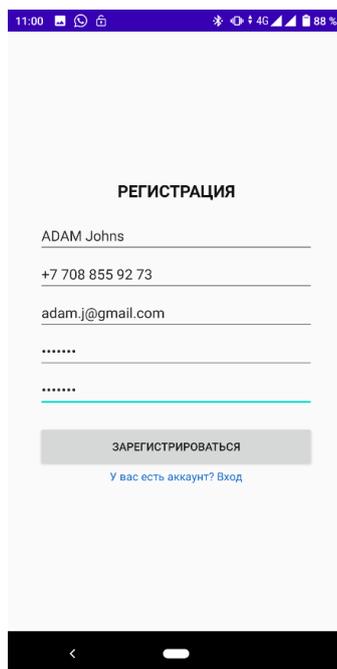


Рисунок 17 – Страница регистрации

Как показано на рисунке 18 в основном меню приложения пользователю дается возможность выбрать дальнейшие действия. Если он нашел карту, то он нажимает на кнопку «добавить карту» и переходит на страницу с формой для заполнения данных карты. Согласно рисунку 19 пользователь должен ввести номер карты, фамилию и имя владельца, и срок действия. После этого данные помещаются в список найденных карт. Если он потерял банковскую карту и желает ее найти, то он должен нажать на кнопку «Я ищу карту», и перейдет на страницу где показан список банковских карт, которые были найдены другими пользователями приложения.



Рисунок 18 – Основное меню приложения

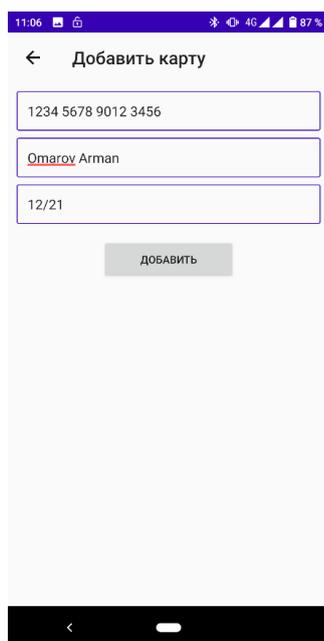


Рисунок 19 – Форма для заполнения данных о карте

Как мы можем видеть на рисунке 20 в списке найденных карт показаны лишь имена и фамилии владельцев и первые и последние цифры банковской карты. Это сделано для того, чтобы мошенники не смогли заполучить карту другого человека. Как показано на рисунках 21 и 22 в таблице можно осуществлять поиск по имени владельца и по номеру банковской карты соответственно.



Рисунок 20 – Список найденных банковских карт

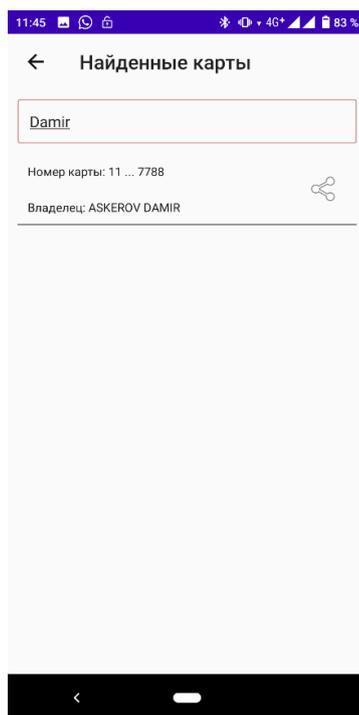


Рисунок 21 – Поиск по имени

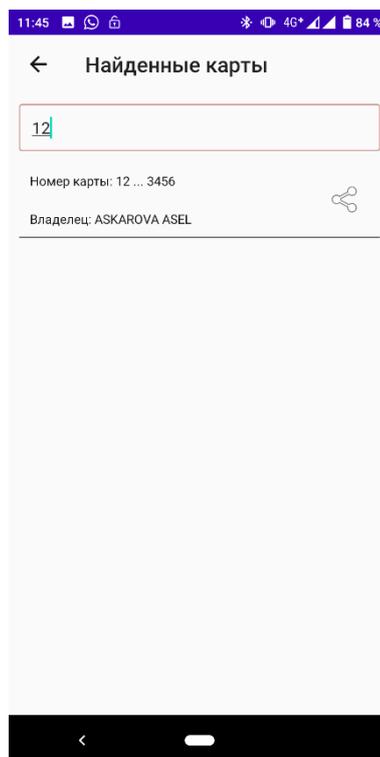


Рисунок 22 – Поиск по номеру карточки

Если пользователь нашел свою утерянную карту, то он может посмотреть контакты нашедшего нажав на данные карты, но перед этим пользователю нужно подтвердить то, что именно он является владельцем данной карты. Как показано на рисунке 23 подтверждение осуществляется при помощи ввода номера банковской карты. Согласно рисунку 24 если пользователь вводит верный номер карты, то он переходит на форму где указаны контактные данные нашедшего и пользователь может отблагодарить нашедшего карты нажав на кнопку «Поблагодарить». Как показано на рисунке 25 нажав на номер телефона, показанный в данной форме, пользователь сразу переходит к звонкам. В соответствии с рисунком 26 в том случае, если пользователь вводит данные карты неправильно, то приложение об этом сообщает.



Рисунок 23 – Подтверждение карты



Рисунок 24 – Контактные данные нашедшего

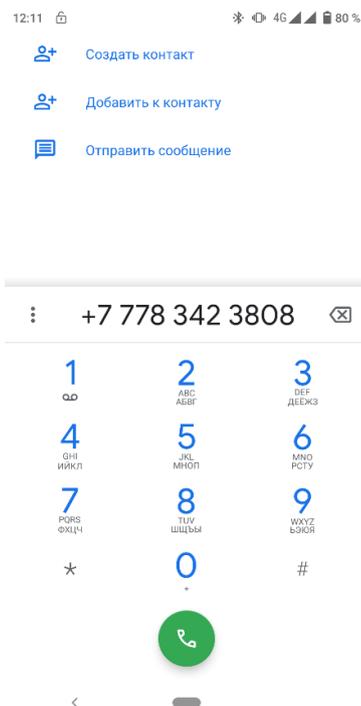


Рисунок 25 – Ссылка с приложение на звонок



Рисунок 26 – Ошибка при вводе данных

В списке, где показаны найденные карты, можно поделиться найденной карты с друзьями при помощи различных мессенджеров и социальных сетей. Для

этого нужно нажать соответствующую кнопку справа стороны списка. Как показано на рисунке 27 после нажатия появится список возможных средств передачи данных.

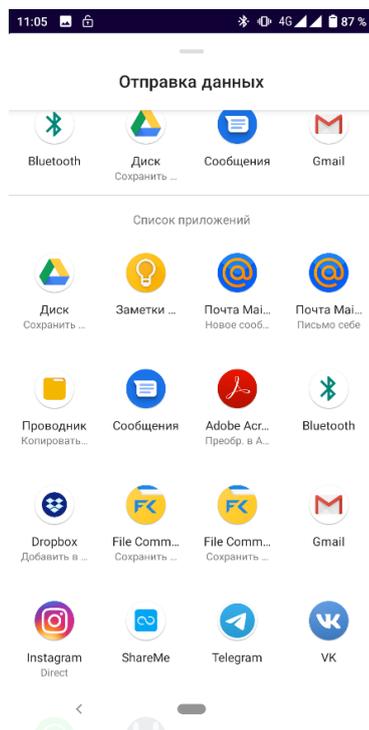


Рисунок 27 – Список средств передачи

К примеру, в соответствии с рисунком 28 если пользователь хочет передать данные при помощи мессенджера WhatsApp, то он выбирает его из списка и отправляет данные.

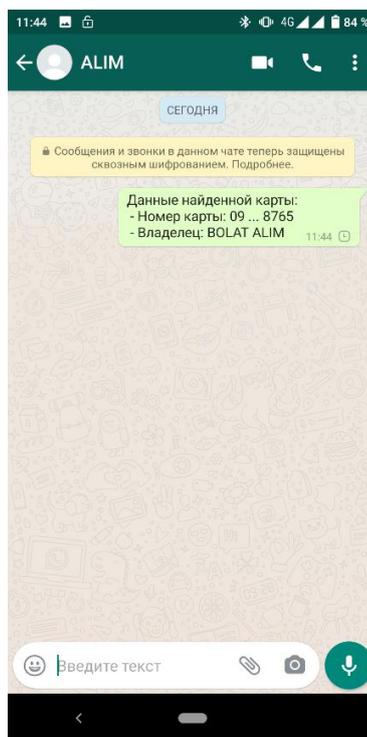


Рисунок 28 – Отправка данных по WhatsApp

3.2 Организация базы данных

Непосредственно база данных расположена на облачной СУБД Firebase, которая находится по адресу – «console.firebase.google.com».

Как показано на рисунке 29 в данной базе данных имеется информация полученная из двух различных форм. Первая – данные о карте и ее владельце (папка cards) и вторая – о том пользователе который обнаружил вашу карту (папка users).

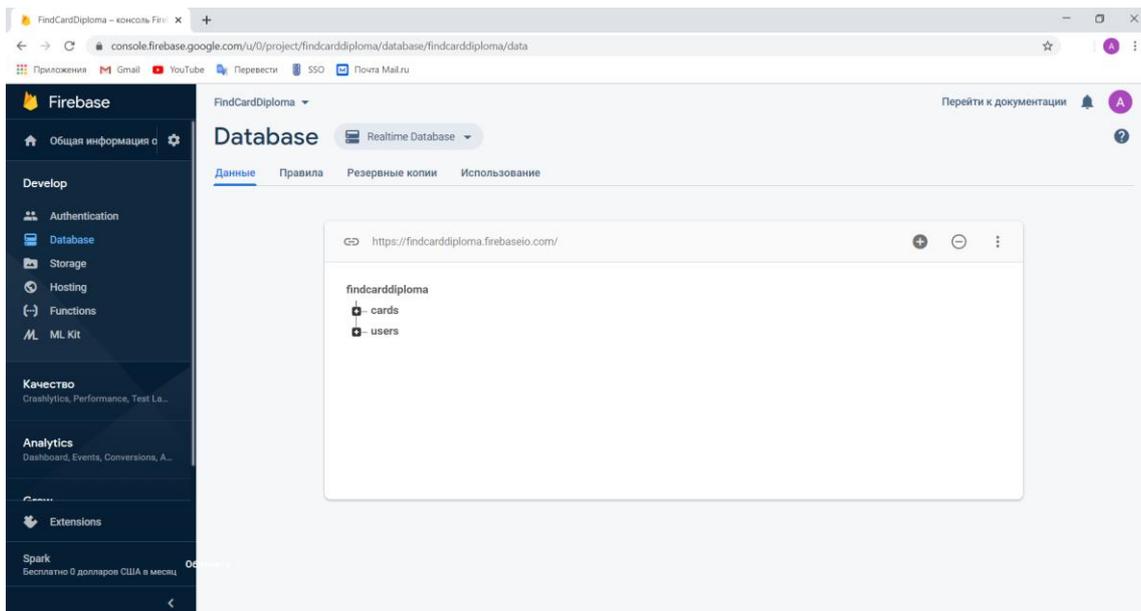


Рисунок 29 – Страница с данными из форм приложения

Согласно рисунку 30 Если перейти в папку «cards», то можно увидеть список добавленных карт.

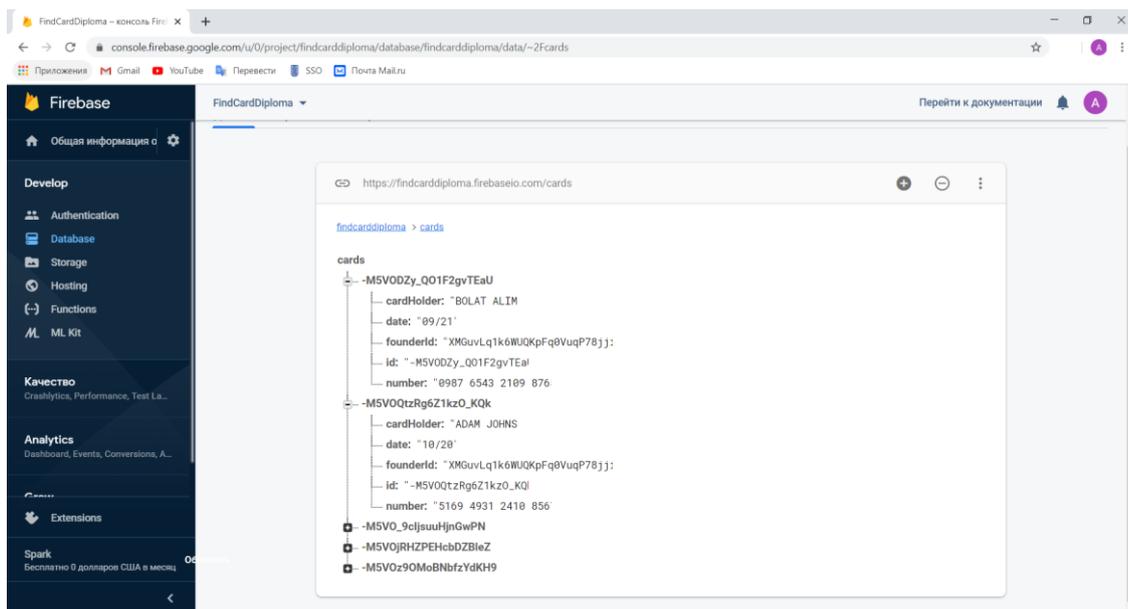


Рисунок 30 – Список добавленных карт

Как показано на рисунке 31 если перейти в папку «users», то можно увидеть список пользователей, которые зарегистрировались в приложении.

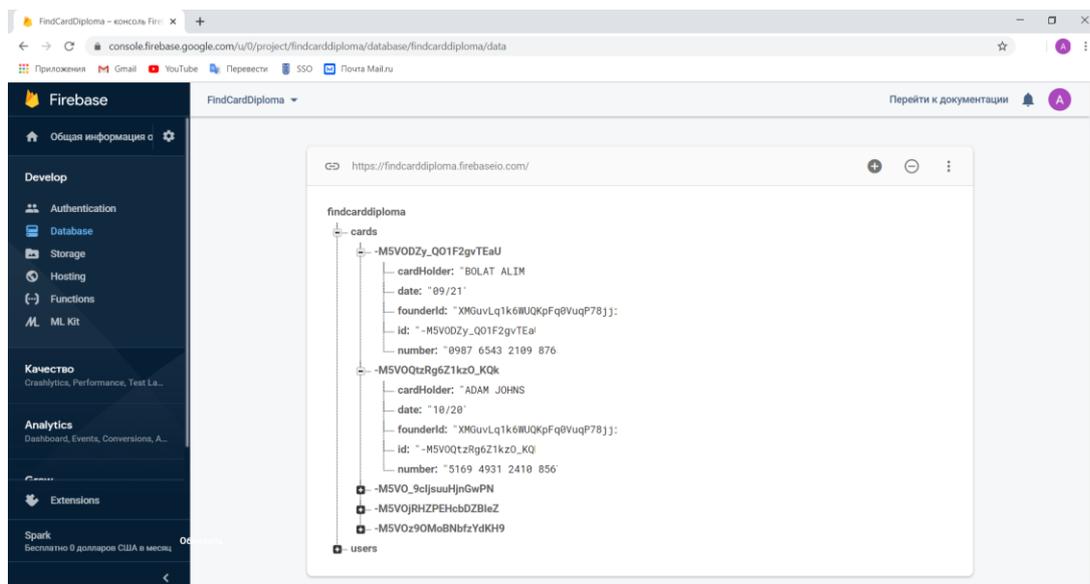


Рисунок 31 – Список пользователей

Данные в папках «cards» и «users» можно изменять напрямую в этих окошках. Также, есть возможность удалять данные о картах и пользователях. Как показано на рисунке 32 в том случае, если вы хотите удалить определенную информацию, то система у вас будет запрашивать подтверждение.

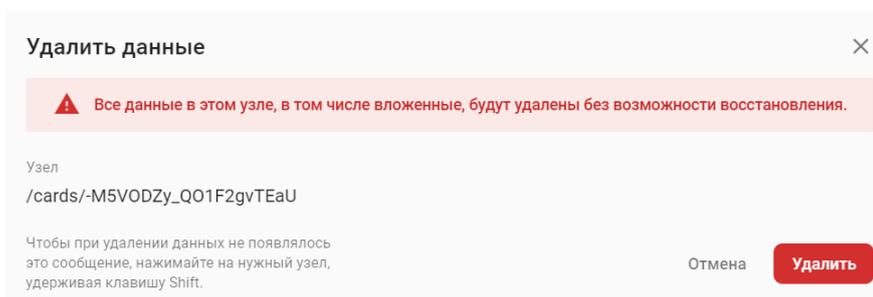


Рисунок 32 – Подтверждения действия

3.3 Мониторинг посещаемости

В СУБД имеется возможность отслеживать активность пользователей. Как показано на рисунке 33 в окне имеются данные о почте, о дате, когда был зарегистрирован пользователь, а также информация о дате с его последнего входа.

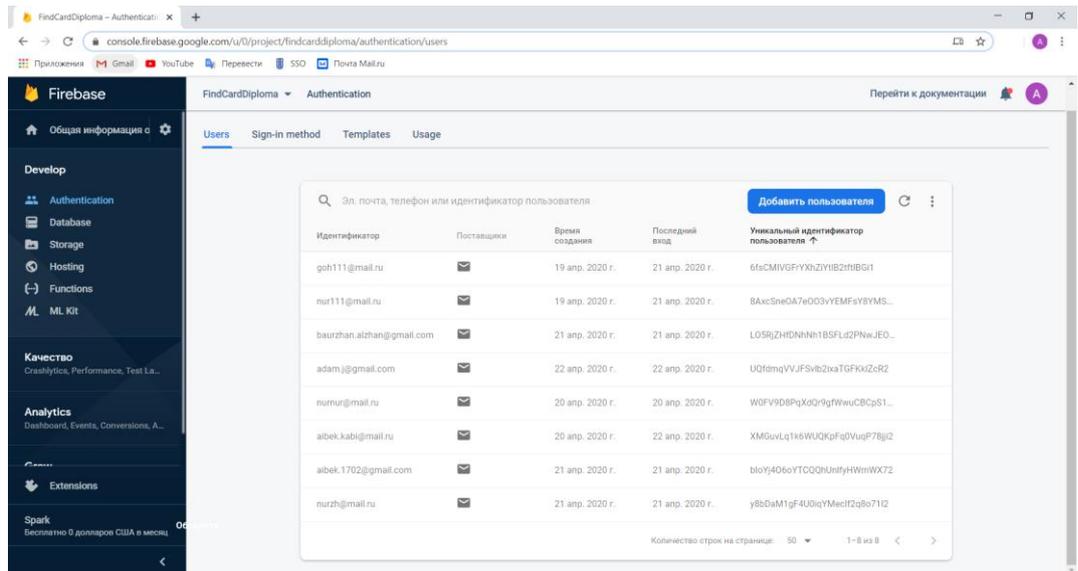


Рисунок 33 – Окно активности пользователей

ЗАКЛЮЧЕНИЕ

В данной дипломной работе разрабатывалась информационная система для поиска потерянных банковских карт

Был разработан алгоритм, который показывал все возможные действия в данной системе. Эти действия в ходе проекта были описаны и показаны на диаграмме действий.

В качестве среды разработки был выбран современный, развивающийся язык программирования Kotlin. Данный язык набирает популярность среди программистов и поддерживается крупными гигантами в сфере IT, следовательно, данный фактор дает понять то, что приложения, написанные на данном языке, будут актуальны и в будущем.

Целью работы являлось создать приложение, которое должно иметь в себе определенные функциональные возможности. Опираясь на требования, которые были заданы данному приложению, была составлена техническая и программная часть системы.

Также, считаю, что основную роль для визуализации данного приложения сыграло прототипирование. На данном этапе был заложен основной каркас, на который делался упор в ходе всего последующего процесса создания системы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Кравцова Г.И. Деньги, кредит, банки. 2010 – 389 с.
- 2 Статистика по использованию платформ [Электронный ресурс]. – Доступ к ресурсу: gs.statcounter.com/os-market-share/mobile/worldwide
- 3 Официальный сайт Kotlin [Электронный ресурс]. – Доступ к ресурсу: <https://kotlinlang.org>
- 4 Paul Miller. Kotlin is an official programming language for Android [Электронный ресурс]. – Доступ к ресурсу: <https://www.theverge.com/2017/5/17/15654988/google-jet-brains-kotlin-programming-language-android-development-io-2017>
- 5 Официальный сайт Android Studio [Электронный ресурс]. – Доступ к ресурсу: developer.android.com
- 6 James Tamplin. Firebase is joining Google [Электронный ресурс]. – Доступ к ресурсу: <https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html>
- 7 Владимир Роман. Облачная СУБД Firebase. 2019 – 299 с.
- 8 Годд Заки Варфел. Прототипирование. Практическое руководство 2013 – 28с.

Приложение А



Рисунок 34 - логотип для иконки приложения

Приложение Б

Листинг программ

Листинг AuthActivity.kt

```
class AuthActivity : AppCompatActivity() {
    private lateinit var auth: FirebaseAuth
    private lateinit var mDatabase: DatabaseReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_auth)
        auth = FirebaseAuth.getInstance()
        mDatabase = FirebaseDatabase.getInstance().getReference("users")
        if (auth.currentUser != null) {
            startActivity(Intent(this, SelectActivity::class.java))
            finish()
        }
        login_btn.setOnClickListener {
            val email = login_email.text.toString()
            val pass = login_password.text.toString()
            loginUser(email, pass)
        }
        signup_btn.setOnClickListener {
            val name = signup_name.text.toString()
            val phone = signup_phone.text.toString()
            val email = signup_email.text.toString()
            val pass = signup_password.text.toString()
            val passConf = signup_password_confirm.text.toString()
            signUpUser(name, phone, email, pass, passConf)
        }
        login_signup.setOnClickListener {
            auth_login.visibility = View.GONE
            auth_signup.visibility = View.VISIBLE
        }
        signup_login.setOnClickListener {
            auth_login.visibility = View.VISIBLE
            auth_signup.visibility = View.GONE
        }
        MaskedTextChangedListener.installOn(signup_phone, "{+7} [000] [000] [00]
[00]")
    }
}
```

Продолжение приложения Б

```
override fun onBackPressed() {
    finish()
}
private fun logInUser(email: String, pass: String) {
    if (email.isNotEmpty() && pass.isNotEmpty()) {
        if (isEmailValid(email)) {
            loader.visibility = View.VISIBLE
            auth.signInWithEmailAndPassword(email, pass).addOnCompleteListener(this)
        } else {
            showToast("Неправильные данные")
        }
        loader.visibility = View.GONE
    } else {
        showToast("Неправильный email")
    }
}
private fun signUpUser(
    name: String,
    phone: String,
    email: String,
    pass: String,
    passConf: String
) {
    if (name.isNotEmpty() && phone.isNotEmpty() && email.isNotEmpty() &&
pass.isNotEmpty() && passConf.isNotEmpty()) {
        if (passConf == pass) {
            if (isEmailValid(email)) {
                loader.visibility = View.VISIBLE
                auth.createUserWithEmailAndPassword(email, pass)
                    .addOnCompleteListener(this) { task ->
                    if (task.isSuccessful) {
                        val userId = mDatabase.push().key
                        userId?.let {
```

Продолжение приложения Б

```
mDatabase.child(email.replace(".", "")).setValue(
    User(auth.currentUser?.uid, name, email, phone, 0
        )
        ).addOnSuccessListener {
startActivity(Intent(this, SelectActivity::class.java))
finish()
        }.addOnFailureListener {
it.printStackTrace()
        } }
        } else {
showToast("Ошибка при регистрации. Повторите попытку")
}
        loader.visibility = View.GONE
        }
        } else {
showToast("Неправильный email")
}
        } else {
showToast("Подтвердите пароль")
}
        } else {
showToast(getString(R.string.fill_all_fields))
} }
private fun showToast(text: String) {
    Toast.makeText(this, text, Toast.LENGTH_SHORT).show()
}
private fun isValidEmail(email: String?): Boolean {
    val expression = "^[\\w\\.\\-]+@[\\w\\.\\-]+\\.([A-Z]{2,4})$"
    val pattern: Pattern = Pattern.compile(expression,
Pattern.CASE_INSENSITIVE)
    val matcher: Matcher = pattern.matcher(email)
    return matcher.matches()
} }
```

Листинг AddCardActivity.kt

```
class AddCardActivity : AppCompatActivity() {
private lateinit var auth: FirebaseAuth
private lateinit var mDatabase: DatabaseReference
```

Продолжение приложения Б

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_add_card)
    setToolbar(add_toolbar)
    auth = FirebaseAuth.getInstance()
    mDatabase = FirebaseDatabase.getInstance().getReference("cards")
        add_button.setOnClickListener {
            val number = add_number_edt.text.trim()
            val cardholder = add_cardholder_edt.text.trim()
            val date = add_date_edt.text.trim()
            if (number.isNotEmpty() && cardholder.isNotEmpty() && date.isNotEmpty())
{
                if (number.length == 19) {
                    if (date.length == 5) {
                        addCard(
                            Card(
                                number = number.toString(),
                                cardHolder = cardholder.toString(),
                                date = date.toString(),
                                founderId = auth.currentUser?.uid
                            ))
                    } else {
                        showToast("Введите дату")
                    }
                } else {
                    showToast("Введите номер карты")
                }
            } else {
                showToast(getString(R.string.fill_all_fields))
            }
        }
        MaskedTextChangedListener.installOn(add_date_edt, "[00]{/}[00]")
        MaskedTextChangedListener.installOn(add_number_edt, "[0000] [0000]
[0000] [0000]")
    }
    override fun onBackPressed() {
        finish()
    }
    private fun addCard(card: Card) {
        loader.visibility = View.VISIBLE
        val cardId = mDatabase.push().key
```

Продолжение приложения Б

```
cardId?.let {
    card.id = it
    mDatabase.child(it).setValue(card).addOnSuccessListener {
        loader.visibility = View.GONE
    }
    finish()
}.addOnFailureListener {
    it.printStackTrace()
    loader.visibility = View.GONE
    showToast(getString(R.string.error_message))
}
private fun showToast(text: String) {
    Toast.makeText(this, text, Toast.LENGTH_SHORT).show()
}
private fun setToolbar(toolbar: Toolbar) {
    toolbar.apply {
        navigationIcon =
        ContextCompat.getDrawable(context,
R.drawable.ic_arrow_back_black_24dp)?.mutate()
        ?.apply {
            val headerColor = ContextCompat.getColor(applicationContext, R.color.black)
            DrawableCompat.setTint(this, headerColor)
        }
    }
    setSupportActionBar(this)
}
supportActionBar?.setDisplayHomeAsUpEnabled(true)
    toolbar.setNavigationOnClickListener { onBackPressed() }
}
override fun finish() {
    super.finish()
    overridePendingTransition(R.anim.from_left_in, R.anim.from_right_out);
}
}}
```

Листинг Card.kt

```
@IgnoreExtraProperties
data class Card (
    var id: String? = null,
    val number: String,
    val cardHolder: String,
```

Продолжение приложения Б

```
var date: String,  
var founderId: String?  
)
```

Листинг CardAdapter.kt

```
class CardAdapter(  
private val context: Context,  
private val onItemClicked: (Card) -> Unit  
) : RecyclerView.Adapter<CardAdapter.CardViewHolder>() {  
private val cardList: List<Card> = emptyList()  
fun setData(list: List<Card>) {  
cardList = list  
notifyDataSetChanged()  
}  
class CardViewHolder(view: View): RecyclerView.ViewHolder(view) {  
private val numberView: TextView =  
itemView.findViewById(R.id.item_card_number)  
private val holderView: TextView =  
itemView.findViewById(R.id.item_card_holder)  
private val shareView: AppCompatImageView =  
itemView.findViewById(R.id.item_share)  
fun bind(context: Context, card: Card) {  
val textForNumber = "Номер карты: " + card.number.substring(0, 2) + " ... " +  
card.number.substring(card.number.length-4, card.number.length)  
numberView.text = textForNumber  
val textForCardHolder = "Владелец: " + card.cardHolder  
holderView.text = textForCardHolder  
shareView.setOnClickListener {  
val sharingIntent = Intent(Intent.ACTION_SEND).apply {  
type = "text/plain"  
putExtra(Intent.EXTRA_SUBJECT, R.string.share)  
putExtra(Intent.EXTRA_TEXT, "Данные найденной карты:\n" +  
" - $textForNumber\n" +  
" - $textForCardHolder\n"  
)  
}  
context.startActivity(Intent.createChooser(sharingIntent,  
context.getString(R.string.share)))
```

Продолжение приложения Б

```
        }}}
    override fun onCreateView(parent: ViewGroup, viewType: Int):
    CardViewHolder {
        val inflater = LayoutInflater.from(parent.context)
        return CardViewHolder(inflater.inflate(R.layout.item_card, parent, false))
    }
    override fun getItemCount(): Int = cardList.size
    override fun onBindViewHolder(holder: CardViewHolder, position: Int) {
        holder.bind(context = context, card = cardList[position])
        holder.itemView.setOnClickListener { onItemClick(cardList[position])
    }
    }
    }
```

Листинг CardListActivity.kt

```
class CardListActivity : AppCompatActivity() {
    private var list: List<Card> = emptyList()
    private lateinit var adapter: CardAdapter
    private lateinit var auth: FirebaseAuth
    private lateinit var mDatabase: DatabaseReference
    private lateinit var confirmationBottomFragment:
    ConfirmationBottomFragment
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_card_list)
        setToolbar(list_toolbar)
        auth = FirebaseAuth.getInstance()
        mDatabase = FirebaseDatabase.getInstance().getReference("cards")
        adapter = CardAdapter(
            context = this,
            onItemClick = {
                if (it.founderId != auth.currentUser?.uid)
                    showAlertDialog(it)
                else
                    deleteCard(it)
            }
        )
        list_search_edt.addTextChangedListener(object : TextWatcher{
            override fun afterTextChanged(s: Editable?) {}
        })
    }
}
```

Продолжение приложения Б

```
override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after:
Int) {}
override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count:
Int) {
    s?.let { filterList(s.trim().toString()) }
        })
        list_recycler.adapter = adapter
    }
    override fun onResume() {
        super.onResume()
        mDatabase.addValueEventListener(listListener)
    }
    override fun onStop() {
        super.onStop()
        mDatabase.removeEventListener(listListener)
    }
    override fun onBackPressed() {
        finish()
    }
    private val listListener = object : ValueEventListener {
        override fun onCancelled(p0: DatabaseError) {
            p0.toException().printStackTrace()
        }
        override fun onDataChange(p0: DataSnapshot) {
            list = if (p0.value != null) {
                val values: HashMap<String, HashMap<String, String>> =
                p0.value as HashMap<String, HashMap<String, String>>
                values.values.map {
                    Card(
                        it.getValue("id"),
                        it.getValue("number"),
                        it.getValue("cardHolder"),
                        it.getValue("date"),
                        it.getValue("founderId")
                    )
                } else {
                    emptyList()
                }
            }
            adapter.setData(list)
            list_loader.visibility = View.GONE
        }
    }
}
```

Продолжение приложения Б

```
if (list.isNullOrEmpty()) {
    list_empty.visibility = View.VISIBLE
} else {
    list_recycler.visibility = View.VISIBLE
}}
private fun deleteCard(card: Card) {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Эту карту добавили вы")
    builder.setMessage("Вы хотите удалить? ")
    builder.setPositiveButton("Да") { dialog, _ ->
        card.id?.let {
            mDatabase.child(it).removeValue()
        }
        dialog.cancel()
    }
    builder.setNegativeButton("Нет") { dialog, _ ->
        dialog.cancel()
    }
    builder.show()
}
@SuppressLint("DefaultLocale")
private fun filterList(key: String) {
    var filtered = list.filter {
        it.cardHolder.toLowerCase().contains(key.toLowerCase())
            || it.number.toLowerCase().substring(0,
2).contains(key.toLowerCase())
            || it.number.toLowerCase().substring(it.number.length-4,
it.number.length).contains(key.toLowerCase())
    }
    if (filtered.isEmpty()) filtered = list
    adapter.setData(filtered)
}
private fun showAlertDialog(card: Card) {
    card.founderId?.let {
        confirmationBottomFragment = ConfirmationBottomFragment.newInstance(it,
card.number)
        confirmationBottomFragment.show(supportFragmentManager,
"CONF_TAG")
    }
}
private fun setToolbar(toolbar: Toolbar) {
```

Продолжение приложения Б

```
        toolbar.apply {
            navigationIcon = ContextCompat.getDrawable(context,
R.drawable.ic_arrow_back_black_24dp)?.mutate()?.apply {
                val headerColor = ContextCompat.getColor(applicationContext, R.color.black)
                DrawableCompat.setTint(this, headerColor)
            }
        }
        setSupportActionBar(this)
    }
    supportActionBar?.setDisplayHomeAsUpEnabled(true)
        toolbar.setNavigationOnClickListener { onBackPressed() }
    }
    override fun finish() {
        super.finish()
        overridePendingTransition(R.anim.from_right_in, R.anim.from_left_out)
    }
}
Листинг ConfirmationBottomFragment.kt
class ConfirmationBottomFragment : BottomSheetDialogFragment() {
    private lateinit var mDatabase: DatabaseReference
    private var id: String? = null
    private var cardNumber: String? = null
    private var user: User? = null
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?
    ): View? {
        id = arguments?.getString(ID)
        cardNumber = arguments?.getString(CARD_NUMBER)
        mDatabase = FirebaseDatabase.getInstance().getReference("users")
        return inflater.inflate(R.layout.layout_confirm_bottom_sheet, container, false)
    }
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        confirmation_check.setOnClickListener {
            val cardNumEdtTxt = confirmation_card_number.text
            if (cardNumEdtTxt.length == 19) {
                if (confirmation_card_number.text.toString() == cardNumber) {
                    loader.visibility = View.VISIBLE
                    mDatabase.addValueEventListener(listener)
                } else {
                    Toast.makeText(context, R.string.it_is_not_years,
Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
}
```

Продолжение приложения Б

```
        }} else {
            Toast.makeText(context, R.string.fill_all_fields,
                Toast.LENGTH_SHORT).show()
        }}
        confirmation_thanks.setOnClickListener {
            userr?.let {
                loader.visibility = View.VISIBLE
                mDatabase.addValueEventListener(listener2)
                mDatabase.child(it.email.replace(".", "")).child("profit").setValue(it.profit + 1)
            }}
            MaskedTextChangedListener.installOn(confirmation_card_number, "[0000]
[0000] [0000] [0000]")
        }
        override fun onStop() {
            super.onStop()
            try {
                mDatabase.removeEventListener(listener)
                mDatabase.removeEventListener(listener2)
            } catch (e: Exception) {
                e.printStackTrace()
            }}
            private val listener = object : ValueEventListener {
                override fun onDataChange(snapshot: DataSnapshot) {
                    snapshot.children.mapNotNull {
                        val post = it.getValue<User>(User::class.java)
                        if (post?.id == id) {
                            userr = post
                            confirmation_name?.text = "${getString(R.string.found_card)}
                            ${post?.name}"
                            confirmation_phone?.text = post?.phone
                            confirmation_email?.text = post?.email
                            confirmation_phone?.setOnClickListener { sendToPhone(post) }
                            confirmation_email?.setOnClickListener { sendToEmail(post) }
                            loader.visibility = View.GONE
                            lay1.visibility = View.GONE
                            lay2.visibility = View.VISIBLE
                        }
                    }
                }
            }
            return
        }
        loader.visibility = View.GONE
    }
}
```

Продолжение приложения Б

```
    Toast.makeText(context, R.string.it_is_not_years,
Toast.LENGTH_SHORT).show()
    }
    override fun onCancelled(err: DatabaseError) {
        loader.visibility = View.GONE
        Toast.makeText(context, R.string.error_message,
Toast.LENGTH_SHORT).show()
        err.toException().printStackTrace()
    }}
    private val listener2 = object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            loader.visibility = View.GONE
            Toast.makeText(context, R.string.thanks_received,
Toast.LENGTH_SHORT).show()
            this@ConfirmationBottomFragment.dismiss()
        }
        override fun onCancelled(err: DatabaseError) {
            loader.visibility = View.GONE
            Toast.makeText(context, R.string.error_message,
Toast.LENGTH_SHORT).show()
            err.toException().printStackTrace()
            this@ConfirmationBottomFragment.dismiss()
        }}
    private fun sendToPhone(post: User?) {
        val intent = Intent(Intent.ACTION_DIAL, Uri.fromParts("tel", post?.phone,
null))
        startActivity(intent)
    }
    private fun sendToEmail(post: User?) {
        val emailIntent = Intent(Intent.ACTION_SENDTO, Uri.fromParts("mailto",
post?.email, null))
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, R.string.email_subject)
        emailIntent.putExtra(Intent.EXTRA_TEXT,
"${getString(R.string.email_text)} $cardNumber")
        startActivity(Intent.createChooser(emailIntent, getString(R.string.write)))
    }
    companion object {
        private const val ID = "ID"
        private const val CARD_NUMBER = "CARD"
        fun newInstance(id: String, number: String) =
```

Продолжение приложения Б

```
ConfirmationBottomFragment().apply {  
arguments = Bundle().apply {  
putString(ID, id)  
putString(CARD_NUMBER, number)  
}}}}}
```

Листинг SelectActivity.kt

```
class SelectActivity : AppCompatActivity() {  
private lateinit var auth: FirebaseAuth  
private lateinit var mDatabase: DatabaseReference  
override fun onCreate(savedInstanceState: Bundle?) {  
super.onCreate(savedInstanceState)  
setContentView(R.layout.activity_select)  
auth = FirebaseAuth.getInstance()  
mDatabase = FirebaseDatabase.getInstance().getReference("users")  
select_found_card.setOnClickListener {  
val intent = Intent(this, AddCardActivity::class.java)  
startActivity(intent)  
overridePendingTransition(R.anim.from_right_in, R.anim.from_left_out);  
}  
select_search_card.setOnClickListener {  
val intent = Intent(this, CardListActivity::class.java)  
startActivity(intent)  
overridePendingTransition(R.anim.from_left_in, R.anim.from_right_out);  
}}  
override fun onResume() {  
super.onResume()  
mDatabase.addValueEventListener(listener)  
select_logout.setOnClickListener {  
mDatabase.removeEventListener(listener)  
FirebaseAuth.getInstance().signOut()  
startActivity(Intent(this, AuthActivity::class.java))  
finish()  
}}  
override fun onStop() {  
super.onStop()  
mDatabase.removeEventListener(listener)  
}  
}
```

Продолжение приложения Б

```
override fun onBackPressed() {
    finish()
}
val listener = object : ValueEventListener {
    override fun onCancelled(p0: DatabaseError) {
        p0.toException().printStackTrace()
        main_loader.visibility = View.GONE
        Toast.makeText(baseContext, R.string.error_message,
            Toast.LENGTH_SHORT).show()
    }
    override fun onDataChange(p0: DataSnapshot) {
        p0.children.mapNotNull {
            val post = it.getValue<User>(User::class.java)
            if (post?.id == auth.currentUser?.uid) {
                main_name.text = "Ваше имя: ${post?.name}"
                main_profit.text = "Ваш баланс: ${post?.profit}"
                main_loader.visibility = View.GONE
                main_profit.visibility = View.VISIBLE
                main_name.visibility = View.VISIBLE
            }
        }
        return
    }
    main_loader.visibility = View.GONE
}}}
```

ЛистингUser.kt

```
@IgnoreExtraProperties
data class User(
    var id: String? = "",
    var name: String = "",
    var email: String = "",
    var phone: String = "",
    var profit: Int = 0
)
```

Листингactivity_add_card.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
```

Продолжение приложения Б

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
tools:context=".AddCardActivity"
android:fillViewport="true"
android:fitsSystemWindows="false"
android:layout_width="match_parent"
android:layout_height="match_parent">
<androidx.constraintlayout.widget.ConstraintLayout
android:focusableInTouchMode="true"
android:layout_width="match_parent"
android:layout_height="wrap_content">
<androidx.appcompat.widget.Toolbar
    android:id="@+id/add_toolbar"
app:title="Добавить карту"
app:layout_constraintTop_toTopOf="parent"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
<LinearLayout
    android:id="@+id/add_fields_card"
android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@id/add_toolbar"
android:paddingStart="10dp"
android:paddingEnd="10dp"
android:layout_marginTop="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content">
<EditText
    android:id="@+id/add_number_edt"
android:hint="0000 0000 0000 0000"
android:lines="1"
android:digits="0123456789 "
android:layout_marginBottom="8dp"
android:inputType="numberDecimal"
    android:background="@drawable/edittext_background"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/add_cardholder_edt"
android:hint="Владелец карты"
android:lines="1"
```

Продолжение приложения Б

```
android:inputType="text"
android:textAllCaps="true"
android:maxLength="30"
android:layout_marginBottom="8dp"
    android:background="@drawable/edittext_background"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/add_date_edt"
android:hint="MM/ГГ"
android:lines="1"
android:digits="0123456789/"
android:inputType="numberDecimal"
    android:background="@drawable/edittext_background"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>
<Button
    android:id="@+id/add_button"
android:layout_width="150dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/add_fields_card"
app:layout_constraintEnd_toEndOf="parent"
android:text="Добавить"
    />
<LinearLayout
    android:id="@+id/loader"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#4B827F7F"
android:visibility="gone"
android:clickable="true"
android:focusable="true"
android:gravity="center">
<ProgressBar
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    />
```

Продолжение приложения Б

```
</LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>
```

Листинг activity_auth.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:focusableInTouchMode="true"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <LinearLayout  
        android:id="@+id/auth_login"  
        android:visibility="visible"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:gravity="center"  
        android:layout_margin="32dp"  
        android:orientation="vertical">  
        <TextView  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="Вход"  
            android:textAllCaps="true"  
            android:textAlignment="center"  
            android:textStyle="bold"  
            android:textColor="#111"  
            android:textSize="21sp"  
            android:layout_marginBottom="16dp"  
        />  
        <EditText  
            android:id="@+id/login_email"  
            android:inputType="textEmailAddress"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:lines="1"  
            android:hint="e-mail"
```

Продолжение приложения Б

```
    />
<EditText
    android:id="@+id/login_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:lines="1"
    android:hint="Пароль"
    />
<Button
    android:id="@+id/login_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Войти"
    android:layout_marginTop="16dp"
    />
<TextView
    android:id="@+id/login_signup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textColor="#156BC8"
    android:text="У вас нет аккаунта? Регистрация"/>
</LinearLayout>
<LinearLayout
    android:id="@+id/auth_signup"
    android:visibility="gone"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:layout_margin="32dp"
    android:orientation="vertical">
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Регистрация"
    android:textStyle="bold"
    android:textColor="#111"
    android:textSize="21sp"
    android:textAlignment="center"
```

Продолжение приложения Б

```
android:textAllCaps="true"
android:layout_marginBottom="16dp"
    />
<EditText
    android:id="@+id/signup_name"
    android:inputType="textEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Имя"
    android:maxLength="20"
    android:lines="1"
    />
<EditText
    android:id="@+id/signup_phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:lines="1"
    android:hint="Номер телефона"
    />
<EditText
    android:id="@+id/signup_email"
    android:inputType="textEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="e-mail"
    android:lines="1"
    />
<EditText
    android:id="@+id/signup_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:hint="Пароль"
    android:lines="1"
    />
<EditText
    android:id="@+id/signup_password_confirm"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
```

Продолжение приложения Б

```
android:hint="Подтвердите пароль"
android:lines="1"
    />
<Button
    android:id="@+id/signup_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Зарегистрироваться"
    />
<TextView
    android:id="@+id/signup_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="#156BC8"
    android:gravity="center"
    android:text="У вас есть аккаунт? Вход"/>
</LinearLayout>
<LinearLayout
    android:id="@+id/loader"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4B827F7F"
    android:visibility="gone"
    android:clickable="true"
    android:focusable="true"
    android:gravity="center">
    <ProgressBar
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        />
</LinearLayout>
</RelativeLayout>
```

Листинг activity_card_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

Продолжение приложения Б

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:focusableInTouchMode="true"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".CardListActivity">
<androidx.appcompat.widget.Toolbar
    android:id="@+id/list_toolbar"
    app:title="Найденные карты"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/list_search_edt"
    android:hint="Поиск"
    android:layout_margin="10dp"
    android:background="@drawable/search_background"
    app:layout_constraintTop_toBottomOf="@id/list_toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<TextView
    android:id="@+id/list_empty"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:textStyle="bold"
    android:visibility="gone"
    app:layout_constraintTop_toBottomOf="@id/list_search_edt"
    android:gravity="center"
    app:layout_constraintBottom_toBottomOf="parent"
    android:textAllCaps="true"
    android:textSize="21sp"
    android:text="Пусто" />
<ProgressBar
    android:id="@+id/list_loader"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/list_search_edt"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

Продолжение приложения Б

```
app:layout_constraintBottom_toBottomOf="parent"
    />
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/list_recycler"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:visibility="gone"
    android:layout_marginTop="10dp"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    tools:listitem="@layout/item_card"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    app:layout_constraintTop_toBottomOf="@id/list_search_edt"
    app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Листинг activity_select.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SelectActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        android:orientation="vertical">
        <Button
            android:id="@+id/select_search_card"
            android:text="Я ищу карту"
            android:textAlignment="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
```

Продолжение приложения Б

```
<Button
    android:id="@+id/select_found_card"
    android:text="Я нашёл карту"
    android:textAlignment="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/select_logout"
    android:text="Выход"
    android:textAlignment="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:gravity="center"
    android:orientation="vertical">
<ProgressBar
    android:id="@+id/main_loader"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
<TextView
    android:id="@+id/main_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone"
    android:text="Ваше имя: "/>
<TextView
    android:id="@+id/main_profit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone"
    android:text="Ваш баланс: "/>
</LinearLayout>
```

Продолжение приложения Б

Листинг item_card.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/selectableItemBackground"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto">
<View
    android:id="@+id/divider"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    app:layout_constraintBottom_toBottomOf="parent"
    android:background="#a111"/>
<TextView
    android:id="@+id/item_card_number"
    tools:text="Номер карты: 5143 **** * 3474"
    android:textColor="#111"
    android:padding="10dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<TextView
    android:id="@+id/item_card_holder"
    tools:text="Владелец: ANDREY VOROBEL"
    android:textColor="#111"
    android:padding="10dp"
    app:layout_constraintTop_toBottomOf="@id/item_card_number"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<androidx.appcompat.widget.AppCompatImageView
    android:id="@+id/item_share"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:src="@android:drawable/ic_menu_share"
```

Продолжение приложения Б

```
android:layout_marginEnd="16dp"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Листинг layout_confirm_bottom_sheet.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:gravity="center"
        android:layout_alignParentTop="true"
        android:text="Подтвердите, что это ваша карта"
        android:textColor="#000"
        android:layout_marginTop="16dp"
        android:textSize="21sp"
        android:textStyle="bold" />
    <LinearLayout
        android:id="@+id/lay1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_below="@id/title"
        android:padding="16dp"
        android:visibility="visible">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Введите номер вашей карты, чтобы узнать контакты
человека, нашедшего эту карту:"
            android:textColor="#000" />
```

Продолжение приложения Б

```
<EditText
    android:id="@+id/confirmation_card_number"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="0000 0000 0000 0000" />
<Button
    android:id="@+id/confirmation_check"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Проверить" />
</LinearLayout>
<LinearLayout
    android:id="@+id/lay2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp"
    tools:visibility="visible"
    android:layout_below="@id/title"
    android:visibility="gone">
<TextView
    android:id="@+id/confirmation_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:text="Вашу карту нашёл: " />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:layout_marginTop="10dp"
    android:text="Номер телефона: " />
<TextView
    android:id="@+id/confirmation_phone"
    android:layout_width="0dp"
    android:layout_weight="1"
```

Продолжение приложения Б

```
android:layout_height="wrap_content"
android:textSize="16sp"
android:textColor="#0551AE"
android:layout_marginTop="10dp"/>
</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textSize="16sp"
android:layout_marginTop="10dp"
android:text="Почта: " />
<TextView
    android:id="@+id/confirmation_email"
android:layout_width="0dp"
android:layout_weight="1"
android:layout_height="wrap_content"
android:textSize="16sp"
android:textColor="#C13737"
android:layout_marginTop="10dp"/>
</LinearLayout>
<Button
    android:id="@+id/confirmation_thanks"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="48dp"
android:text="Поблагодарить !" />
</LinearLayout>
<LinearLayout
    android:id="@+id/loader"
android:layout_width="match_parent"
android:layout_height="250dp"
android:background="#4B827F7F"
android:visibility="gone"
android:clickable="true"
android:focusable="true"
android:gravity="center">
<ProgressBar
```

Продолжение приложения Б

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
    />  
</LinearLayout>  
</RelativeLayout>
```