

**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:** Темирханов Дидар Диясович

**Название:** Разработка мобильного приложения – «Приемная комиссия»

**Координатор:** Жибек Алибиева

**Коэффициент подобия 1:**16,2

**Коэффициент подобия 2:**12,1

**Замена букв:**0

**Интервалы:**0

**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*

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

**ОТЗЫВ**

**НАУЧНОГО РУКОВОДИТЕЛЯ**

на \_\_\_\_\_ дипломный проект \_\_\_\_\_

\_\_\_\_\_ Темирханов Дидар Диясович \_\_\_\_\_

\_\_\_\_\_ 5B070400 – Вычислительная техника и программное обеспечение \_\_\_\_\_

Тема: « Разработка мобильного приложения для приёмной комиссии на платформах iOS, Android».

Данная дипломная работа нацелена на продвижение и представление деятельности приёмной комиссии университета в мобильной среде, а так же облегчит подачу документов, консультирование абитуриентов.

Дипломный проект состоит из введения, 4-х основных разделов и заключения:

В введении описывается актуальность выбранной темы, определяются основные задачи и объект исследования.

В исследовательском разделе подробно описана предметная область, разбор аналогичного продукта.

В технологическом разделе дается обоснование выбора технологии и их описание с подробной аргументацией выбора.

В проектном разделе приведена архитектура мобильного приложения, диаграмма деятельности, конфигурация базы данных.

Последний раздел , экспериментальный является основным в нём изложен ход работы по разработке интерфейса и программной части продукта.

Дипломная работа оформлена согласно стандарту ГОСТ, пройдена на антиплагиат и рекомендована к защите. Темирханов Дидар заслуживает присвоения академической степени «бакалавра» по специальности 5B070400 – «Вычислительная техника и программное обеспечение»

Научный руководитель:

Лектор, Магистр технических наук \_\_\_\_\_ Кайрбеков А.М.

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

Темирханов Д.Д

Разработка мобильного приложения «Приемная комиссия»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное  
обеспечение

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой ПИ

канд.техн.наук, доцент,

ассоциированный профессор

Р. Юнусов

" \_\_\_\_\_ " \_\_\_\_\_ 2020 г.

### **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: "Разработка мобильного приложения для приёмной комиссии"

по специальности 5B070400 – Вычислительная техника и программное  
обеспечение

Выполнил

Темирханов Д.Д.

Научный руководитель  
мастер технических наук

\_\_\_\_\_ Каирбеков А.М

" \_\_\_\_\_ " \_\_\_\_\_ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

**5B070400 – Вычислительная техника и программное обеспечение**

**УТВЕРЖДАЮ**

Заведующий кафедрой ПИ  
канд.техн.наук, доцент,  
ассоциированный профессор  
\_\_\_\_\_ Р. Юнусов  
" \_\_\_\_ " \_\_\_\_\_ 2020 г.

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Обучающемуся Темирханову Дидару Диясовичу

Тема: Разработка мобильного приложения для приёмной комиссии

Утверждена приказом проректора по академической работе № \_\_\_\_\_  
от " \_\_\_\_ " \_\_\_\_\_ 2020 г.

Срок сдачи законченного проекта " 17 " мая 2020 г.

Исходные данные к дипломному проекту: Техническое задание, описание необходимых функций приложения.

Перечень подлежащих разработке в дипломном проекте вопросов:

- а) проектирование базы данных Firebase согласно требованиям проекта;
- б) реализация архитектуры мобильного приложения в соответствии с концепцией Vanilla;
- в) проектирование и разработка пользовательского интерфейса;
- д) разработка, отладка, тестирование программного комплекса.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 32 слайда презентации.

Рекомендуемая основная литература: из 5 наименований.

## ГРАФИК

подготовки дипломного проекта

<b>Наименование разделов, перечень разрабатываемых вопросов</b>	<b>Сроки представления научному руководителю и консультантам</b>	<b>Примечание</b>
1. Анализ предметной области, разработка технического задания на проектирование программного комплекса		
2. Реализация регистрации данных и загрузки документов		
3. Разработка пользовательского интерфейса		
4. Разработка структуры приложения с реализацией логической цепочки иерархических вызовов		
5. Тестирование приложения		
6. Написание пояснительной записки к дипломному проекту		

### Подписи

консультантов и нормоконтролера на законченный дипломный проект  
с указанием относящихся к ним разделов проекта

<b>Наименования разделов</b>	<b>Консультанты, И.О.Ф. (уч. степень, звание)</b>	<b>Дата подписания</b>	<b>Подпись</b>
Программное обеспечение	Каирбеков А.М. мастер техн. наук, лектор		
Нормоконтролер	Алибиева Ж.М. магистр техн. наук, лектор		

Научный руководитель \_\_\_\_\_ Каирбеков А.М

Задание принял к исполнению обучающийся \_\_\_\_\_ Темирханов Д.Д

Дата " \_\_ " \_\_\_\_\_ 2020г.

## АННОТАЦИЯ

Дипломная работа нацелена на продвижение и представление деятельности приёмной комиссии университета в мобильной среде.

Пояснительная записка состоит из введения, четырех основных разделов и заключения:

В введении описывается актуальность выбранной темы, определяются основные задачи и объект исследования.

В исследовательском разделе подробно описана предметная область, разбор аналогичного продукта.

В технологическом разделе дается обоснование выбора технологии и их описание с подробной аргументацией выбора.

В проектном разделе приведена архитектура мобильного приложения, диаграмма деятельности, конфигурация базы данных.

Последний раздел, экспериментальный является основным в нём изложен ход работы по разработке интерфейса и программной части продукта.

Записка "мобильное приложение для приёмной комиссии" состоит из 31 страниц. К написанию работы использовались 12 литературных статей.



## ANNOTATION

The thesis is aimed at promoting and presenting the activities of the university selection committee in a mobile environment.

The explanatory note consists of an introduction, four main sections and a conclusion:

The introduction describes the relevance of the chosen topic, defines the main tasks and the object of study.

The research section describes in detail the subject area, the analysis of a similar product.

The technological section provides the rationale for the choice of technology and their description with detailed argumentation of the choice.

The design section shows the architecture of the mobile application, activity diagram, database configuration.

The last section, the experimental one, is the main one; it outlines the work on developing the interface and the software part of the product.

Note "mobile application for the selection committee" consists of 31 pages. To write the work used 12 literary articles.

## АНДАТПА

Дипломдық жұмыс университеттің қабылдау комиссиясының қызметін, мобильді ортада насихаттауға және ұсынуға бағытталған.

Дипломдық жоба кіріспеден, негізгі төрт бөлімнен және қорытындыдан тұрады:

Кіріспе таңдалған тақырыптың өзектілігін сипаттайды, негізгі міндеттер мен зерттеу объектісін анықтайды.

Зерттеу бөлімінде тақырыптың егжей-тегжейлі сипаттамасы, ұқсас өнімнің талдауы берілген.

Технологиялық бөлім таңдауды толық дәлелдей отырып, технологияны таңдау негіздемесін және олардың сипаттамасын ұсынады.

Жоба бөлімінде мобильді қосымшаның архитектурасы, қызмет диаграммасы, мәліметтер базасының конфигурациясы көрсетілген.

Соңғы бөлім, эксперименталды, негізгі болып табылады, онда интерфейсті және өнімнің бағдарламалық бөлігін әзірлеу жұмыстары көрсетілген.

Ескерту «қабылдау комиссиясына арналған мобильді қосымша» 31 беттен тұрады. Жұмыс барысында 12 әдеби мақала пайдаланылды.

## СОДЕРЖАНИЕ

Введение	9
1 Исследовательский раздел	10
1.1 Цель разработки приложения	11
1.2 Термины и сокращения	11
1.3 Обзор аналогичных приложений	12
1.3.1 Мобильное приложение «Хочу в ТГУ»	12
2 Технологический раздел	13
2.1 Обзор используемых инструментов	13
2.1.1 Flutter SDK	13
2.1.2 Язык программирования Dart	14
2.1.3 Android Studio	15
2.1.4 AVD Manager	15
3 Проектный раздел	16
3.1 Архитектура системы	16
3.2 Описание диаграммы деятельности	17
3.3 Проектирование базы данных	18
4 Экспериментальный раздел	20
4.1 Создание пользовательского интерфейса	20
4.2 Сбор и запись данных в БД	23
4.3 Асинхронная загрузка файлов в хранилище	25
Заключение	28
Список использованной литературы	29
Приложение А. Техническое задание	30

## ВВЕДЕНИЕ

Современная деловая среда в сфере образования немыслима без эффективного управления и обработки информации. Одним из ключевых показателей успешности высших учебных заведений, заключается в том, что информационные технологии являются неотъемлемой частью при выполнении разных приоритетных задач.

Целью данного дипломного проекта является повышение эффективности и упрощение работы приёмной комиссии университета, посредством разработки мобильного приложения. С помощью приложения, ожидаемым результатом станет оперативность и качество в работе с абитуриентами.

Основными задачами приёмной комиссии являются:

- организация и обеспечение приёма документов поступающих лиц;
- слежение за доступностью информации для абитуриентов в период приёмной кампании;
- обеспечивает строгое соблюдение всех требований законодательства РК по набору и зачислению;

По результатам прошлогодних наблюдений, в период приёмной кампании были замечены очереди к окошкам по приёму документов и заявлениям.

Потерянное время в очередях, скопление абитуриентов не приносит за собой никакой пользы имиджу, эффективности в работе приёмной комиссии. Удобное в использовании мобильное приложение избавит абитуриентов от необходимости стоять в очередях, заполнять бумажный бланк на местах. Члены комиссии будут иметь возможность получать заявления и другие документы, а так же основные данные абитуриента.

## 1 Исследовательский раздел

### 1.1 Цель разработки приложения

Данная дипломная работа может быть использована и интегрирована с существующей информационной системой приёмной комиссии. Цель разработки этой работы состоит в том чтобы , создать продукт в виде мобильного приложения которое обеспечивает деятельность комиссии.

Выгодой такого программного продукта являются уменьшение бумажной работы, удобство для поступающего, расширение существующей системы переходом в мобильную среду.

Итоговый продукт даст поступающим возможность:

- Возможность авторизации и регистрации;
- Указать первичные данные о себе;
- Подать и загрузить нужные документы;
- Получать оповещения от приёмной комиссии;

### 1.2 Термины и сокращения

Термины примененные в ходе разработки, а так же связанные с подходом к реализации и использованными технологиями отображены в таблице 1.

**Таблица 1 - Термины , сокращения, и их определению**

SDK	Software Development Kit
БД	База данных
UI	Пользовательский интерфейс
IDE	Integrated Development Environment
ЯП	Язык программирования
JSON	JavaScript Object Notation
NoSQL	Not only SQL

CRUD

Базовые функции БД: создание, чтение, модификация, удаление.

### 1.3 Обзор аналогичных приложения

Выборочный анализ высших учебных заведения на наличие аналогичного приложения был произведен с учётом национального рейтинга лучших технических вузов Казахстана, в котором наш вуз занимает 3-е место.

Были выбраны два учебных заведения занимающие соответственно первое и второе место. В результате проверки, аналогичных мобильных приложения в отеческом сегменте не выявлено. В большинстве случаев, требуемый функционал возложен на отдельный веб-сайт.

Однако, российский рынок имеет аналогичный продукт. К примеру, в 2016 году Томский государственный университет запустил мобильное приложение для абитуриентов «Хочу в ТГУ» см.рисунок 1.1

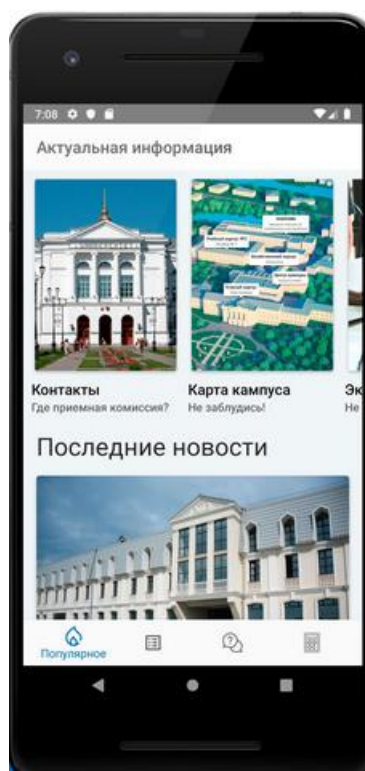


Рисунок-1.1 — Главный экран приложения Хочу в ТГУ

### 1.3.1 Обзор приложения Хочу в ТГУ

Из функциональности приложения стоит отметить такие возможности как подача необходимых документов, карта кампуса и рейтинг абитуриентов по результатам ЕГЭ, в целях оценивания абитуриентом вероятности на поступление по гранту.

Плюсы:

- Лаконичный и минималистский интерфейс;
- Интуитивно понятный , с учётом UX;
- Удобная навигация.

Минусы:

- Лишний, местами невостребованный функционал;
- Неприглядность отдельных графических элементов.

## **2 Технологический раздел**

### **2.1 Обзор используемых инструментов**

#### **2.1.1 Flutter SDK**

Flutter является фреймворком от компании Google, который позволяет создавать кроссплатформенные приложения, которые могут использовать один и тот же код. Спектр платформы широкий - это веб-приложения, мобильные приложения под Android и iOS, графические приложения под настольные операционные системы Windows, MacOS, Linux.

Особенностью работы с Flutter является то, что приложения под разные платформы могут иметь один и тот же код. Поскольку используемые платформы не эквиваленты, то какие-то отдельные части кода необходимо настраивать под определенную ОС, например, под iOS, но тем не менее большая часть кода может совпадать. Это позволяет разработчикам существенно сэкономить время и ресурсы на создание приложений под обе платформы.

В качестве языка разработки используется язык программирования Dart. При построении приложения Flutter транслирует код на Dart в нативный код приложения, которое можно запускать на Android или iOS или другой платформе.

Надо отметить, что Flutter относительно новый фреймворк. Хотя прототип фреймворка появился еще в 2015 году, а первая альфа-версия вышла в мае 2017 года, но первый стабильный релиз - Flutter 1.0 увидел свет только в декабре 2018 года. Тем не менее регулярно с небольшой периодичностью выходят подверсии, добавляя новый функционал и исправляя имеющиеся баги. На момент публикации данной статьи последней стабильной версией является Flutter 1.16.3.

#### **2.1.2 Язык программирования Dart**

Высокоуровневый, интерпретируемый язык программирования, которая разрабатывается компанией Google. На данный момент является единственной альтернативой JavaScript. В 2011 году стала доступна первая версия интерпретатора.

Создатель языка Марк Миллер называет повод создания Dart тем, что у JavaScript имеются «архитектурные изъяны», которые невозможно исправить. Dart не имеет вышесказанных изъянов.



При разработки ЯП разработчики стремились сделать его:

- с низким порогом вхождения, чтобы Dart было возможно быстро освоить;
- структурированным для быстрой и удобной разработки;
- производительным – для того чтобы написанные на этом ЯП приложения работали быстро как в браузерах, так и прочих окружениях (например, на смартфонах).

Dart имеет несколько преимуществ. Легкость в освоении языка. Любой JavaScript-программист сможет быстро освоить синтаксис кода на Dart. Лишь нужно только знать основные принципы этого ЯП.

- Высокая производительность. Созданные на Dart программы, в большинстве, показывают себя быстрее, чем созданные на JavaScript;
- Строгая типизация. Сделать ошибки при написании кода существенно труднее, чем при создании программ на JavaScript;
- Высокая оплата труда Dart-программистов. Данный язык программирования распространен не так сильно, как, к примеру, Python. Однако умеющие на нем писать ценятся на рынке куда выше. При этом язык активно развивается, и с каждым годом становится все более популярным;
- Документации. Так как разработкой интерпретатора для Dart занимается Google, весь функционал языка детально расписан. Это дает возможность оперативно получать ответы на возникающие вопросы, которые возникают в ходе обучения, так и разработки;
- Мобильная разработка на Dart многообещающее направление, сообщество которого бурно развивается. Это мощный и функциональный язык, на котором программировать.

### **2.1.2 IDE Android Studio**

Android Studio - это официальная интегрированная среда разработки (IDE) для разработки приложений Android. Он основан на IntelliJ IDEA, интегрированной среде разработки программного обеспечения на Java, и включает в себя инструменты редактирования кода и средства разработки.

Для поддержки разработки приложений в операционной системе Android Android Studio использует систему сборки на основе Gradle, эмулятор, шаблоны кода и интеграцию с Github.

Каждый проект в Android Studio имеет одну или несколько модальностей с исходным кодом и файлами ресурсов. К этим модальностям относятся модули приложений Android, модули библиотеки и модули Google AppEngine.

Android Studio использует функцию Instant Push для отправки изменений

кода и ресурсов в работающее приложение. Редактор кода помогает разработчику в написании кода и предлагает завершение, рефракцию и анализ кода. Приложения, созданные в Android Studio, затем компилируются в формат APK для отправки в Google Play Store.

Впервые программное обеспечение было анонсировано на Google I / O в мае 2013 года, а первая стабильная сборка была выпущена в декабре 2014 года. Android Studio доступна для настольных платформ Mac, Windows и Linux. Он заменил Eclipse Android Development Tools (ADT) в качестве основной среды разработки приложений для Android.

### **2.1.3 AVD Manager**

Одно из главных преимуществ Android Studio перед конкурентами — наличие встроенного в пакет эмулятора Android-устройств, на котором можно тестировать уже готовые проекты, корректность работы утилит, находящихся на стадии разработки. Это существенно удобнее, нежели использовать для всех тестов реальное устройство.

Эмулятор Android Studio обладает богатым функционалом. Его достаточно, чтобы проверить работу приложения на разных конфигурациях экрана, смоделировать точные соотношения сторон, нужные разрешения, технические характеристики смартфона, планшета или другого Android-устройства.

### 3 Проектный раздел

#### 3.1 Архитектура системы взаимодействия

Разрабатываемое мобильное приложение является инструментом манипуляцией данными. В процессе использования приложения, конечный пользователь задействует 4 этапа архитектурного взаимодействия. В первом этапе, Flutter Framework инициализирует слой абстракции, при помощи которой происходит взаимодействие с функцией рендеринга Flutter Engine. После выстраиваются так называемые виджеты, элементы интерфейса приложения из локальной памяти устройства с помощью GPU. Во втором этапе, данные в формате документа записываются в базу данных Firebase через статические виджеты.

На следующем этапе, по запросу динамических виджетов поступают данные из БД. В основе мобильного приложения лежит бессерверная архитектура с использованием сервисов Google Cloud (см.рис.3.1).

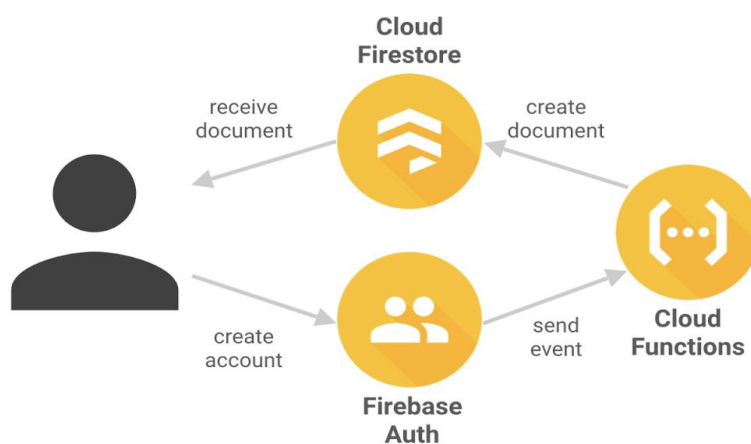


Рисунок-3.1 — Бессерверная архитектура

#### 3.2 Описание диаграммы деятельности

Диаграмма деятельности описывает, как действия координируются для предоставления услуг, которые могут быть на разных уровнях абстракции. Как

правило, конечная цель или состояние может быть достигнуто с помощью некоторых операций, в частности, операции могут быть последовательными или параллельными.

Для разработки диаграммы деятельности было использовано бесплатное web приложение creately.com для построения диаграмм с шаблонами.

Полноценное использование приложения начинается с прохождения авторизации и регистрации. В Firebase Authentication хранятся токены и уникальные id, позволяющие пройти авторизацию, они имеют свою структуру и содержатся в коллекциях.

После авторизации, идёт сбор данных из БД и их вывод в виджеты. При их отсутствии дается возможность внесения данных, в обратном случае их редактирование. Основная цель использования приложения в доставке своих персональных данных.

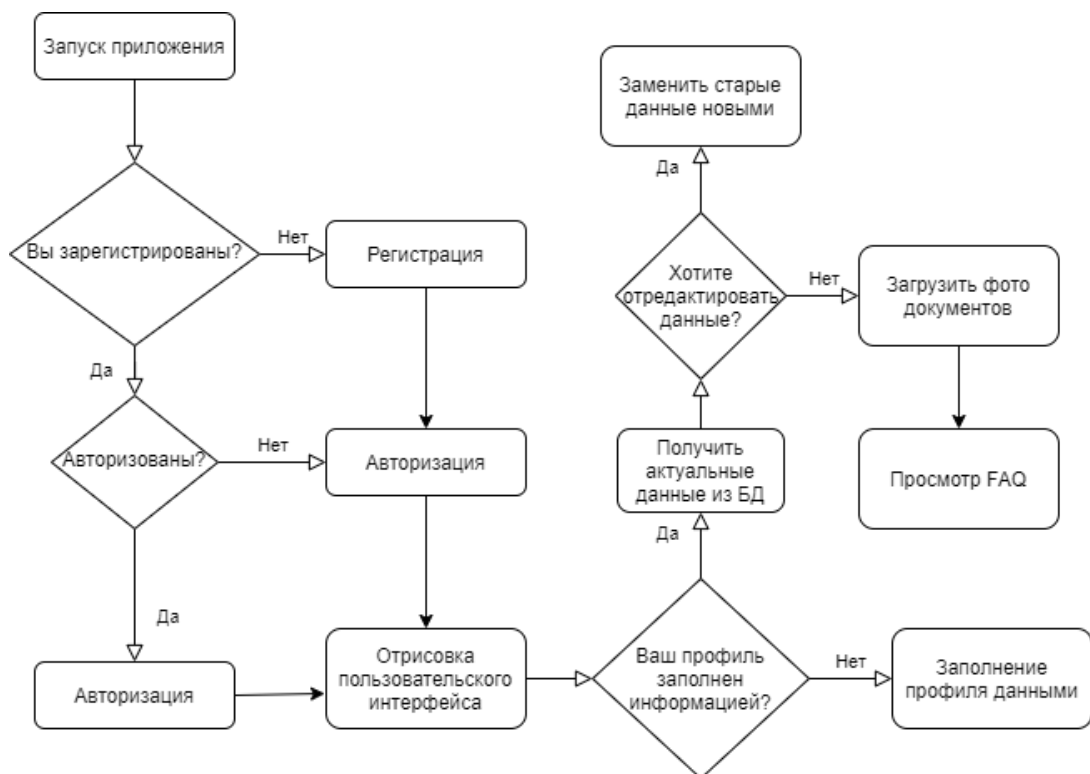


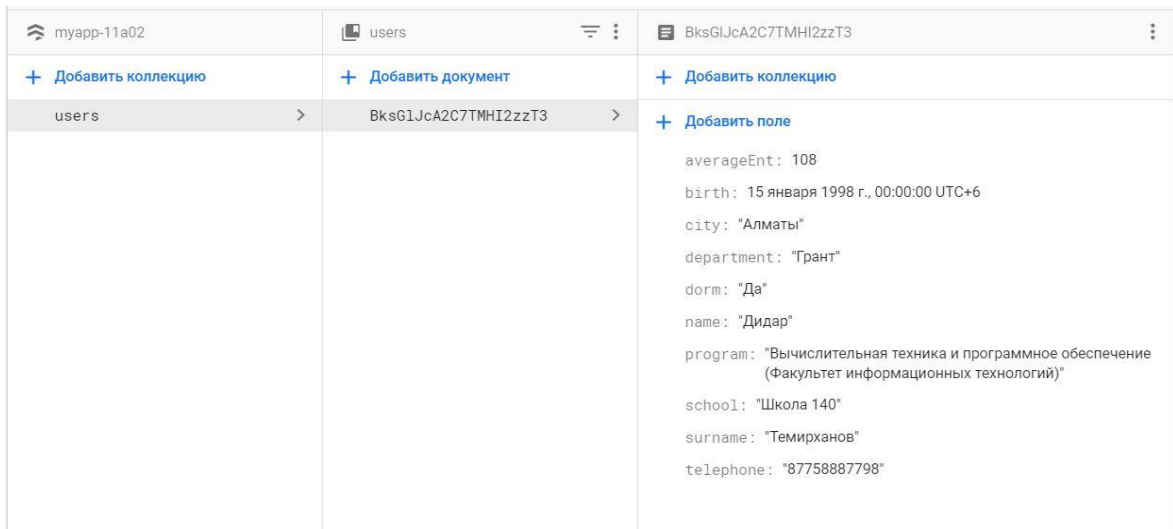
Рисунок-3.2 — Диаграмма деятельности

### 3.3 Проектирование базы данных

Для хранения данных используется FireBase Database, которая является облачной NoSQL базой данных. Данные поступающие из приложения хранятся

в большом дереве JSON. Перед началом эксплуатации была подготовлена структура JSON таким образом, чтобы данные были доступны более простым способом, избегая использование вложенных дочерних узлов.

Для каждого пользователя, создается собственный документ в коллекции. Документ в свою очередь содержит поля, которые принимают значения при CRUD операциях.



**Рисунок-3.3 — Коллекция записей**

Так как проект находится в тестовом режиме, разрешаем неавторизованным пользователям проводить операции CRUD следующим образом:

```
1 rules_version = '2';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4
5     // This rule allows anyone on the internet to view, edit, and delete
6     // all data in your Firestore database. It is useful for getting
7     // started, but it is configured to expire after 30 days because it
8     // leaves your app open to attackers. At that time, all client
9     // requests to your Firestore database will be denied.
10    //
11    // Make sure to write security rules for your app before that time, or else
12    // your app will lose access to your Firestore database
13    match /{document=**} {
14      allow read, write: if request.time < timestamp.date(2020, 6, 19);
15    }
16  }
17 }
```

**Рисунок-3.4 — Файл конфигурации базы данных**

База данных должна обеспечивать стабильную поддержку нашего мобильного приложения в автономном режиме, то есть сохранять данные при отсутствии интернет подключения и отправлять их сразу же при подключении к сети. Для этого , задаем такую конфигурацию:

```
FirebaseDatabase database = new FirebaseDatabase();  
database.setPersistenceEnabled(true);  
database.setPersistenceCacheSizeBytes(10000000);
```

**Рисунок-3.5 — Включение автономного режима**

## 4 Экспериментальный раздел

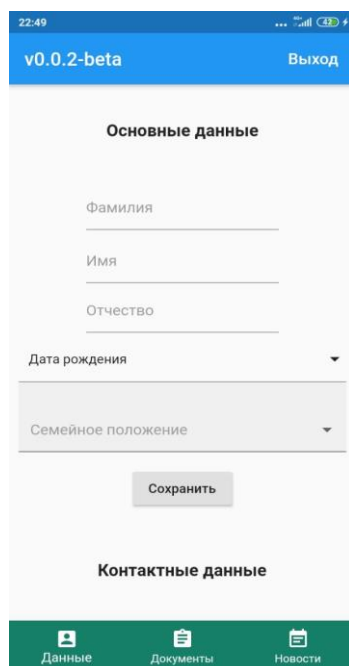
### 4.1 Проектирование пользовательского интерфейса

Мобильное приложение состоит из 5 экранов. Первый экран представляется после открытия приложения. (см.рис.4.1). Состоит из двух кнопок, двух input и логотипа университета. На этом экране осуществляется авторизация и регистрация.



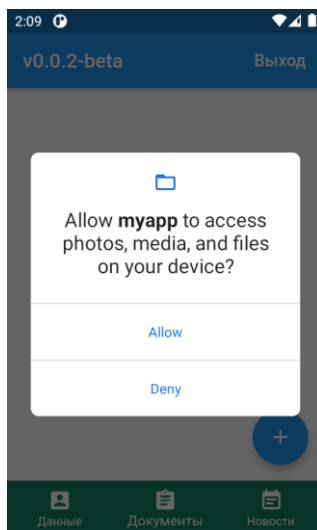
**Рисунок-4.1 – Первый экран приложения**

Второй экран предназначается для отображения полей, полей с выпадающими списками и прочих элементов в целях сбора и внесения введённых данных. После авторизации, пользователь попадает именно на второй экран. По структуре состоит из нижнего меню (BottomNavBar), в нём расположены три кнопки. При инициации нажатия на одну из кнопок, пользователь перенаправляется на соответствующий экран. (рис.4.2)



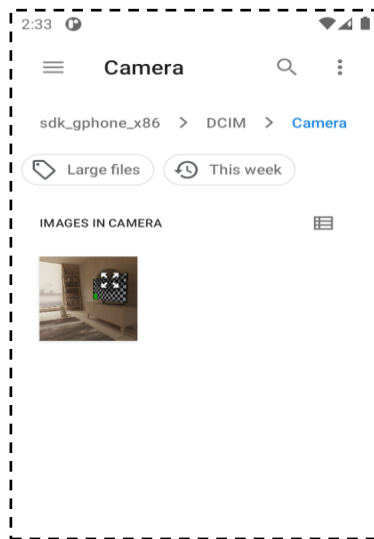
**Рисунок-4.2 — Второй экран приложения**

Третий экран и его элементы интерфейса нацелены на обеспечение возможности загрузки фотографии заявления абитуриента. Присутствует список разрешенных документов для отправки. Из элементов стоит отметить кнопку добавления, нажатие на которую инициирует запрос доступа (рис.4.3) а после открытие файлового менеджера (рис.4.4).



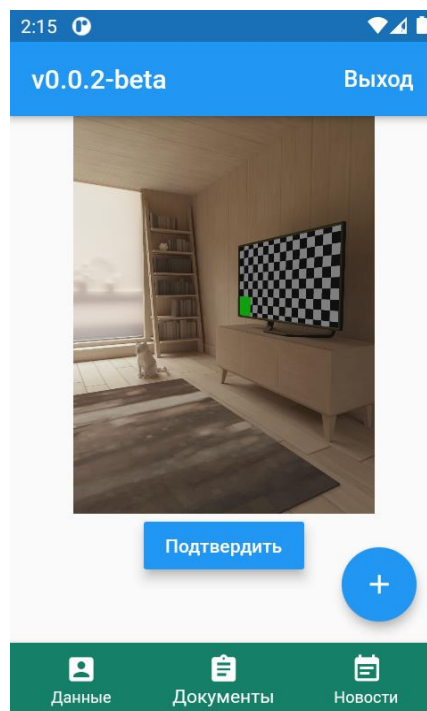
**Рисунок-4.3 — Запрос разрешения к доступу**





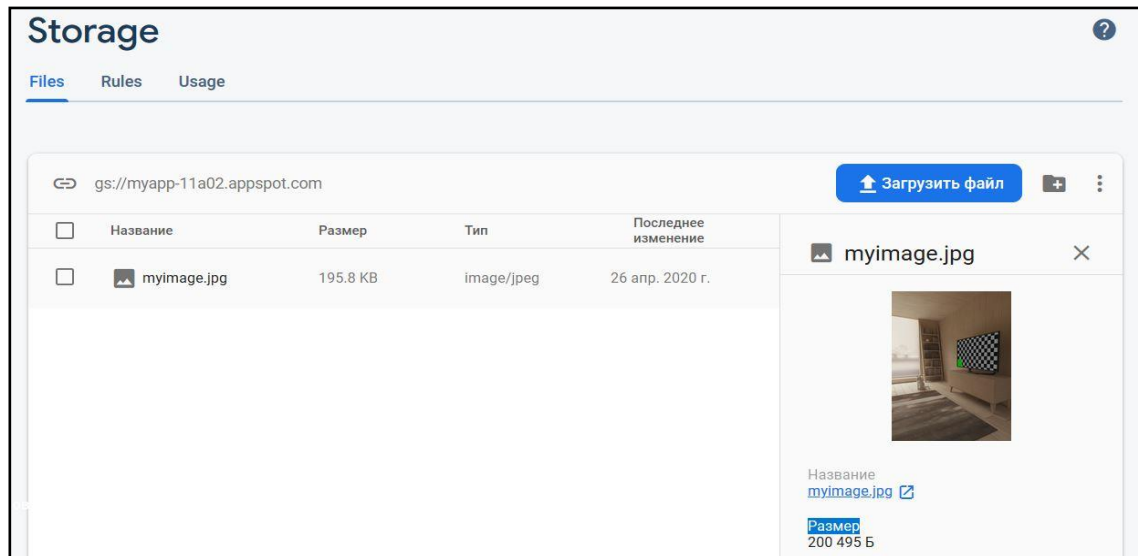
**Рисунок-4.4 — Менеджер файлов смартфона**

После выбора изображения , выводится превью и кнопка загрузки.(рис.4.5)



**Рисунок-4.5 — Загрузка фотографии в Storage**

После успешной загрузки, изображение хранится в Firebase Storage.(рис.4.6). Так же перспективно ,загруженное изображение открыто к отображению в пользовательском интерфейсе по требованию или необходимости.



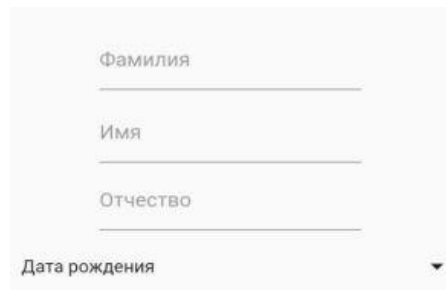
**Рисунок-4.5 — Хранилище фотографии в Storage**

## 4.2 Сбор и запись данных в БД

Одним из первых обязательных пунктов к выполнению является установка соединения с БД:

```
final  
dbRef=FirebaseDatabase.instance.reference().child("users");
```

Второй экран приложения , описанный в предыдущим разделе представляет собой пользовательский интерфейс из наборов текстовых полей, выпадающих списков.(рис 4.6). Все элементы интерфейса описаны в отдельном классе form.dart.



**Рисунок-4.6 – Виджеты TextField**

Каждый из этих элементов является виджетами и их значения, мы должны сохранить как локально так и записать в базу данных.

Для этой цели инициализируются контроллеры , позволяющие родительскому виджету манипулировать данными дочерних виджетов. Нужно добавить следующие параметры в виджеты:

```
Center(  
  child: Container(      margin: new EdgeInsets.all(10.0),  
    width: 200,  
    child: FocusWidget(  
      focusNode: _surname,  
      child: TextField(  
        controller: field1,  
        focusNode: _surname,  
        decoration: InputDecoration(  
          hintText: 'Фамилия'),  
        ),  
      ),  
    ),  
  ),  
  child: Container(  
    width: 200,  
    child: FocusWidget(  
      focusNode: _name,  
      child: TextField(  
        controller: field2,  
        focusNode: _name,  
        decoration: InputDecoration(  
          hintText: 'Имя'),  
        ),  
      ),  
    ),  
  ),  
),  
)
```

Далее идёт инициализация переменных:

```
TextEditingController field1 = new TextEditingController();  
TextEditingController field2 = new TextEditingController();  
TextEditingController field3 = new TextEditingController();  
TextEditingController field4 = new TextEditingController();
```

Чтобы сохранить значение из этих переменных в наш заранее заданный документ в коллекции БД Firebase, применяется метод push():

```
db.Ref.push().set({  
  "surname": field1.text,  
  "name": field2.text
```

```

    }).then((_) {
      Scaffold.of(context).showSnackBar(
        SnackBar(content: Text('Успешно добавлено')));
      field1.clear();
      field2.clear();
    }).catchError((onError) {
      print(onError);
    });
  });

```

Метод `push()` создать уникальный `id` для документа в коллекции, `set()` внесёт данные из текстовых полей в поля документа, `then()` будет выполнен после записи и выведет сообщение об успешной операции, с помощью `catchError` отслеживаются ошибки.

### 4.3 Асинхронная загрузка файлов в Firestore

Реализация загрузки изображения в облачное хранилище Firestore, подразумевает собой определение класса `DocApply` от суперкласса `StatefulWidget`.

```

class DocApply extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    return new _UploadScreenMode();
  }
}

```

Метод `createState()` возвращает экземпляр класса `_UploadScreenMode`, который наследует свойства от суперкласса `State`, что позволяет управлять состоянием. Данный экземпляр класса состоит из асинхронного метода загрузки изображения, обязательного метода `buildContext`, виджета `enableUpload()`.

В качестве переменной содержащей загруженное изображение, объявляется `sampleImage`. Метод `getImage()` задается с ключевыми словами `async` и `await` и типом `Future`, позволяющие сделать метод асинхронным. Асинхронность позволяет завершать другие задачи в ожидании завершения операции. В нашем случае, ожиданием является выбор изображения пользователем.

```

class _UploadScreenMode extends State<DocApply> {
  File sampleImage;
  Future getImage() async {

```

```

    var tempImage = await ImagePicker.pickImage(source:
ImageSource.camera);
    setState(() {
      sampleImage = tempImage;
    });
  }
}

```

Далее метод `build(BuildContext context)` отрисовывает кнопку добавления изображения в нижнем левом углу. Нажатие инициирует запрос на разрешение доступа к камере :

```

Widget build(BuildContext context) {
return new Scaffold(
  body: new Center(
    child: sampleImage == null ? Text('Добавьте фото заявления')
: enableUpload(),
  ),
  floatingActionButton: new FloatingActionButton(
    onPressed: getImage,
    tooltip: 'Загрузите изображение заявления',
    child: new Icon(Icons.add),
  ),
);
}

```

Третий и последний метод `enableUpload()` генерирует кнопку подтверждения выбора пользователя, загрузку изображения в `Firestore`.

```

Widget enableUpload() {
return Container(
  child: Column(
    children: <Widget>[
      Image.file(sampleImage, height: 300.0, width: 300.0),
      RaisedButton(
        elevation: 7.0,
        child: Text('Подтвердить'),
        textColor: Colors.white,
        color: Colors.blue,
        onPressed: () {
          final StorageReference firebaseStorageRef =
            FirebaseStorage.instance.ref().child('myimage.jpg');

```

```
final StorageUploadTask task =
    firebaseStorageRef.putFile(sampleImage);
    Scaffold.of(context).showSnackBar(
        SnackBar(content: Text('Успешно добавлено')));
    },
    )
    ],
    ),
);
}
}
```

## ЗАКЛЮЧЕНИЕ

Конечной целью дипломной работы заключается в обеспечении приёмной комиссии университета мобильным приложением. Итоговый продукт ориентирован как и на сотрудников так и на абитуриентов. Основным требованием к приложению является его способность к автоматизации некоторых процессов, таких как сбор информации и подача заявления, при этом исключая необходимость в заполнении бумажных бланков и пребывание в очереди.

В процессе интегрирования могут быть уточнены некоторые дополнительные требования к системе. Кроме системы по приему заявления, предусмотрен раздел для оповещения абитуриентов и сбор первичных данных.

В процессе исследования предметной области был проведён анализ рынка на аналоги. Рынок показал дефицит такого рода программного обеспечения. Большинство схожих по задаче проекты реализованы в виде веб-приложения.

Данное приложение открыто к программному усовершенствованию. В дальнейшем может быть интегрировано с существующей системой и снабжена с новыми функциями, например по считыванию баллов из сертификатов ЕНТ. Возможность абитуриенту отслеживать свою позицию в рейтинге и оценки своего шанса получения гранта. Это еще больше повысит привлекательность как приложения, так и университета.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Введение в Flutter SDK // Электронная версия на сайте  
<https://metanit.com/dart/flutter>
2. Особенности и преимущества Dart // Электронная версия на сайте  
<https://appfox.ru/blog/kod/preimushestvah-yazyka-programmirovaniya-dart>
3. Среда разработки Android Studio // Электронная версия на сайте  
<https://androidstudio.ru/>
4. Dart language overview // Электронная версия на сайте  
<https://dart.dev/guides/language>
5. Introduction to widgets // Электронная версия на сайте  
<https://flutter.dev/docs/development/ui/widgets-intro>



## **Приложение А**

(обязательное)

### Техническое задание

#### **А.1.1 Техническое задание на разработку мобильного приложения**

Настоящее техническое задание распространяется на разработку мобильного приложения для приёмной комиссии. Использовать данное приложение смогут сотрудники комиссии и абитуриенты. Приложение направлено на оказание услуг в образовательной сфере.

##### **А.1.1.1 Назначение**

Мобильное приложение приёмной комиссии для помощи абитуриентам онлайн поступления.

##### **А.1.1.2 Требование к внешнему виду приложения**

Внешний вид приложения обязан быть интуитивно понятным и прост в эксплуатации.

##### **А.1.1.3 Требование к функциональным характеристикам**

- Авторизация и регистрация пользователей
- Добавление и редактирование информации о себе
- Рабочая загрузка фотографии документов и заявления
- Предусмотреть раздел часто задаваемых вопросов

## **Продолжение приложения А**

### **А.1.1.4 Требование к надежности**

Предусмотреть контроль вводимой информации. Безопасность базы данных.

Предусмотреть корректность работы.

### **А.1.1.5 Требование к составу и параметрам технических средств**

Приложение должно работать на операционных системах Android, iOS.

Для разработки мобильного приложения требуется определить следующее технические требования:

- Dart
- Firebase Storage
- Firebase Database
- Flutter SDK

### **А.1.1.6 Требование к программной документации**

Написанный программной код должен быть закомментирован по усмотрению разработчика.