

Санкт-Петербургский государственный университет

Мясников Владислав Николаевич

Выпускная квалификационная работа

Разработка кроссплатформенного
мобильного приложения для
образовательной платформы

Уровень образования: бакалавриат

Направление *09.03.04 “Программная инженерия”*

Основная образовательная программа *СВ.5080.2017 “Программная инженерия”*

Научный руководитель:
доц., к.ф.-м.н. Д.В. Луцив

Рецензент:
инженер-программист ООО «Тензор» С.А. Никитин

Санкт-Петербург
2021

Saint Petersburg State University

Vladislav Myasnikov

Bachelor's Thesis

Development of a cross-platform mobile application for an educational platform

Education level: bachelor

Speciality *09.03.04 "Software Engineering"*

Programme *CB.5080.2017 "Software Engineering"*

Scientific supervisor:
C.Sc., docent Dmitry Luciv

Reviewer:
software engineer OOO «Tensor» Semyon Nikitin

Saint Petersburg
2021

Оглавление

Введение	4
1. Постановка задачи	5
2. Обзор предметной области	6
2.1. Мобильные приложения	6
2.2. Образовательная платформа TapCourse	7
2.3. Средства кроссплатформенной мобильной разработки	8
3. Мобильное приложение для платформы TapCourse	12
3.1. Архитектура	12
3.2. Пользовательский интерфейс	14
3.3. Конфигурация приложения	15
3.4. Взаимодействие с сервером	16
3.5. Навигация в приложении	18
4. Инструмент для создания графического представления “обогащённого текста”	20
4.1. Входной формат “обогащённого текста”	20
4.2. Преобразование входного формата	21
4.3. Создание графического представления	22
5. Апробация мобильного приложения	23
5.1. Тестирование на пользователях	23
5.2. Опрос тестировщиков	24
5.3. Результаты апробации	28
Заключение	29
Список литературы	30

Введение

С развитием Интернета стало появляться множество онлайн-сервисов, охватывающих различные сферы жизни человека. Так, сегодня любой человек может посетить врача или оплатить налоги, не выходя из дома. Всё, что нужно — это стабильное Интернет-соединение на вашем устройстве.

Сфера образования не стала исключением: в Интернете можно найти большое количество онлайн-курсов различного содержания и качества. Дистанционно даже можно обучиться многим профессиям, например, стать дизайнером интерьера, программистом или инструктором по фитнесу. Онлайн-образование всё больше набирает популярность. По прогнозам аналитиков, к 2026 году мировой рынок в этом направлении будет расти в геометрической прогрессии и достигнет более 370 миллиардов долларов [1]. Влияние на развитие этой сферы оказала и начавшаяся весной 2020 года пандемия COVID-19: школы и университеты стали массово переходить на дистанционное обучение, в связи с чем резко возрос спрос на онлайн-курсы и образовательные платформы для их поддержки.

Ввиду актуальности и перспективности сферы онлайн-образования была разработана веб-платформа для проведения курсов под названием *TapCourse*, которая в настоящее время используется онлайн-школами по подготовке к ЕГЭ *Экзамис* [2], *Екзам.Онлайн* [3] и *Unit Skills* [4].

В связи со значительной ролью смартфонов в жизни человека [5] и их перспективным использованием для онлайн-обучения [6] кажется актуальным, чтобы платформа *TapCourse* была доступна и в мобильном приложении. Это способствует развитию платформы как бизнес-продукта, а также послужит маркетинговым инструментом для расширения аудитории.

1. Постановка задачи

Целью данной работы является разработка с использованием кроссплатформенных технологий мобильного приложения, интегрируемого в экосистему образовательной платформы *TapCourse* и предназначенного для прохождения онлайн-курсов. Для достижения этой цели были поставлены следующие задачи.

1. Провести обзор существующих решений, в том числе технических средств по кроссплатформенной мобильной разработке.
2. Спроектировать архитектуру мобильного приложения для прохождения онлайн-курсов, а также пользовательский интерфейс.
3. Реализовать мобильное приложение с учётом выбранных инструментов.
4. Создать инструмент для генерации графического представления “обогащенного текста”.
5. Провести апробацию разработанного приложения на онлайн-школах клиентов.

2. Обзор предметной области

В данной главе приведён обзор существующих решений, представленных на рынке образовательных мобильных приложений. Также здесь проводится обзор образовательной платформы TapCourse и кроссплатформенных средств мобильной разработки.

2.1. Мобильные приложения

Рынок онлайн-образования представлен множеством платформ для дистанционного обучения. Далее будут рассмотрены мобильные приложения трёх наиболее крупных образовательных платформ по подготовке к ЕГЭ в России: *Умскул*, *Фоксфорд.Учебник* и *Maximum ЕГЭ*.

Умскул

Приложение предоставляет возможность выполнять практические задания и просматривать видео-уроки (онлайн-вебинары отсутствуют).

Однако процесс взаимодействия с приложением сопровождался заметными задержками реакции интерфейса на действия пользователя (при определённых сценариях использования, например, при пролистывании списка заданий). Кроме того, была отмечена общая нестабильность приложения согласно отзывам пользователей в Google Play и App Store.

Фоксфорд.Учебник

Приложение предоставляет доступ к текстовым и видео-материалам по школьным предметам для разных классов.

Однако функциональность по прохождению онлайн-курсов не была перенесена из веб-версии.

Maximum ЕГЭ

Приложение позволяет просматривать текстовые и видео-материалы, а также проходить тестовые задания.

Однако организация вебинаров не встроена в данную образовательную платформу (используется сервис для видео-конференций *Zoom*). Также не было обнаружено заданий, в которых нужно прикреплять файлы (например, для написания сочинения по русскому языку).

Выводы

Таким образом, даже у одних из наиболее крупных онлайн-школ по подготовке к ЕГЭ в России мобильные приложения имеют определённые недочёты, связанные с пользовательским интерфейсом либо ключевой функциональностью.

В частности, ни в одно из трёх приложений не были встроены вебинары, которые являются основной составляющей онлайн-обучения для взаимодействия учеников с преподавателями. TapCourse позволяет проводить вебинары на базе платформы, и в мобильном приложении также будет реализована данная функциональность.

2.2. Образовательная платформа TapCourse

Образовательная платформа, неразрывно связанная с темой данной работы, была разработана студентами Ярославского Государственного университета им. П. Г. Демидова Евгением и Романом Власовыми. Информацию о платформе можно найти в [7].

Данная образовательная платформа имеет следующие основные функциональные возможности:

- создание и редактирование онлайн-курсов преподавателями с возможностью последовательного предоставления доступа к учебным материалам;
- проверка выполненных учениками заданий (часть автоматизирована), выставление оценки с комментариями по проделанной работе;
- прохождение онлайн-курсов: чтение и просмотр лекций, посещение вебинаров, выполнение практических заданий;

- встроенная возможность оплаты курсов.

Также платформа TapCourse обладает следующими особенностями:

- персонализация под каждую из онлайн-школ: варианты авторизации и оплаты, компоненты интерфейса (цвета, логотипы и пр.);
- система геймификации: баллы за задачи, разделение учащихся на группы — *кланы*, рейтинги пользователей и кланов.

Весной 2021 года платформа находилась в использовании у трёх онлайн-школ по подготовке к ЕГЭ *Экзамис* [2], *Екзам.Онлайн* [3] и *Unit Skills* [4].

2.3. Средства кроссплатформенной мобильной разработки

Для создания мобильного приложения был выбран кроссплатформенный подход [8], поскольку он позволяет следующее:

- повторно использовать код с общей функциональностью между Android- и iOS-приложениями;
- уменьшить время выхода на рынок мобильного приложения по сравнению с нативной разработкой;
- обеспечить *usability* (плавность анимации и переключения между экранами, отзывчивость интерфейса и т.п.), практически, такую же как и при нативном подходе.

Далее следует обзор основных инструментов для кроссплатформенной мобильной разработки: *Xamarin*, *React Native* и *Flutter*.

Xamarin

Данный фреймворк с открытым исходным кодом является программным продуктом компании Microsoft и предназначен для кроссплатформенной разработки мобильных приложений на языке C# [9].

Фреймворк Xamarin имеет следующие достоинства:

- потенциально может обеспечить производительность как при нативном подходе;
- позволяет использовать единую кодовую базу для пользовательского интерфейса на основе *Xamarin.Forms* [10];
- имеет функцию *XAML Hot Reload* для внедрения изменений в пользовательский интерфейс приложения “на лету” (без необходимости повторной сборки всего приложения).

Данный фреймворк также обладает определёнными недостатками, представленными ниже:

- требует дополнительных финансовых затрат для профессионального и корпоративного использования;
- не рекомендуется к использованию при создании сложного пользовательского интерфейса (применение нативного подхода предпочтительнее).

React Native

Данный фреймворк является одним из наиболее известных и популярных среди кроссплатформенных фреймворков для мобильной разработки [11]. В качестве языка разработки используется JavaScript. Первая версия фреймворка была выпущена в 2015 году компанией Facebook, Фреймворк React Native имеет следующие достоинства:

- предоставляет доступ к одной из крупнейших в мире экосистеме пакетов;
- обладает крупным сообществом разработчиков;
- имеет функцию *Fast Refresh*, позволяющую ускорить процесс внедрения изменений в код приложения, запущенного на устройстве или эмуляторе.

Данный фреймворк обладает следующими недостатками:

- обновление версии используемого фреймворка в большинстве случаев является трудозатратным процессом;
- может давать меньшую производительность по сравнению с фреймворком Flutter (по причине использования специального “моста” для взаимодействия нативного и Javascript-кода).

Flutter

Данный фреймворк предназначен для разработки приложений на Android, iOS, Linux, Mac, Windows, Google Fuchsia и Web с использованием единой кодовой базы [12]. Первый выпуск стабильной версии состоялся в 2018 году компанией Google.

Фреймворк Flutter имеет следующие достоинства:

- позволяет создавать унифицированный интерфейс для всех целевых платформ при помощи собственной библиотеки графических компонентов;
- имеет функцию *Hot Reload*, позволяющую вносить быстрые изменения в код запущенного приложения;
- для разработчиков представлена отличная документация по использованию фреймворка;
- популярность Flutter непрерывно растет [13].

Данный фреймворк обладает следующими недостатками:

- сообщество разработчиков и количество библиотек значительно меньше, чем у фреймворка React Native;
- пользовательский интерфейс приложения не на 100% повторяет нативный;
- приложение имеет довольно большой размер с самого начала.

Выводы

По результатам проведенного обзора было решено выбрать Flutter в качестве инструмента разработки. Данный фреймворк предоставляет возможность в создании производительного мобильного приложения, обладающего унифицированным выразительным и гибким пользовательским интерфейсом любого уровня сложности. А с плавностью анимаций и жестами также позволяет достичь интуитивно понятного взаимодействия пользователя с приложением. Кроме того, благодаря декларативному способу задания графических компонентов совместно с функцией *Hot Reload*, можно добиться быстрого процесса разработки и выхода на рынок.

Также стоит отметить, что Flutter набирает всё большую популярность: рост сообщества, развитие экосистемы, доклады на конференциях (*Google I/O*, *Mobile World Congress*, etc.), использование фреймворка крупными компаниями (*Google Assistant*, *eBay Motors*, *Alibaba.com*, etc.). К примеру, одним из наиболее известных Flutter-приложений в России является *Meduza* одноимённого новостного агрегатора.

Таким образом, у Flutter есть огромный потенциал в мобильной разработке, и, несмотря на его недостаточную зрелость, интерес разработчиков и компаний к нему с каждым годом только усиливается.

3. Мобильное приложение для платформы TapCourse

В данной главе будут рассмотрены ключевые аспекты, связанные с разработкой мобильного приложения для образовательной платформы TapCourse.

3.1. Архитектура

Мобильное приложение реализовано на языке Dart с использованием технологии кроссплатформенной разработки Flutter.

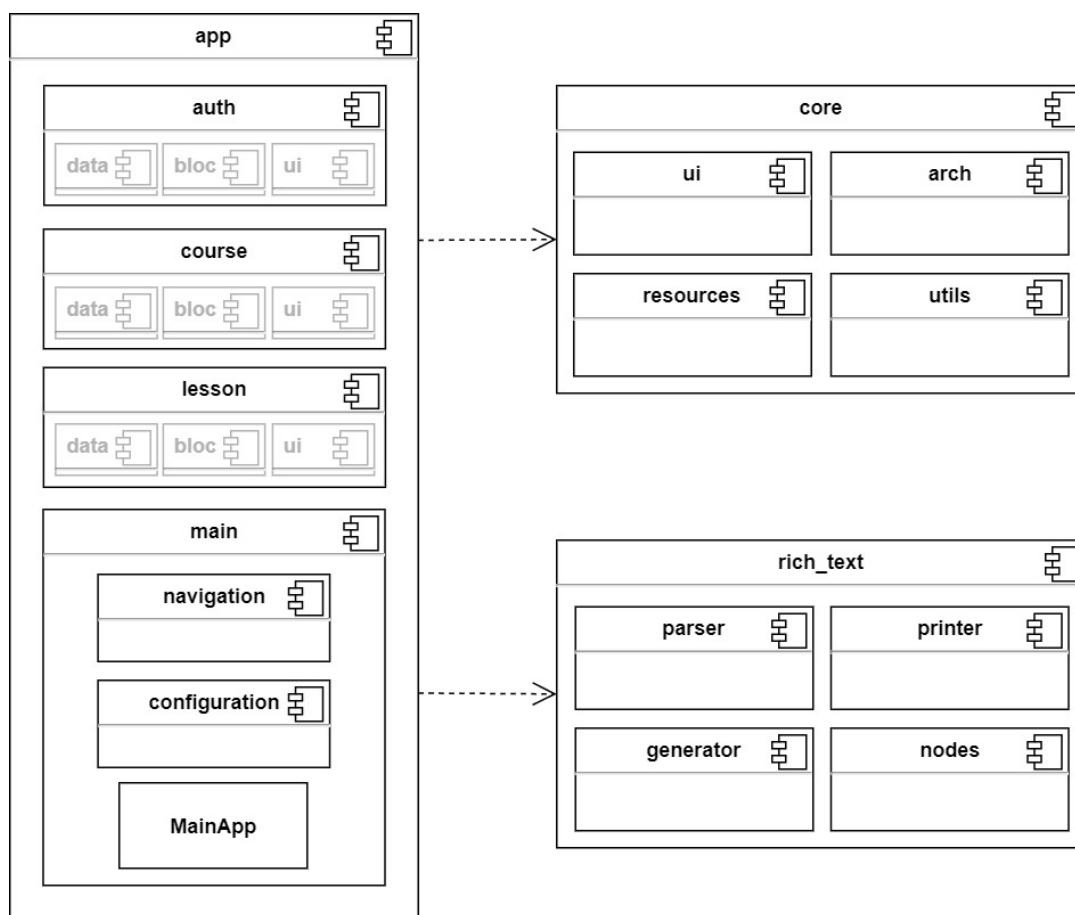


Рис. 1: Диаграмма компонентов мобильного приложения

Кодовая база включает в себя три модуля (Рис. 1): модуль приложения (*app*) и два дополнительных (*core* и *rich_text*), от которых данный модуль зависит.

- **Модуль *app*.** Данный модуль содержит модели данных, элементы бизнес-логики и пользовательского интерфейса с разделением по функциональности (авторизация, пользовательские курсы и уроки). Также здесь определены компоненты, отвечающие за навигацию в приложении и задание начальной конфигурации. Наконец, в данном модуле задаётся начальная точка входа в приложение (функция *main*).
- **Модуль *core*.** Данный модуль содержит общеиспользуемые компоненты, например, элементы пользовательского интерфейса и архитектуры. Также он хранит различные ресурсы, используемые в приложении (строки, изображения, иконки, шрифты и пр.).
- **Модуль *rich_text*.** Данный модуль содержит инструмент по генерации графического представления "обогащенного текста", детали реализации которого будут рассмотрены в следующей главе.

Использование шаблона BLoC

Для реализации бизнес-логики в приложении был выбран архитектурный шаблон *Business Logic Component (BLoC)*, а также библиотека для работы с данным шаблоном на Flutter [14].

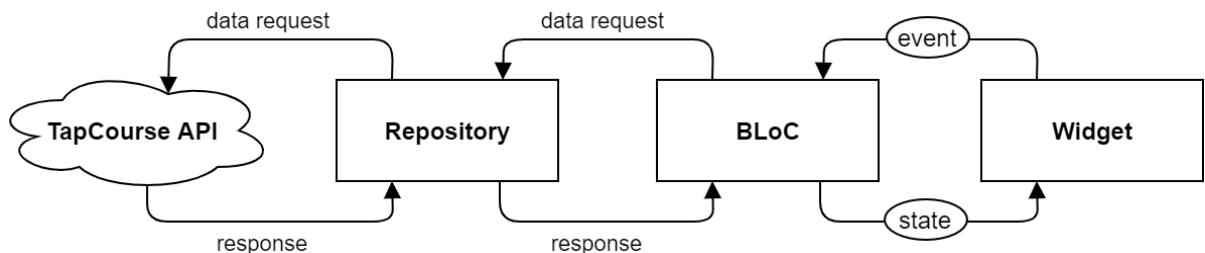


Рис. 2: Принцип работы архитектурного шаблона BLoC

Принцип работы с данным шаблоном представлен на Рис. 2 и заключается в следующем.

1. Из элемента пользовательского интерфейса (виджета) посылается событие в BLoC.

2. ВLoC, получив событие от виджета, запрашивает требуемые данные у репозитория, который, в свою очередь, запрашивает их у сервера.
3. Затем ВLoC в зависимости от ответа сервера создает соответствующее состояние и отправляет его обратно виджету.
4. В результате виджет либо обновляется, либо полностью замещается другим виджетом.

Все действия являются асинхронными, что позволяет не блокировать пользовательский интерфейс в промежутке времени от создания события до получения состояния виджетом.

3.2. Пользовательский интерфейс

Для проектирования пользовательского интерфейса использовался инструмент *Figma*, позволяющий создавать произвольные компоненты интерфейса и прототипы экранов приложения [15].

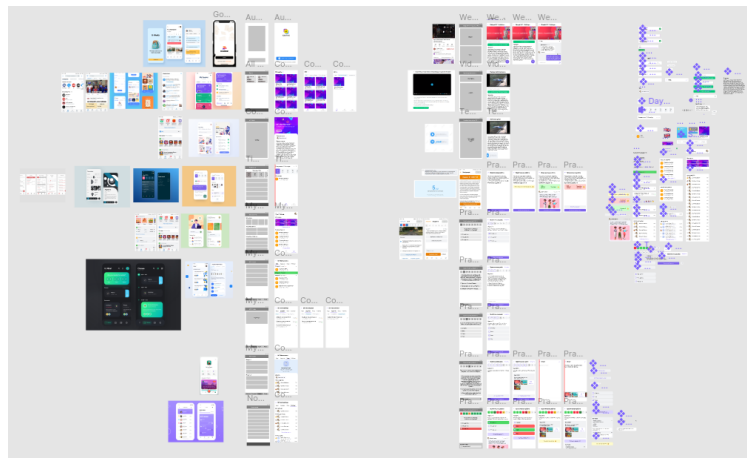


Рис. 3: Макет пользовательского интерфейса в Figma

С помощью данного инструмента был разработан набор собственных графических компонентов. Такой подход был выбран для того, чтобы придать единство и уникальность пользовательскому интерфейсу на обеих целевых платформах.

Также были спроектированы макеты экранов приложения и навигация между ними, задана цветовая палитра и основные стили текста.

3.3. Конфигурация приложения

Ввиду того, что мобильное приложение предназначено для использования несколькими онлайн-школами, необходимо обеспечить его гибкую настройку. С этой целью в приложение для каждой школы добавлен специальный конфигурационный файл.

```
{
  "urls": {
    "platform": "http://localhost/api",
    "webplatform": "http://localhost",
    "websocket": "ws://localhost/websocket",
    "privacy_policy": "http://localhost/policy"
  },
  "auth": {
    "vkontakte": {
      "client_id": 0
    },
    "telegram": {
      "bot_url": ""
    }
  },
  "logo_title": "<logo_title>",
  "logo_path": "path/to/logo_asset.png"
}
```

Рис. 4: Шаблон конфигурационного файла для мобильного приложения

На Рис. 4 представлен шаблон данного файла в формате JSON. В конфигурационном файле содержится следующее:

- основные веб-адреса для взаимодействия с платформой TapCourse;
- данные для авторизации через социальные сети ВКонтакте и/или Телеграм;
- компоненты пользовательского интерфейса (заголовок и логотип на начальном экране загрузки приложения).

Данные из конфигурационного файла соответствующей школы извлекаются при запуске приложения и представляются во внутренней структуре в виде экземпляра класса, реализующего шаблон *Одиночка*, для дальнейшего использования.

3.4. Взаимодействие с сервером

В TapCourse применяется архитектурный стиль REST, поэтому основное взаимодействие мобильного приложения с платформой происходит посредством HTTP-запросов.

Авторизация

В приложении реализована авторизация OAuth 2.0 через социальные сети *ВКонтакте* и *Telegram*.

Процесс авторизации через социальную сеть *ВКонтакте* происходит за несколько шагов:

1. пользователь вводит свой логин и пароль и совершает вход в социальную сеть;
2. пользователь предоставляет доступ приложению к определённой информации об аккаунте *ВКонтакте* (в частности, к ФИО и фотографии профиля);
3. получив от сервера *ВКонтакте* код авторизации, приложение отправляет данный код на сервер платформы TapCourse;
4. платформа отправляет ответное сообщение с JWT-токеном (JSON Web Token).

Процесс авторизации через социальную сеть *Telegram* происходит за несколько шагов:

1. пользователь перенаправляется на страницу чата с Telegram-ботом (в браузере или официальном приложении Telegram);
2. в данном чате пользователь нажимает кнопку “Войти”;
3. происходит перенаправление обратно в мобильное приложение, куда передаётся определённая информация о Telegram-аккаунте пользователя (в частности, ФИО, псевдоним и фотография профиля);

4. полученная информация отправляется на сервер TapCourse, откуда в ответ приходит JWT-токен.

Полученный одним из описанных выше способов JWT-токен сохраняется на мобильном устройстве и используется для дальнейшего взаимодействия приложения с платформой. Если при очередном запросе обнаруживается, что токен больше не действителен (например, истек срок его действия), происходит автоматический переход на страницу авторизации.

Онлайн-вебинары

В приложении реализовано проведение вебинаров для общения учеников и преподавателей онлайн-школ. Вебинар представлен в виде встроенной YouTube-трансляции и онлайн-чата через протокол WebSocket.

Для организации онлайн-чата используется двухканальное взаимодействие с сервером: один канал — для отправления сообщений от пользователя, другой канал — для получения новых сообщений от сервера.

Также чат поддерживает *пагинацию*, то есть автоматически подгружает более ранние сообщения, когда пользователь достигает верхнего края списка сообщений.

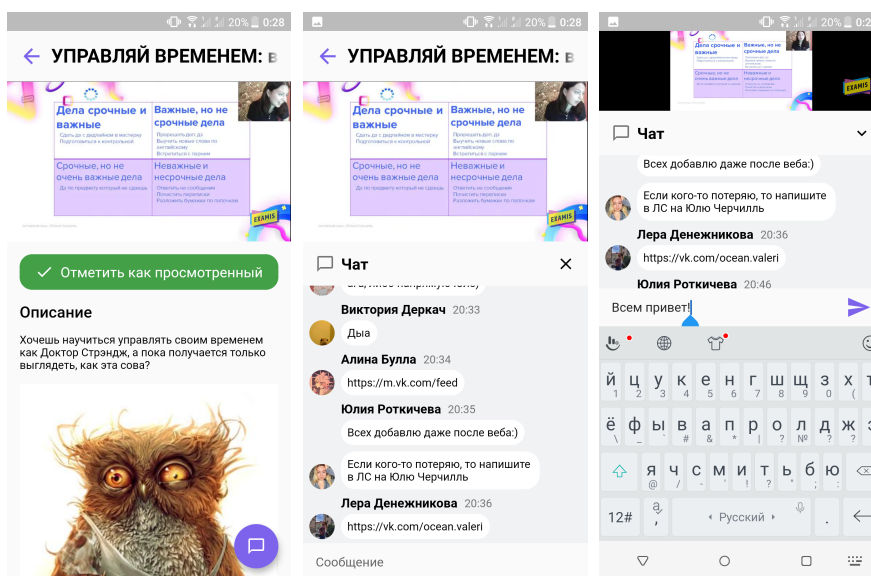


Рис. 5: Снимки экрана с проведением онлайн-вебинара в мобильном приложении

3.5. Навигация в приложении

Приложение состоит из следующих экранов.

- *Первоначальный экран загрузки.* Отображаются логотип и название школы, индикатор загрузки.
- *Экран авторизации.* Включает в себя логотип и название школы, кнопки для авторизации и ссылку на пользовательское соглашение.
- *Страница “Мое обучение”.* Здесь отображаются приветственное сообщение (с именем и фотографией пользователя), список курсов, на которые записан пользователь, а также уроки и работы, которые нужно выполнить на текущей неделе.
- *Учебный план курса.* На данном экране представлена информация о содержимом курса пользователя (список уроков и практических работ с разделением на модули). Для каждого урока отображается его название, крайний срок выполнения и текущий статус урока (например, “выполнен” или “находится на проверке”).
- *Страница урока.* Здесь представлены материалы по уроку и способ его выполнения (кнопка “Выполнить” или поле для ввода кодового слова). Уроки могут быть трёх видов: с текстовыми или видео-материалами, а также онлайн-вебинары со встроенным чатом.
- *Начальная страница практической работы.* Содержит общую информацию о работе (описание, количество заданий, комментариев куратора).
- *Основная страница практической работы.* Содержит задания и навигационную панель для переключения между ними. Каждое практическое задание включает в себя формулировку, форму для указания ответа пользователя или результат выполнения (правильный ответ, комментарий куратора, полученные баллы).

- Редактор решения к заданию с развернутым ответом. Предоставляет возможность прикреплять и откреплять файлы, изменять текстовый ответ пользователя.

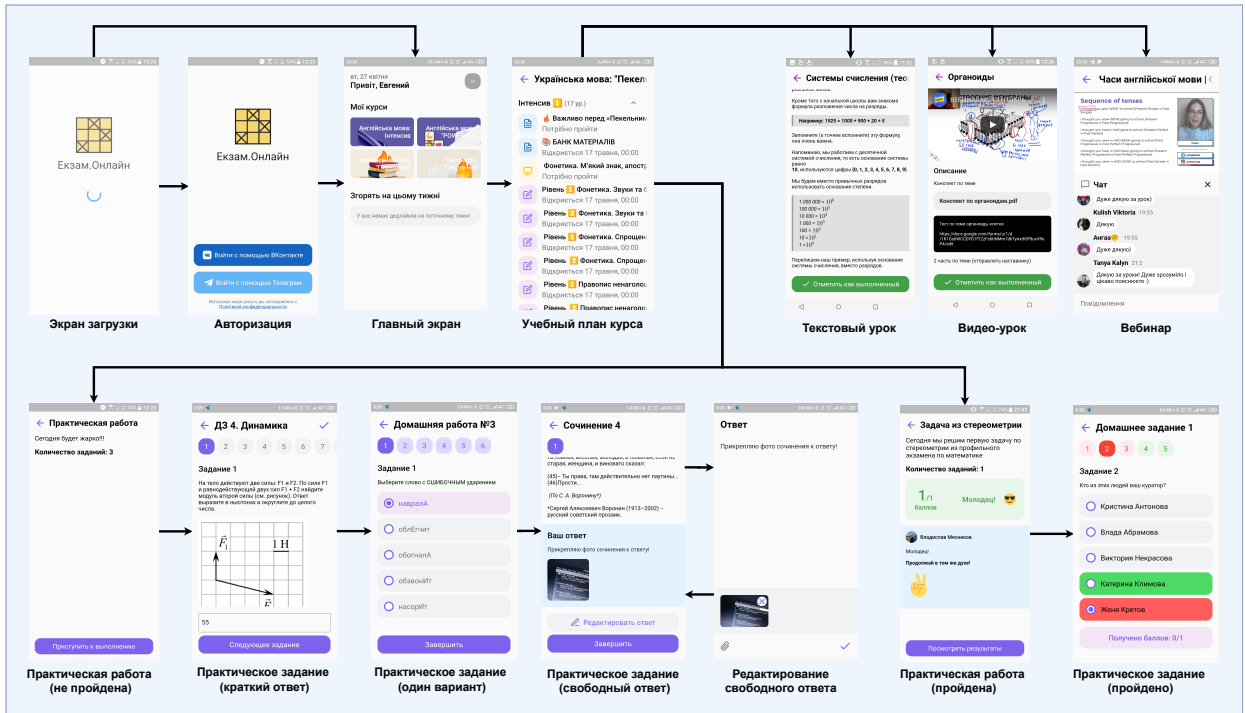


Рис. 6: Схема навигации в приложении

На Рис. 6 представлена схема навигации в мобильном приложении. При его запуске пользователь попадает на экран загрузки. Если доступ к основному содержимому разрешён (JWT-токен есть и действителен), то происходит переход на главную страницу с курсами ученика, иначе открывается страница авторизации. Перейдя на главную страницу, пользователь может просматривать учебный план по каждому из курсов, на которые он записан. Учебный план представлен в виде модулей, содержащих уроки и практические работы. Авторизованный ученик имеет возможность просматривать текстовые материалы и видеоуроки, выполнять практические работы и узнавать их результаты, а также участвовать в онлайн-вебинарах и общаться в чате с другими пользователями.

4. Инструмент для создания графического представления “обогащённого текста”

Обогащённый текст — это текст, в котором присутствуют различные стили (полужирный, курсив и др.), а также дополнительные компоненты (изображения, формулы, таблицы и пр.).

В платформе TapCourse в виде “обогащённого текста” представлена различная информация (в частности, описание курсов и уроков, текстовые лекции, задания к практическим работам, комментарии кураторов и др.).

ProseMirror — это набор инструментов для реализации текстовых редакторов в браузере, обладающий собственными форматами хранения “обогащённого текста” в виде JSON. Платформой TapCourse используется модифицированная версия данного формата.

Однако в официальном сервисе по размещению библиотек для Flutter [16] существующего решения по отображению “обогащённого текста” в формате ProseMirror *не было найдено*. В связи с этим возникла необходимость в разработке собственного инструмента, о котором и пойдёт речь в данной главе.

4.1. Входной формат “обогащённого текста”

Входной формат “обогащённого текста” представлен в виде JSON и имеет древовидную структуру из *звеньев*. Каждое *звено* может быть внутренним узлом дерева либо листом, а также содержать дополнительные атрибуты. На Рис. 7 представлен пример фрагмента данного формата.

Разработанный инструмент поддерживает следующие основные типы *звеньев*: обычные тексты со стилями (полужирный, курсив, подчёркнутый, перечёркнутый), текстовые ссылки со стилями, заголовки, упорядоченные и неупорядоченные списки, блоки кода и цитат, изображения с подписью, прикрепленные файлы, формулы и таблицы.

```

{
  "type": "blockquote",
  "content": [
    {
      "type": "paragraph",
      "content": [
        {
          "type": "text",
          "marks": [
            {
              "type": "bold"
            }
          ],
          "text": "Алфавит системы счисления - это цифры, которые
используются для записи чисел в этой системе."
        }
      ]
    }
  ]
},

```

Рис. 7: Пример фрагмента входного формата “обогащённого текста”

4.2. Преобразование входного формата

Преобразование входного формата JSON во внутреннюю древовидную структуру для последующей генерации графического представления происходит в два этапа. На первом этапе выполняется синтаксический разбор входной строки формата JSON в промежуточное дерево представления. На втором этапе выполняется преобразование промежуточного дерева в конечное дерево представления. На Рис. 8 представлен пример работы второго этапа.

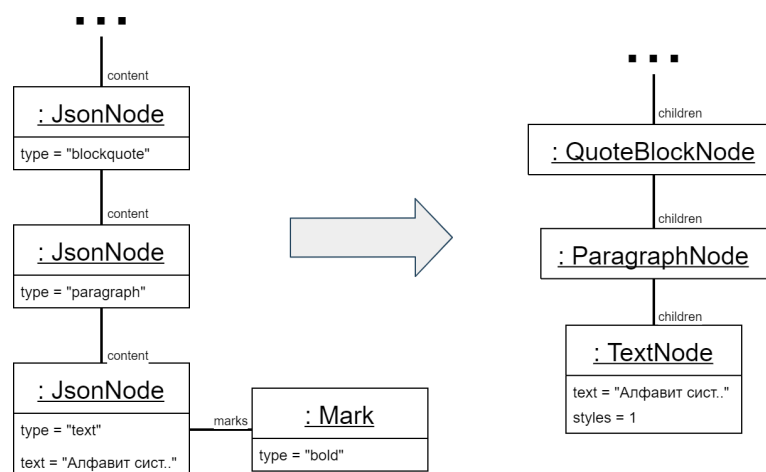


Рис. 8: Процесс преобразования промежуточного дерева в конечное на примере фрагмента “обогащённого текста”

4.3. Создание графического представления

В качестве входного параметра для генератора “обогащённого текста” передаётся конечное дерево представления, полученное в результате преобразования данных на предыдущем шаге. Процесс генерации построен на основе шаблона проектирования *Посетитель* (Рис. 9).

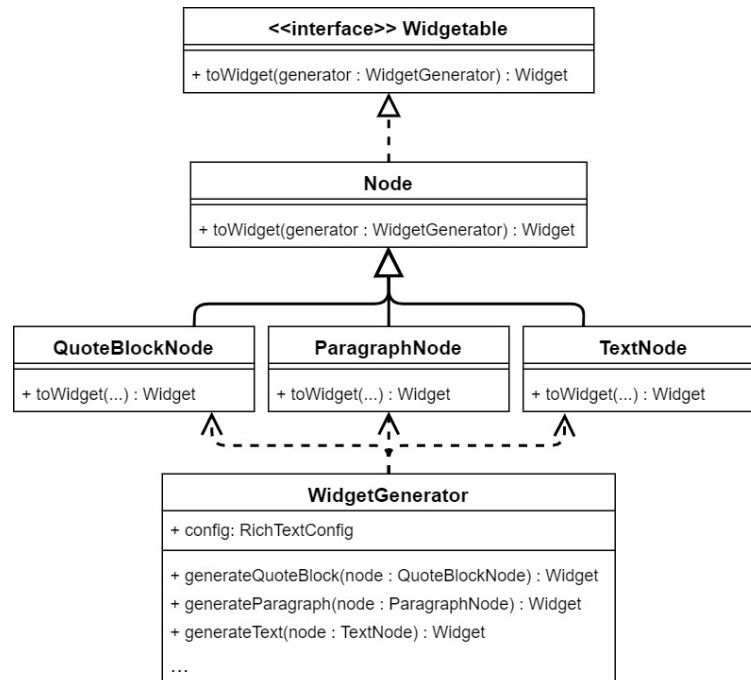


Рис. 9: Схема генерации графического представления “обогащённого текста” с использованием шаблона проектирования *Посетитель*

Каждое звено “обогащённого текста” является наследником класса *Node* и реализует интерфейс *Widgetable*. Данный интерфейс имеет единственный метод *toWidget*, принимающий объект класса *WidgetGenerator* в качестве аргумента и возвращающий элемент пользовательского интерфейса (виджет).

В классе *WidgetGenerator* определены функции по генерации графического представления для всех видов звеньев. Таким образом, реализация метода *toWidget* для каждого звена сводится к вызову соответствующей функции генератора.

Итоговым результатом работы генератора является дерево виджетов, которое отображается на экране устройства посредством фреймворка Flutter.

5. Апробация мобильного приложения

Апробация проводилась на платформе Android. К тестовой версии мобильного приложения был предоставлен доступ для нескольких десятков учеников онлайн-школы *Экзамис*. Тестирование продолжалось в течение недели, после чего тестировщикам было предложено пройти опрос.

На следующем этапе также было проведено тестирование приложения для онлайн-школы *Екзам.Онлайн*, в котором приняло участие более 50 учеников.

По результатам проведённого тестирования на учащихся двух онлайн-школ, а также на основании опроса пользователей было выявлено и устранено несколько проблем, поставлены дальнейшие задачи по улучшению и развитию мобильного приложения.

5.1. Тестирование на пользователях

Для размещения тестовой версии в Google Play использовался онлайн-сервис *Play Console* [17]. Данный сервис позволяет проводить следующие виды тестирования:

- *внутреннее* — среди выбранных тестировщиков без ожидания проверки опубликованной версии компанией Google (не более 100 тестировщиков);
- *закрытое* — среди выбранных тестировщиков с ожиданием проверки опубликованной версии компанией Google;
- *открытое* — то же, что и закрытое, но доступное для всех пользователей Google Play.

Для распространения приложений среди разработчиков команды TarCourse применялось внутреннее тестирование, а среди учащихся онлайн-школ — закрытое тестирование. Такой подход позволяет предварительно тестировать новые версии приложений прежде, чем выпускать их для пользователей.

Для автоматической сборки, подписки и публикации мобильных приложений для двух школ использовался онлайн-сервис *Codemagic* [18].

На первом этапе в закрытом тестировании приняло участие несколько десятков учеников онлайн-школы *Экзамис*. Каждому тестирующему была предоставлена ссылка на скачивание мобильного приложения в Google Play и время в течение одной недели на тестирование. По истечении этого времени им было предложено пройти опрос, о котором будет рассказано в следующем разделе.

На втором этапе в закрытом тестировании приняло участие более 50 учеников онлайн-школы *Екзам.Онлайн*.

5.2. Опрос тестирующих

В опросе приняло участие 10 учащихся онлайн-школы *Экзамис*. Каждому из них было предложено ответить на ряд вопросов, связанных с тестовой версией приложения.

Вопросы были разделены на следующие три группы:

1. вопросы, связанные с удобством использования мобильного приложения;
2. вопросы, в которых необходимо дать оценку пользовательскому интерфейсу мобильного приложения;
3. вопросы, касающиеся функциональных возможностей мобильного приложения.

Далее будут рассмотрены результаты проведённого опроса по каждой из групп вопросов. Данные результаты представлены в виде графиков распределений ответов учащихся и комментариев к ним.

Группа вопросов 1. Удобство использования

Первая группа вопросов была связана с удобством использования мобильного приложения по сравнению с веб-версией.

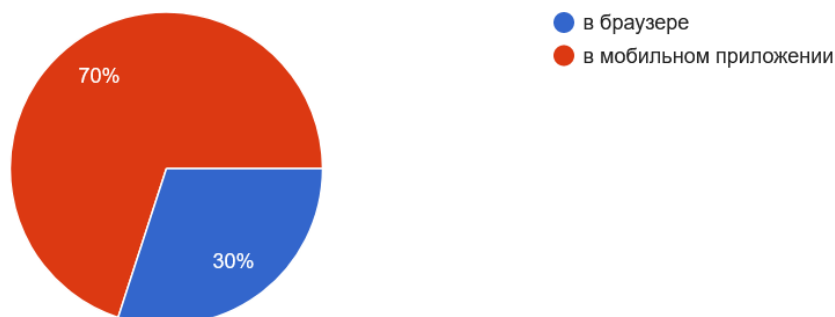


Рис. 10: Распределение ответов на вопрос: *“На мобильном устройстве Вам удобнее работать в браузере или мобильном приложении?”*

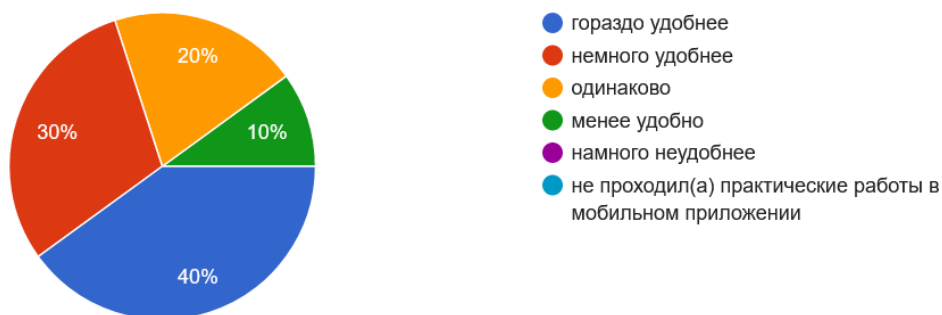


Рис. 11: Распределение ответов на вопрос: *“Насколько Вам удобнее проходить практические работы в мобильном приложении по сравнению с браузером?”*

На основании ответов учащихся (Рис. 10, Рис. 11) можно заключить, что на данный момент мобильное приложение не в полной мере превосходит веб-версию по удобству в случаях использования, связанных с прохождением онлайн-курсов. Но несмотря на это, для большинства опрошенных процесс выполнения практических заданий, являющийся ключевой функциональностью — в мобильном приложении не менее удобен, чем в браузере.

Группа вопросов 2. Пользовательский интерфейс

Вторая группа вопросов была связана с пользовательским интерфейсом мобильного приложения.

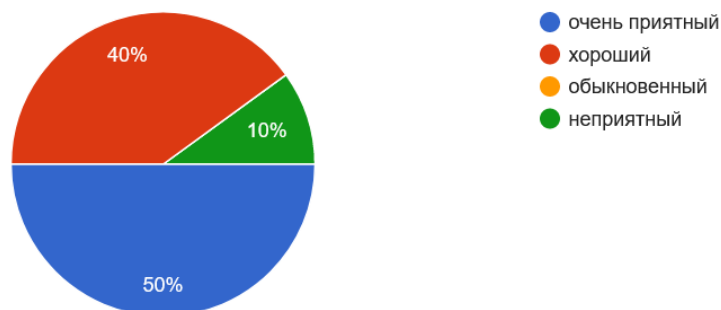


Рис. 12: Распределение ответов на вопрос: *“Как Вы оцениваете общий дизайн приложения (палитра цветов, стили заголовков, надписей и прочего текста и др.)?”*

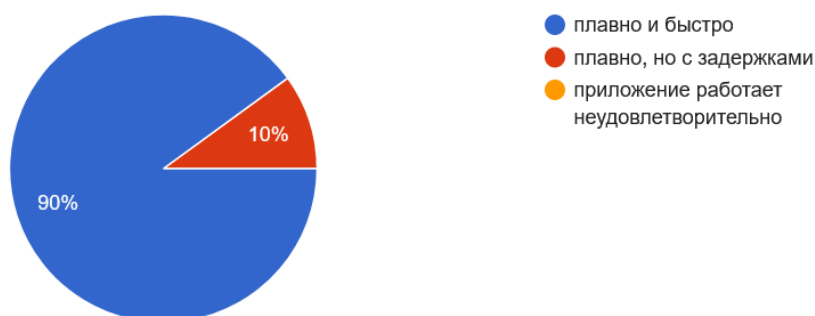


Рис. 13: Распределение ответов на вопрос: *“Как Вы оцениваете плавность и отзывчивость интерфейса (например, при вертикальном пролистывании контента, переходе между экранами)?”*

Практически все учащиеся оценили интерфейс как *“очень приятный”* или *“хороший”*, а также отметили его плавность и отзывчивость (Рис. 12, Рис. 13).

Группа вопросов 3. Используемая функциональность

Третья группа вопросов была связана с функциональностью мобильного приложения.

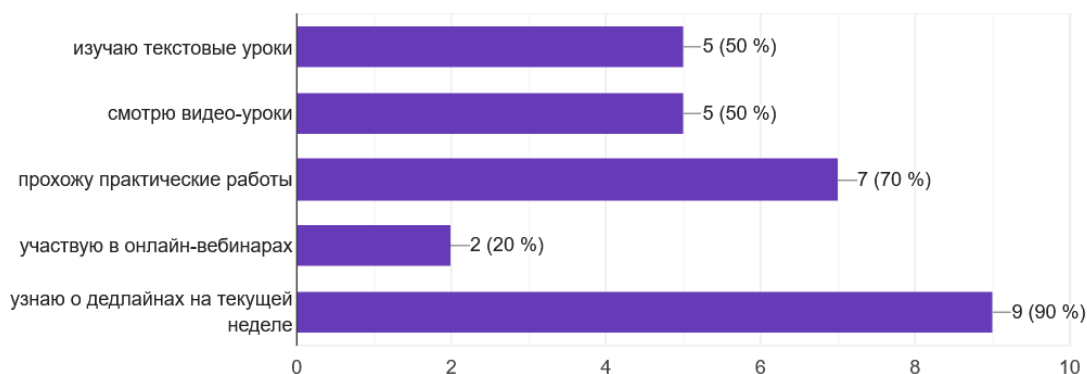


Рис. 14: Распределение ответов на вопрос: “Какие задачи Вы выполняете преимущественно в мобильном приложении?”



Рис. 15: Распределение ответов на вопрос: “Что бы Вы хотели видеть в ближайших обновлениях больше всего?”

Согласно ответам на первый вопрос (Рис. 14) мобильное приложение чаще всего использовалось для того, чтобы узнать, какие задачи необходимо выполнить на текущей неделе. Участие же в онлайн-вебинарах оказалось наименее популярным случаем использования среди опрошенных.

Также наиболее желаемой функциональностью в ближайших обновлениях оказалось добавление страницы с практическими работами (Рис. 15). Данная функциональность позволит облегчить поиск нужной работы для учеников.

5.3. Результаты апробации

В ходе проведённой апробации на учащихся онлайн-школ *Экзамис* и *Екзам.Онлайн* был выявлен и исправлен ряд проблем, связанных с отсутствием некоторой функциональности или неправильной работой мобильного приложения. В частности, были внесены изменения в отображение определённых компонентов интерфейса на страницах видеоуроков и вебинаров.

Согласно проведённому опросу для большинства респондентов использование мобильного приложения оказалось удобнее по сравнению с веб-версией, также была проявлена заинтересованность опрошенных в дальнейшем развитии приложения и его функциональности. В среднем пользовательский интерфейс был оценен как *“хороший, плавный и отзывчивый”*. По результатам опроса в следующем крупном обновлении планируется добавить страницу с практическими работами, за которую было отдано 50% всех голосов.

Таким образом, запуск тестовой версии приложения был проведен успешно и показал хорошие результаты. Были обнаружены и исправлены ряд проблем, а также поставлены задачи по дальнейшему развитию мобильного приложения в том направлении, которое желают пользователи.

Заключение

В ходе данной работы были получены следующие результаты.

1. Проведен обзор существующих решений на примере трёх популярных онлайн-школ по подготовке к ЕГЭ в России (Умскул, Фоксфорд.Учебник, Maximum ЕГЭ), а также выполнен обзор наиболее распространенных технических средств по кроссплатформенной мобильной разработке (Xamarin, React Native, Flutter).
2. Спроектирована архитектура мобильного приложения для прохождения онлайн-курсов (использованы шаблоны Business Logic Component и Repository), а также его пользовательский интерфейс (использована технология Figma).
3. Реализовано мобильное приложение с использованием кроссплатформенной технологии Flutter.
4. Создан инструмент для генерации графического представления “обогащенного текста” с использованием кроссплатформенной технологии Flutter. Доступ к данному инструменту предоставлен через следующий онлайн-ресурс https://bitbucket.org/tapcourse/tapcourse_rich_text.
5. Проведена апробация мобильного приложения на онлайн-школах *Экзамис* и *Екзам.Онлайн*. Проведено анкетирование среди учащихся онлайн-школы *Экзамис* (собрано 10 анкет). Обнаружено и исправлено несколько проблем. Поставлены задачи для дальнейшего развития и улучшения мобильного приложения.

Список литературы

- [1] Size of the global e-learning market in 2019 and 2026, by segment. — Access mode: <https://www.statista.com/statistics/1130331/e-learning-market-size-segment-worldwide> (online; accessed: 01.05.2021).
- [2] Официальный сайт онлайн-школы по подготовке к ЕГЭ Examis. — Access mode: <https://examis.ru> (online; accessed: 01.05.2021).
- [3] Официальный сайт онлайн-школы по подготовке к ЕГЭ Экзам.Онлайн. — Access mode: <https://class.examschool.online> (online; accessed: 01.05.2021).
- [4] Официальный сайт онлайн-школы по подготовке к ЕГЭ Unit Skills. — Access mode: <https://unitskills.ru> (online; accessed: 01.05.2021).
- [5] Global smartphone penetration rate as share of population from 2016 to 2020. — Access mode: <https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005> (online; accessed: 01.05.2021).
- [6] 10 Stats That Prove Mobile Learning Lives up to The Hype. — Access mode: <https://www.docebo.com/blog/10-stats-prove-mobile-learning-lives-up-to-hype> (online; accessed: 01.05.2021).
- [7] Власов Роман Маркович. Механики геймификации при разработке образовательной веб-платформы, анализ, внедрение и его результаты // Заметки по информатике и математике. — Ярославский государственный университет им. П. Г. Демидова, 2020. — Р. 37–44. — ISBN: 978-5-8397-1200-3. — Access mode: <https://www.elibrary.ru/item.asp?id=43163397>.

- [8] Mobile App Development Approaches Explained. — Access mode: <https://railsware.com/blog/native-vs-hybrid-vs-cross-platform> (online; accessed: 01.05.2021).
- [9] Официальный сайт фреймворка Xamarin, предназначенного для кроссплатформенной мобильной разработки на языке C#. — Access mode: <https://dotnet.microsoft.com/apps/xamarin> (online; accessed: 01.05.2021).
- [10] Официальный сайт фреймворка Xamarin.Forms, предназначенного для создания кроссплатформенного пользовательского интерфейса на языке C#. — Access mode: <https://dotnet.microsoft.com/apps/xamarin/xamarin-forms> (online; accessed: 01.05.2021).
- [11] Официальный сайт фреймворка React Native, предназначенного для кроссплатформенной мобильной разработки на языке JavaScript. — Access mode: <https://reactnative.dev> (online; accessed: 01.05.2021).
- [12] Официальный сайт фреймворка Flutter, предназначенного для мультиплатформенной разработки на языке Dart. — Access mode: <https://flutter.dev> (online; accessed: 01.05.2021).
- [13] Stack Overflow Developer Survey 2020. — Access mode: <https://insights.stackoverflow.com/survey/2020> (online; accessed: 01.05.2021).
- [14] A predictable state management library that helps implement the BLoC design pattern. — Access mode: <https://bloclibrary.dev> (online; accessed: 01.05.2021).
- [15] Официальный сайт онлайн-сервиса Figma, предназначенного для прототипирования пользовательского интерфейса. — Access mode: <https://www.figma.com> (online; accessed: 01.05.2021).

- [16] Официальный сайт онлайн-сервиса по размещению библиотек на языке Dart. — Access mode: <https://pub.dev> (online; accessed: 01.05.2021).
- [17] Официальный сайт онлайн-сервиса Google Play Console, предназначенного для публикации Android-приложений в Google Play. — Access mode: <https://play.google.com/console/about> (online; accessed: 01.05.2021).
- [18] Официальный сайт инструмента Codemagic, предназначенного для организации CI/CD мобильных приложений. — Access mode: <https://codemagic.io> (online; accessed: 01.05.2021).