

**ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»**

37523

регистрационный номер

Секция: Информационные технологии (ИУ7)

**СОЗДАНИЕ КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО
ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ XAMARIN. «НОВОСТНОЙ
АГРЕГАТОР»**

Автор: Глущенко Дмитрий Иванович
ГБОУ «инженерная школа №1581», 11 «И» класс

Работа выполнена без научного руководителя

СОЗДАНИЕ КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ XAMARIN. «НОВОСТНОЙ АГРЕГАТОР»

Глушенко Дмитрий Иванович

11-й «И» класс, Государственное бюджетное образовательное учреждение города Москвы «инженерная школа №1581»

Работа выполнена без научного руководителя

Аннотация

Сейчас, в эпоху развития информационных технологий, источники и потоки информации настолько велики, что поиск необходимого требует колоссальных затрат времени. Для упрощения поиска информации были придуманы новостные агрегаторы.

Одной из главных проблем новостных приложений является ограниченное количество доступных сайтов. При разработке своего приложения приоритетной задачей для меня стояло создание конструктора API, который позволит пользователям создавать собственные API в специальной вкладке приложения. Также пользователи смогут делиться созданными API внутри приложения.

Цель работы:

Разработать собственное кроссплатформенное мобильное приложение «новостной агрегатор», с открытым исходным кодом, которое будет иметь выгодные преимущества по сравнению с аналогами.

Задачи:

- Изучить различные платформы для разработки мобильных кроссплатформенных приложений.
- Научиться работать с серверными и локальными СУБД.
- Получить опыт разработки кроссплатформенных мобильных приложений.
- Отладить работу приложения на выбранных платформах.

Материалы и методы:

- Платформа для создания мобильных приложений Xamarin.
- IDE Microsoft Visual Studio.
- Локальная СУБД – SQLite.
- Backend - язык программирования C#.
- Frontend - язык разметки XAML.

Результаты:

Разработанное мной мобильное приложение позволяет пользователям получать информацию из различных интернет-ресурсов, сортировать её по категориям и времени, добавлять собственные интернет-ресурсы и делиться ими. Также приложение корректно работает на всех доступных платформах и имеет выгодные преимущества по сравнению с аналогами.

Содержание

1. Введение	4
2. Основная часть	6
3. Заключение	11
4. Список литературы	12
Приложение 1	13
Приложение 2	14
Приложение 3	15
Приложение 4	16
Приложение 5	17
Приложение 6	19

Введение

Актуальность и новизна:

В современном, стремительно развивающемся и меняющемся мире каждому человеку, независимо от его социального и материального положения, необходимо постоянно получать новую для себя информацию. Это является обязательным условием для личностного и профессионального роста. Сейчас, в эпоху развития информационных технологий, источники и потоки информации настолько велики, что поиск необходимого требует колоссальных затрат времени. Для упрощения поиска информации были придуманы новостные агрегаторы.

При разработке своего приложения главной целью для меня стояло создание конкурентноспособного приложения, которое за счет своего уникального функционала заняло бы сильную позицию в нише новостных приложений.

Одной из главных проблем новостных приложений, я выделил ограниченное количество доступных новостных источников. Чтобы решить данную проблему я реализовал конструктор API [1]. Конструктор позволит пользователям самостоятельно добавлять новые новостные ресурсы. Данный модуль вносит новизну в рынок новостных агрегаторов, и позволит расширить количество источников информации.

Цель работы:

Разработать собственное кроссплатформенное мобильное приложение, с открытым исходным кодом, которое будет иметь выгодные преимущества по сравнению с аналогами.

Задачи:

- Изучить платформы разработки мобильных кроссплатформенных приложений
- Изучить особенности и недостатки аналогичных приложений
- Научится работать с серверными и локальными СУБД
- Отладить работу приложения на выбранных платформах

- Получить опыт разработки кроссплатформенных мобильных приложений
- Получить опыт в проектировании ux/ui

Основная часть

1. Выбор программных средств и инструментов. Методы

1.1 Тип приложения

При создании приложения остро встает вопрос о его кроссплатформенности. Ведь каждому разработчику хочется максимально расширить аудиторию пользователей. Поэтому перед началом разработки решается вопрос будет приложение кроссплатформенным или нативным.

Преимущества кроссплатформенной разработки над нативной:

- Охват большей аудитории. Это происходит за счёт выпуска приложения одновременно на нескольких платформах
- Более низкие затраты при разработке. У компании нет необходимости нанимать разработчиков под конкретную платформу
- Увеличение скорости разработки. Это происходит за счет создания одного проекта

Но недостатков у кроссплатформенной разработки также достаточно:

- Низкая гибкость. При создании кроссплатформенных приложений разработчики создают единый интерфейс, и могут не учесть некоторые особенности платформ. Как следствие пользователи могут столкнуться с графическими ошибками
- Более низкая производительность. Многие разработчики считают, что сравнивать производительность кроссплатформенных и нативных приложений некорректно, так как нативные приложения в отличие от кроссплатформенных создаются под одну целевую платформу специалистами, знающими все нюансы данной платформы. Но все же если говорить какое из приложений будет производительнее, то это нативное

1.2 Платформа разработки

В качестве платформы разработки приложения была выбрана Microsoft Xamarin.

Xamarin – это платформа с открытым исходным кодом, предназначенная для разработки производительных кроссплатформенных мобильных приложений для IOS, Android и UWP.

Почему Xamarin?

- Поддерживает три платформы: IOS, Android и UWP
- Имеет удобный менеджер пакетов NuGet

1.3 Парсер

Для реализации парсера была использована библиотека с открытым исходным кодом AngleSharp. AngleSharp представляет из себя набор C# классов позволяющих анализировать html код интернет-ресурсов.

Почему AngleSharp?

- Библиотека постоянно развивается и обновляется
- Имеет демонстрационный проект, при помощи которого можно ознакомиться с основными элементами библиотеки
- Имеет неплохую производительность и скорость работы. Так как API построен на основе официальной спецификации по JavaScript DOM
- Использование CSS селекторов для навигации в DOM дереве

1.4 Локальная база данных

Для хранения пользовательских настроек была использована СУБД SQLite. SQLite это встраиваемая система, которая позволяет управлять базами данных. В отличие от многих СУБД, которые используют клиент-серверную архитектуру, SQLite использует устройство пользователя для хранения данных.

Почему SQLite?

- Не нужно учить отдельные языки запросов. Система представляет из себя набор C# классов, которые позволяют использовать большой набор SQL команд
- Это быстро и надёжно. Приложение взаимодействует с SQLite при помощи прямых вызовов файла без использования сторонних интерфейсов

1.5 Клиент-серверное взаимодействие

За клиент-серверную архитектуру отвечает сервис Firebase. Для хранения API сайтов используется Firebase Realtime Database, а для авторизации Firebase Authentication.

Почему Firebase?

- Имеет готовое решение для системы авторизации
- Предоставляет API для IOS и Android
- Есть сторонний API для Xamarin

1.6 Проектирование ux/ui

Перед разработкой пользовательского интерфейса создается макет, который затем будет служить ориентиром при верстке интерфейса. Макет нужен для оперативного внесения изменений в пользовательский интерфейс. Для его создания был использован графический редактор Figma.

2. Описание основных модулей. Техническое описание продукта

2.1 Парсер

Одним из ключевых модулей моего приложения является парсер. Парсер необходим для получения статей с интернет-ресурсов. Модуль состоит из трех основных классов: “HtmlLoader”, “ParserConstructor”, “StartParser”.

Класс “StartParser” получает настройки парсера, запускает “HtmlLoader” и “ParserConstructor”, после чего возвращает список полученных статей.

Класс “HtmlLoader” по ссылке на ресурс получает код сайта и по нему строит DOM дерево.

Класс “ParserConstructor” осуществляет поиск по DOM дереву с использованием CSS селекторов, затем составляет и возвращает список полученных статей. Структура парсера (Приложение 2).

2.2 Конструктор API

Конструктор является одним из главных преимуществ моего приложения. Он дает пользователям возможность самостоятельно добавлять новые интернет-ресурсы в приложение без участия разработчика.

Реализация:

Сам конструктор находится по пути Community/Constructor. Представляет из себя страницу с полями для ввода CSS селекторов и описания самого интернет-ресурса. Также на странице конструктора есть кнопка, которая перемещает пользователя на страницу отладки, необходимую для проверки работоспособности API [1]. После отладки и проверки пользователь публикует созданный API [1] в открытый доступ и добавляет его в свой личный список.

2.3 Клиент-серверная архитектура

За клиент-серверное взаимодействие отвечает сервис от Google “Firebase”.

2.3.1 Система авторизации

Авторизация проводится с помощью адреса электронной почты пользователя и пароля.

Реализация:

Сначала для каждой платформы (IOS, Android) генерируется Json файл, который содержит информацию о проекте и API [1] ключи для взаимодействия приложения с сервисом Firebase. Для каждого из решений (IOS, Android) определяется класс взаимодействия, который реализует методы входа в аккаунт и создания новой записи в базе данных. Эти классы регистрируются с общим кроссплатформенным кодом.

2.3.2 Хранение созданных API

При разработке приложения у меня появилась идея дать пользователям возможность делиться созданными API [1]. Для ее реализации был использован сервис Firebase “Realtime Database”.

Реализация:

Был создан класс взаимодействия приложения с базой данных, а также сформирована структура данных. Класс взаимодействия реализует методы удаления, получения, сохранения элементов базы данных. База данных обновляется в режиме реального времени, а также использует NoSQL архитектуру и хранит данные в виде ключ-значение.

2.4 Локальная база данных

В локальной базе данных хранится список выбранных интернет-ресурсов (API). Информация о пользователе, статусе авторизации, настройках новостной ленты хранится в свойствах приложения.

Реализация:

При первом запуске приложения создается файл, к которому будет посылать запросы класс взаимодействия. Подключение к файлу происходит с использованием паттерна проектирования “Singleton”. Класс взаимодействия реализует набор методов: удаление/сохранение/изменение элемента, получение всех элементов.

2.5 Структура приложения

Первое с чем сталкивается пользователь при запуске приложения — это страница авторизации, откуда после входа или создания аккаунта он попадает на главную страницу. Всего в приложении существуют три вкладки: “Настройки”, “Сообщество”, “Новости”. Блок схема структуры приложения и скриншоты интерфейса (Приложение 3, 4).

Во вкладке настройки пользователь может посмотреть свой профиль, изменить список выбранных новостных ресурсов и настроить отображение статей.

Во вкладке сообщество пользователь может посмотреть и добавить в свой личный список созданные другими пользователями API [1], воспользоваться поиском, а также перейти на страницу конструктора.

Вкладка новости представляет из себя ленту новостных статей, а также содержит фильтры и сортировки.

Сравнение с аналогами:

На сегодняшний день существует много новостных агрегаторов, все они имеют схожий функционал. Одни позволяют самостоятельно составлять ленту, другие предлагают удобный интерфейс и большое количество интернет-ресурсов. В своем приложении я постарался совместить многие функции в одном месте, а также добавить свои уникальные. Таблица сравнения (Приложение 5).

Заключение

Разработанное мной мобильное приложение позволяет пользователям получать актуальную информацию из различных интернет-ресурсов, сортировать её по категориям и времени, создавать собственные API [1] и делиться ими в специальной вкладке. Также приложение корректно работает на всех доступных платформах и имеет выгодные преимущества по сравнению с аналогами. Код разрабатываемого приложения доступен для всех желающих на моем GitHub (приложение 6).

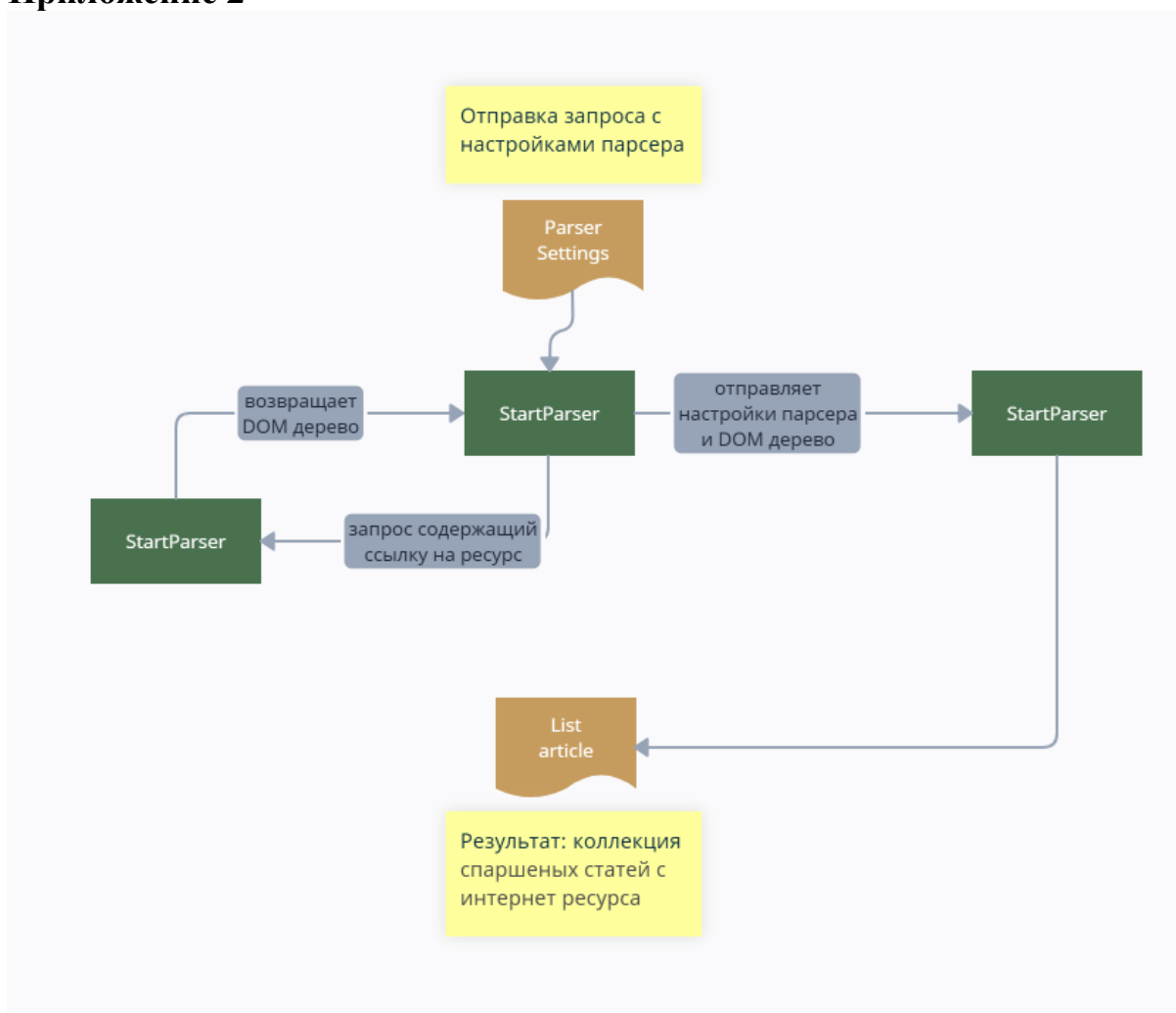
Список используемой литературы

1. Документация по SQLite // URL: <https://learn.microsoft.com/ru-ru/xamarin/xamarin-forms/data-cloud/data/databases#create-a-database-access-class> (дата обращения: 1.02.2024)
2. Документация по библиотеке AngleSharp // URL: <https://anglesharp.github.io/> (дата обращения: 1.02.2024)
3. Документация по C# // URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 1.02.2024)
4. Документация по CSS селекторам // URL: https://developer.mozilla.org/ru/docs/Learn/CSS/Building_blocksSelector (дата обращения: 1.02.2024)
5. Документация Xamarin forms // URL: <https://learn.microsoft.com/ru-ru/xamarin/xamarin-forms/> (дата обращения: 1.02.2024)
6. Документация по Firebase // URL: <https://firebase.google.com/docs?hl=en> (дата обращения: 1.02.2024)

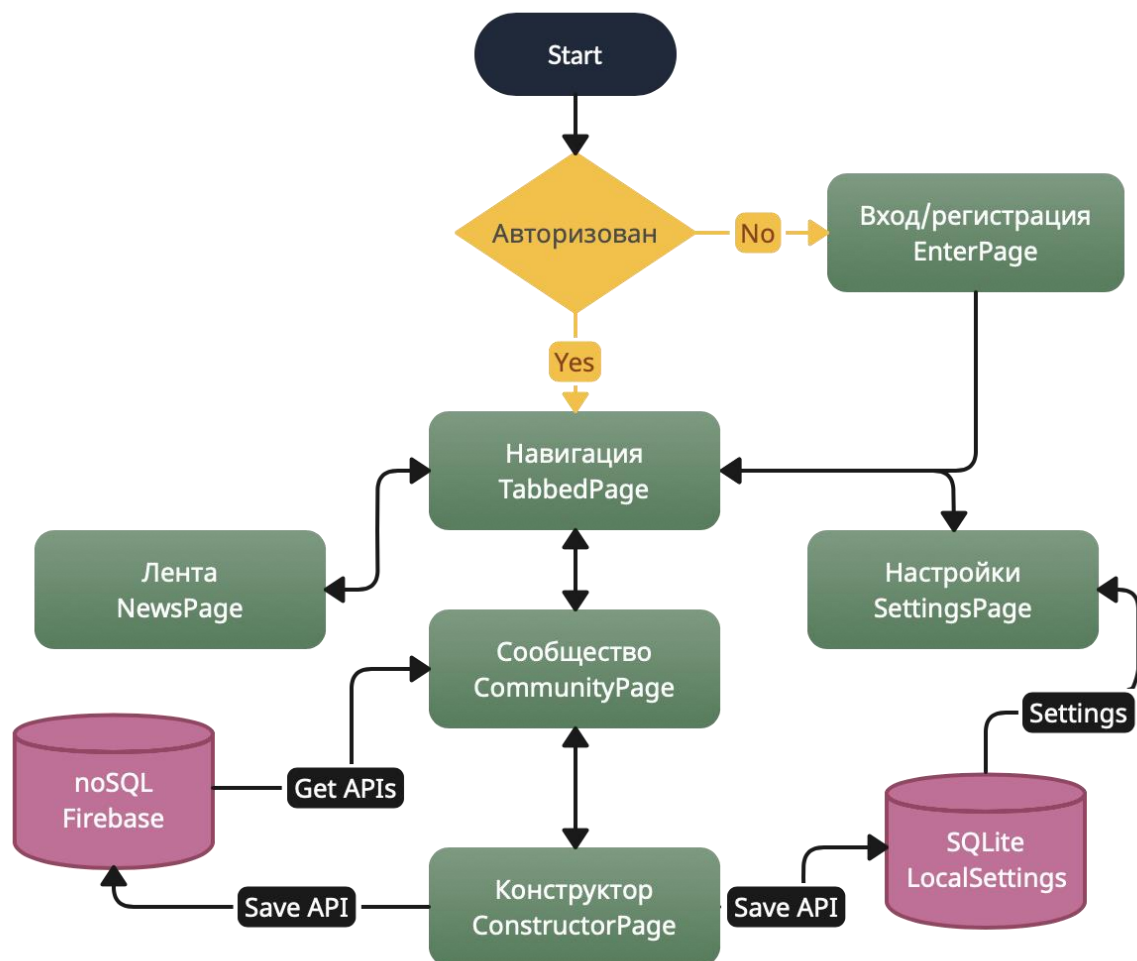
Приложение 1

API - Application Programming Interface.

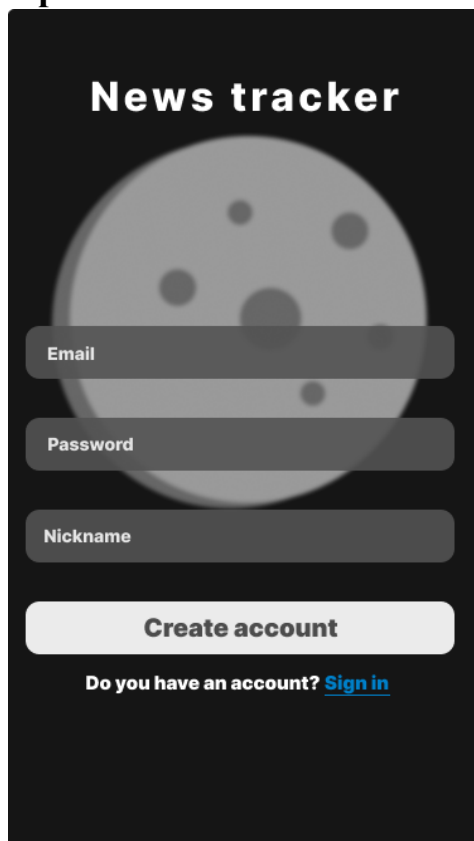
Приложение 2



Приложение 3



Приложение 4



News tracker

Email

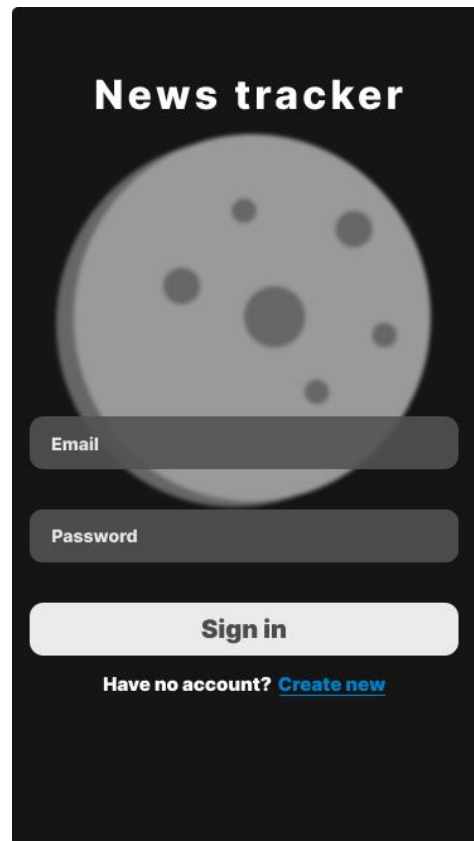
Password

Nickname

Create account

Do you have an account? [Sign in](#)

Страница регистрации



News tracker


Email

Password

Sign in

Have no account? [Create new](#)

Страница авторизации



Конструктор

Поиск

Barazded / Investing.com

Финансовая платформа и новостной веб-сайт; один из трех крупнейших глобальных финансовых веб-сайтов в мире. Он предлагает рыночные котировки, информацию об акциях, фьючерсах, опционах, аналитические материалы, сырьевые товары и экономический...

economic

Подробнее

Barazded / Ingromania.ru

«Игромания» — российское мультиплатформенное издание о видеоиграх: изначально ежемесячный журнал, а впоследствии и сайт. Печатный журнал издавался с сентября 1997 по декабрь 2018 года издательским домом «Техномир»...

games

Settings Community News

Страница Community



Всё

Уолл-стрит упала из-за неопределенности относительно снижения ставки

Investing.com — Американский рынок акций упал во вторник, поскольку надежды на скорое снижение ставки Федеральной резервной системой ослабли, но гигант розничной торгов...

economic investing 20.02.2024

Подробнее

Курс евро превысил 100 рублей впервые с 9 января

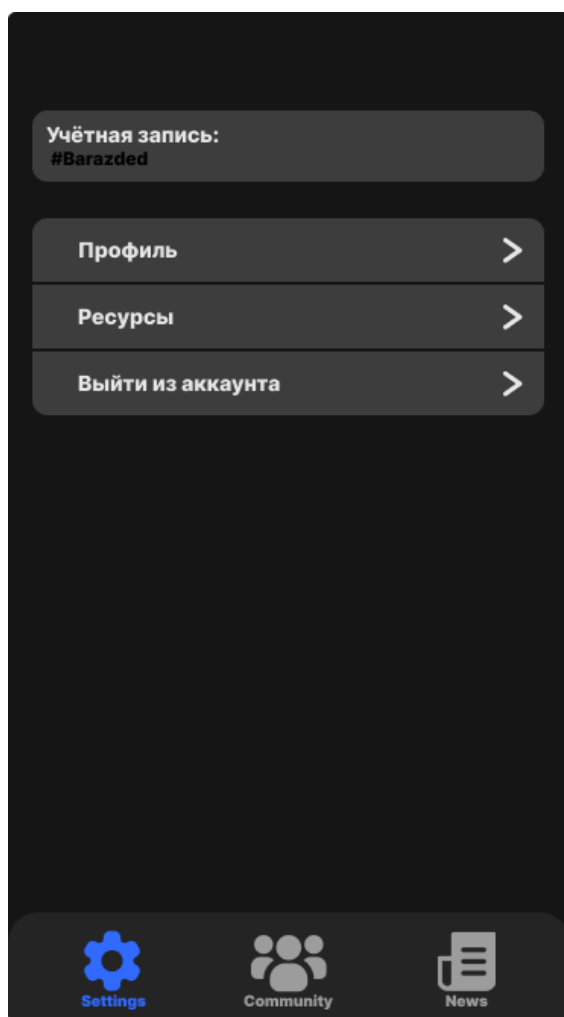
Investing.com — Курс евро вырос на 0,53%, или 0,52 руб., поднявшись до отметки 100 руб., свидетельствуют данные на Московской бирже по состоянию на 17:49 МСК. На отметке 100...

economic investing 20.02.2024

Подробнее

Settings Community News

Страница News



Страница Settings

Приложение 5

Для анализа использовались приложения: “NewsWorm”, “Яндекс.Дзен”, “Google.Новости”, “РБК”, “Pocket”.

Функция	Приложение аналог
Самостоятельное составление ленты	“NewsWorm”
Конструктор API (возможность добавлять свои интернет-ресурсы)	-
Новостная лента с фильтрами по разделам и сортировкой	“Яндекс.Дзен”, “Google.Новости”, “РБК”, “Pocket”
Межпользовательское взаимодействие (возможность делиться любимыми ресурсами)	“Pocket”

1. “NewsWorm” // URL: <https://apps.apple.com/ru/app/новости-by-newsworm/id1526352955>
2. “Яндекс.Дзен” // URL: <https://apps.apple.com/ru/app/дзен-видео-статьи-новости/id1343242452>
3. “Google.Новости” // URL: <https://apps.apple.com/ru/app/google-новости/id459182288>
4. “РБК” // URL: <https://apps.apple.com/ru/app/рбк-новости/id483524731>
5. “Pocket” // URL: <https://apps.apple.com/ru/app/pocket-будьте-в-курсе/id309601447>

Приложение 6

Исходный код проекта на GitHub // URL: <https://github.com/Barazded/Gazebo-apps>

