

Программные системы и вычислительные методы

Правильная ссылка на статью:

Зайков В.П., Прозоров П.Д. — Разработка мобильного приложения учебного расписания занятий студента // Программные системы и вычислительные методы. – 2023. – № 3. DOI: 10.7256/2454-0714.2023.3.39609 EDN: YTYIWB URL: https://nbpublish.com/library_read_article.php?id=39609

Разработка мобильного приложения учебного расписания занятий студента

Зайков Владимир Полиевктович

ORCID: 0000-0003-1641-9458

кандидат технических наук, доктор экономических наук

доцент, кафедра информационных систем и программирования, Кубанский государственный технологический университет

350010, Россия, г. Краснодар, ул. Московская, 2

✉ zaykov_2024@mail.ru



Прозоров Павел Дмитриевич

студент, кафедра информационных систем и программирования, ФГБОУ ВО Кубанский государственный технологический университет

350000, Россия, Краснодарский край, г. Краснодар, ул. Московская, 2

✉ pd.prozorov@yandex.ru



[Статья из рубрики "Языки программирования"](#)

DOI:

10.7256/2454-0714.2023.3.39609

EDN:

YTYIWB

Дата направления статьи в редакцию:

13-01-2023

Дата публикации:

05-10-2023

Аннотация: Целью работы является разработка мобильного приложения, которое может предоставлять возможность создания, редактирования учебного расписания. Предметом разработки является собственное мультиплатформенное мобильное приложение с использованием фреймворка Flutter. При разработке использовался архитектурный паттерн MVC, хранение данных было реализовано с помощью СУБД SQLite. Методология

работы базируется на Scrum-методе, позволяющем разбить рабочий процесс на спринты, каждый из которых позволил сделать вывод о целесообразности реализации принятого решения. Функциональный анализ, как основной метод, позволил выявить базовые требования для удобного использования программы: просмотра расписания по датам календаря, автозаполнение полей, возможность создавать, редактировать и удалять учебную информацию. Для реализации было выбрано гибридное решение. Собственное мультиплатформенное мобильное приложение было разработано с использованием фреймворка Flutter. При разработке был использован архитектурный шаблон MVC. Хранение данных было реализовано с использованием СУБД SQLite. При проектировании базы данных были определены следующие объекты: предмет, учитель, аудитория, тип урока, время, дата, дата урока. Разработанное приложение готово к публикации в AppStore и Google Play. Результатом его использования станет улучшение коммуникации с обучаемыми, ускорение процессов создания и изменения расписания, а также сокращение времени на загрузку нужных данных.

Ключевые слова:

расписание студента, электронное расписание, мобильное приложение, фреймворк flutter, база данных, субд sqlite, хранение данных, архитектурный паттерн MVC, мультиплатформенное решение, архитектурный шаблон

Введение

В настоящее время бурное развитие информационных технологий и изменения в системе образования приводят к необходимости внедрения информационных технологий во все сферы образования, в том числе для поддержки учебного процесса. Одной из главных проблем в процессе обучения является информированность студентов об учебном графике. Студенты должны иметь возможность в любое время просматривать актуальный график занятий. Ключ к решению данной проблемы – электронное расписание.

Передовые образовательные учреждения уже позволяют студентам смотреть учебное расписание на веб-сайте. Такое расписание удобно тем, что доступно для любых устройств, поддерживающих выход в сеть Интернет, однако в нем есть и недостатки, такие как вероятная недоступность ресурса и ограниченный функционал (например, отсутствие возможности смотреть график на определенный день календаря). Более того, остается открытым вопрос, что делать студентам образовательных учреждений, в которых нет электронного расписания.

Помимо веб-сайта, существует другой, более удобный способ отображения расписания в электронном виде – мобильное приложение. Преимущества данного подхода: лучшее взаимодействие с пользователем путем предоставления расширенных настроек, быстрая загрузка данных, а также возможность их кэширования. [\[1\]](#)

На основании вышеизложенного автором было решено реализовать собственное мобильное приложение учебного расписания, которое могло бы удовлетворить нужды большинства студентов.

Анализ существующих решений

Чтобы понять, какой базовый функционал требуется для удобства пользователя в приложении расписания, далее будут проанализированы некоторые уже существующие

решения. Сперва будут рассмотрены реализации электронного расписания в нескольких вузах Краснодар [\[2-4\]](#):

1). КубГУ. Расписание представлено в виде таблиц Excel. Для каждого факультета таблицы составляются по-разному. Такой формат не слишком удобен в использовании. Есть возможность фильтровать информацию, используя, к примеру, формулы и методы поиска в Excel, но это не подходит для неопытных пользователей.

2). КубГАУ. Расписание содержится на сайте университета. Отображается дата обновления расписания. Есть возможность искать занятия по группе и по аудитории. Для удобного ввода аудитории есть выпадающий список, появляющийся при наборе двух и более символов.

3). КубГТУ. Расписание содержится на сайте. Для его отображения требуется заполнить следующие данные: «Институт/Факультет», «Курс», «Группа». Для облегчения ввода данных присутствуют выпадающие списки с доступными вариантами. Есть возможность просмотра расписания не только на текущий, но и на прошлые семестры.

Далее будут проанализированы некоторые популярные мобильные приложения [\[5-7\]](#):

1). Приложение «Skedy». Расписание представляется в виде бесконечного списка «карточек», каждая из которых отображает занятия, прикрепленные к определенной дате. Можно выбрать дату, начиная с которой будет отображаться расписание, по умолчанию это сегодняшний день. Занятие является совокупностью следующих данных: предмет, преподаватель, аудитория, цвет, тип, время (номер, начало, конец), даты. На странице создания/редактирования занятия есть функционал автозаполнения полей для быстрой вставки данных, использующихся в других занятиях. В приложении можно добавлять заметки, которые привязываются к определенным датам и могут быть сгруппированы по папкам. Есть возможность делиться расписанием с группой с помощью случайно сгенерированного кода, что доступно после авторизации через Google аккаунт.

2). Приложение «Расписание для студентов». Расписание отображается на неделю, поскольку рассчитывается, что у пользователя расписание постоянно и разделяется на некоторое количество чередующихся недель (например, две недели – четная и нечетная). Можно отобразить расписание на определенный день календаря. Занятие является совокупностью следующих данных: предмет, преподаватель, аудитория, цвет, тип, время (номер, начало, конец), даты. На странице создания/редактирования занятия есть функционал автозаполнения полей. В приложении есть возможность добавлять не только занятия, но и домашние задания, заметки, а также оценки. Присутствует раздел «Списки», где можно просматривать, редактировать и добавлять различные данные, разделенные по таблицам БД: предметы, преподаватели, аудитории, домашние задания и т.д. Предусмотрена возможность делиться своим расписанием, используя сохранение и загрузку данных в JSON формате.

3). Приложение «Кампус». В данном приложении, в отличие от рассмотренных выше, расписание не составляется вручную, а подгружается с сервера. На данный момент порядка шестидесяти учебных заведений разных городов присоединились к данному проекту. Расписание отображается по дням недели, с возможностью перехода к определенному дню календаря. В карточке занятия показывается номер, время, название предмета, тип, номер аудитории или место проведения и ФИО преподавателя. Приложение позволяет смотреть и оставлять отзывы о преподавателях, добавлять в ленту новостей анонимные посты.

В ходе анализа автором был выявлен функционал, который может быть добавлен в мобильное приложение расписания. Рассмотренные программы имеют свои интересные особенности, которые можно учесть при собственной реализации – например, достаточно удобное отображение расписания по датам в «Skedy» и управление данными таблиц БД через раздел «Списки» в приложении «Расписание для студентов».

Проектирование приложения

Когда нет доступа к базе данных учебного заведения, есть три основных способа реализации расписания в мобильном приложении:

1. Парсинг веб-страницы расписания. В данном случае при загрузке расписания пользователем будут происходить автоматический сбор и структурирование данных с веб-страницы, на которой размещено расписание. Этот подход позволяет не задумываться о добавлении и обновлении расписания в приложении, но достаточно сложен в разработке и предполагает сильную зависимость от кода верстки страницы с расписанием. Кроме того, не стоит полагаться на быструю скорость работы такого приложения. [\[8\]](#)

2. Использование удаленной базы данных. Такой подход предполагает наличие сервера, на котором располагается БД расписания. Приложение получает данные с удаленного сервера и кэширует их. Базу данных обновляет время от времени разработчик. Это позволяет не зависеть от верстки веб-страницы расписания, и пользователям при этом не придется добавлять и обновлять график занятий, однако эту ношу на себя берет разработчик.

3. Использование локальной базы данных. При таком варианте база данных расписания находится на стороне клиента. Преимущества данного подхода в том, что разработчику не приходится задумываться об обновлении расписания, так как пользователь самостоятельно вносит для себя все нужные корректировки. Также это позволяет пользователю составлять собственный график занятий, и вдобавок пользоваться электронным расписанием студентам тех учебных заведений, которые не предоставляют его. Однако самостоятельное внесение данных пользователем также можно рассматривать и как недостаток, поскольку он при этом тратит свое время. В таком случае было бы неплохо добавить возможность делиться расписанием с другими людьми посредством экспорта и импорта данных, либо загружая локальные данные на удаленный сервер.

Авторами было решено реализовать доступ к расписанию с помощью локальной базы данных, поскольку этот способ наиболее универсален и прост в разработке.

Для реализации приложения учебного расписания был выбран набор средств мобильной разработки Flutter, который состоит из двух частей: SDK и Framework. SDK – это набор инструментов для компиляции кода в нативном машинном коде. Framework – это коллекция стандартных элементов пользовательского интерфейса (виджетов). Flutter позволяет создавать мультиплатформенные мобильные приложения на объектно-ориентированном языке Dart, который направлен на развитие веб-разработки и позиционируется в качестве альтернативы Javascript. [\[9\]](#)

Для хранения данных в приложении будет использоваться СУБД SQLite. Это реляционная база данных, совместимая с SQL. Вся логика SQLite встроена в программу в виде библиотеки C, а все данные БД хранятся в одном файле. Таким образом, данная СУБД

работает везде, где есть поддержка C и постоянного хранилища данных. [10]

Спроектированная структура базы данных представлена на рис. 1.

Разработка приложения будет осуществляться с использованием архитектурного паттерна MVC, который предлагает при организации кода выделять блоки, отвечающие за решение разных задач, а именно: model (данные приложения), view (внешний вид), controller (контроль работы приложения) [11]. В связи с некоторыми трудностями реализации MVC в Flutter, было принято решение использовать библиотеку mvc_pattern [12].

Для запросов к базе данных будет использоваться библиотека sqfentity [13], которая позволяет автоматически сгенерировать классы для всех сущностей БД и затем легко выполнять SQL команды при помощи методов-цепочек.

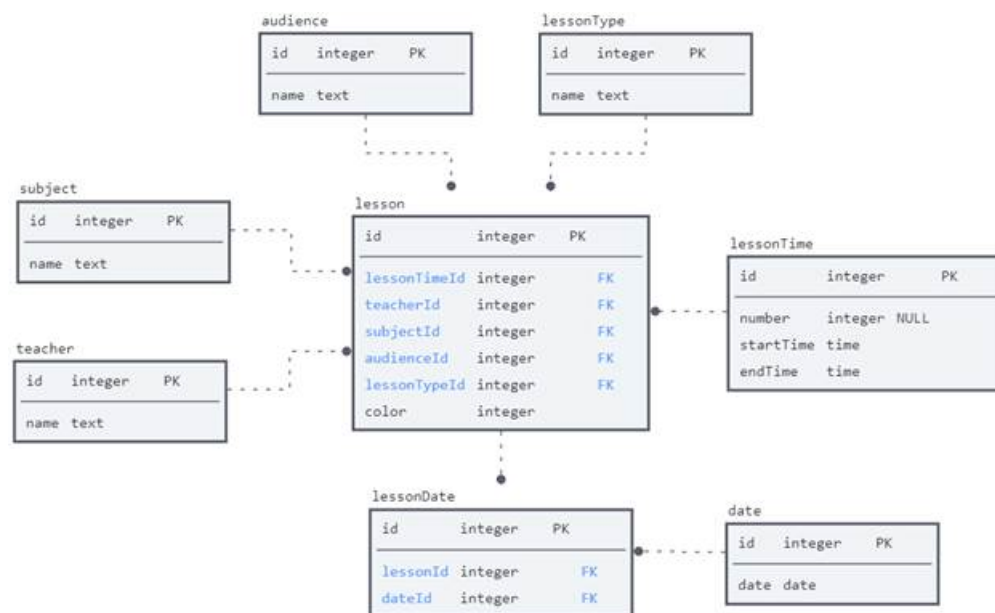


Рис. 1 – Структура базы данных приложения

Figure 1 – Application database structure

При проектировании интерфейса использовались идеи из приложений, рассмотренных выше. В частности, в «Skedy» автору показалось удобным вертикальное пролистывание расписания, а также страница редактирования занятий и страница добавления занятия. А в «Расписании для студентов», по мнению автора, весьма удобно нижнее навигационное меню и раздел, через который можно осуществлять управление различными данными расписания. Также были выслушаны пожелания нескольких опрошенных людей по поводу расположения некоторых элементов интерфейса. На этой основе было принято решение на странице расписания в верхнем баре приложения добавить кнопку для выбора даты, для которой нужно показать расписание, а также кнопку для перехода к редактированию расписания.

Реализация приложения

Страница расписания реализована в виде бесконечного списка карточек по датам (рис. 2, а), каждая из которых отображает все занятия, прикрепленные к соответствующему дню. Данный список был реализован с помощью конструктора ListView.builder. Пользователь может выбрать день, начиная с которого будет отображаться расписание. Для этого требуется нажать на иконку календаря справа сверху, после чего появится

окошко с выбором даты (рис. 2, б).

Для реализации этого был использован виджет SfDateRangePicker из библиотеки syncfusion_flutter_datepicker [14]. Этот виджет позволяет пролистывать даты вертикально, а также, что наиболее важно, выбирать несколько дат одновременно (понадобится на странице редактирования занятия для установки дней, в которые оно проводится).

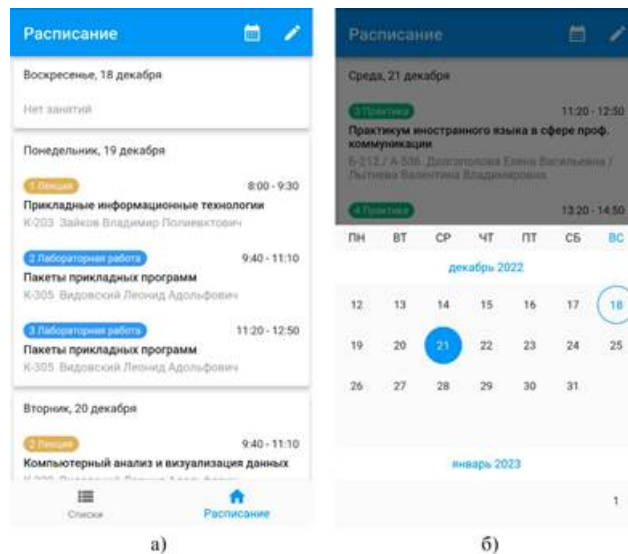


Рис. 2 – Страница расписания (а) и окошко с выбором даты, на которую требуется отобразить расписание (б)

Fig. 2 – The schedule page (a) and a window with a choice of the date on which you want to display the schedule (b)

Для внесения изменений в расписание пользователю требуется нажать на кнопку с иконкой карандаша справа сверху. Откроется страница со всеми существующими занятиями, разделенными по дням недели (рис. 3, а). Информацию о занятиях можно редактировать, дублировать и удалять, используя контекстное меню, которое вызывается путем долгого нажатия на карточку занятия. Данное меню было реализовано с помощью функции `showMenu`.

Для добавления нового занятия необходимо нажать на круглую кнопку с иконкой плюса. После этого отобразится страница для ввода данных (рис. 3, б). При вводе данных пользователю будут предлагаться значения, уже вводимые ранее, т.е. существующие в базе данных. Для реализации данного функционала использовался стандартный виджет `RawAutocomplete`.

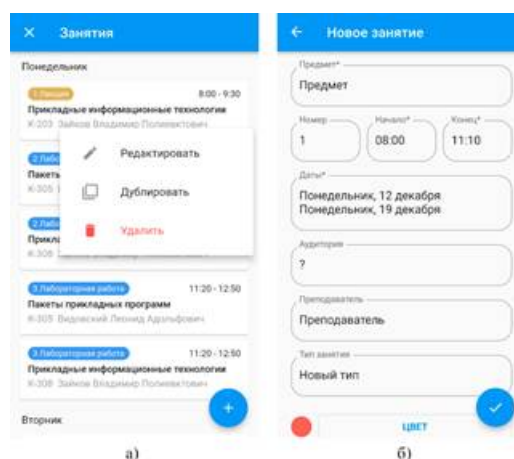


Рис. 3 – Страница «Занятия» и контекстное меню (а); страница для добавления занятия (б)

Fig. 3 – The "Classes" page and the context menu (a); the page for adding classes (b)

Через страницу «Списки» (рис. 4), доступную из нижнего навигационного бара, можно редактировать, удалять и создавать: занятия, предметы, преподавателей, аудитории, время и типы.

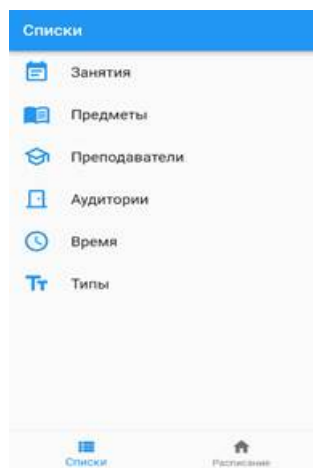


Рис. 4 – Страница, через которую можно перейти к управлению различными данными расписания

Fig. 4 – The page through which you can go to the management of various schedule data

Выводы

В статье была рассмотрена проблема создания мобильного приложения для просмотра и редактирования учебного расписания. Практическая значимость разработки приложения состоит в возможности его применения студентами для повышения своей организованности в процессе учебы. Был проведен анализ готовых решений, разработана и представлена собственная реализация для решения рассмотренной проблемы.

Разработанное приложение готово к публикации в AppStore и Google Play. Есть возможность его расширения путем добавления раздела «Настройки» с функционалом для изменения цветовой схемы, а также сохранения и загрузки расписания. Можно также добавить возможность просматривать занятия, привязанные к определенному предмету, преподавателю, времени и т.п.

Предложенное решение учитывает алгоритм составления расписания и внесения в него изменений. Созданный продукт может быть применим к расписанию практически любого вуза, хотя и основывается на механизмах функционирования расписания в Кубанском государственном технологическом университете.

Библиография

1. Дубовик Е.С. Применение мобильных приложений в сбытовой деятельности торговых предприятий // 57-я научная конференция аспирантов, магистрантов и студентов БГУИР. 2021. С. 214-215.
2. КубГУ [Электронный ресурс]. URL: <https://www.kubsu.ru/> (дата обращения: 07.12.2022).

3. Расписание занятий – Кубанский государственный аграрный университет [Электронный ресурс]. URL: <https://s.kubsau.ru/> (дата обращения: 07.12.2022).
4. Расписание занятий – Кубанский государственный технологический университет [Электронный ресурс]. URL: <https://elkaf.kubstu.ru/timetable/default/time-table-student-fof> (дата обращения: 07.12.2022).
5. Skedy-расписание занятий УГНТУ [Электронный ресурс]. URL: <https://skedy.ru/> (дата обращения: 07.12.2022).
6. Приложения в Google Play – Расписание для студентов [Электронный ресурс]. URL: <https://play.google.com/store/apps/details?id=com.bezgrebelnygregory.timetableforstudents> (дата обращения: 09.12.2022).
7. Приложения в Google Play – Кампус-Расписание занятий [Электронный ресурс]. URL: <https://play.google.com/store/apps/details?id=ru.dewish.campus> (дата обращения: 09.12.2022).
8. Скрыпников А.В., Денисенко В.В., Берестовой А.А. Парсинг данных для мобильного приложения университета // Вестник Воронежского института ФСИИ России. 2022. № 2. С. 126-132.
9. Калиневич Н., Гильванов Р.Г. Разработка кросс-платформенных приложений на языке Dart при помощи фреймворка Flutter // Интеллектуальные технологии на транспорте. 2021. № 4 (28). С. 21-26.
10. Ткачук Е.О. Применение СУБД sqlite в образовательном процессе современного вуза // Научные исследования высшей школы : сборник статей IV Международной научно-практической конференции, Пенза, 10 сентября 2020 года. – Пенза: "Наука и Просвещение" (ИП Гуляев Г.Ю.), 2020. С. 12-15.
11. Клеблеев Ш.А., Ибрагимов О.Ю. Веб-приложение с использованием технологии MVC // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. № 1 (19). С. 53-59.
12. mvc_pattern | Flutter Package [Электронный ресурс]. URL: https://pub.dev/packages/mvc_pattern (дата обращения: 10.12.2022).
13. sqfentity | Flutter Package [Электронный ресурс]. URL: <https://pub.dev/packages/sqfentity> (дата обращения: 11.12.2022).
14. syncfusion_flutter_datepicker | Flutter Package [Электронный ресурс]. URL: https://pub.dev/packages/syncfusion_flutter_datepicker (дата обращения: 11.12.2022).

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Рецензируемая статья посвящена задаче создания приложения для решения задач поиска данных по заданным критериям применительно к образовательному процессу. Авторы показывают актуальность перехода с веб-интерфейса в мобильное приложение для обеспечения оперативного доступа к данным, в т.ч. при отсутствии подключения. Рассматриваются подходы к проектированию структуры приложения, объяснен выбор доступа, перечислены элементы структуры и основные возможности приложения. Недостатком статьи является анализ только 2 аналогов и отсутствие обоснования выбора элементов UX-дизайна. С точки зрения подхода к проектированию приложения не отражены использованные качественные исследования, результаты тестирования

приложения, отзывы пользователей или анализ ошибок и внесенные изменения.

Собственное исследование Автора в статье не отражено, научную новизну определить затруднительно. Статья посвящена решению практической задачи.

Стиль изложения. В статье в необходимой мере используется профессиональная терминология, стилистические и смысловые ошибки не выявлены. Имеются иллюстрации, отражающие основные возможности приложения.

Структура статьи отвечает требованиям к публикации.

Библиография содержит 11 источников, из них 2 зарубежных нерцензируемых изданиях, в отечественных рецензируемых журналах - 2. Ссылки по тексту имеются.

Замечания.

В обзоре аналогов необходимо рассмотрение существующих приложений дополнить сведениями о форме представления расписания различными вузами (из открытых источников), например, наличие только статичного расписания на сайте, наличие расписание с возможностью поиска группы/аудитории, наличие и расписания, и мобильного приложения у вуза. Критерием выбора может стать город, в котором расположен вуз Авторов, или выборка по вузам аналогичного профиля и т.п.

Если Автор проводил опрос студентов или преподавателей с целью выявления требований к приложению, необходимо отразить результаты. Возможно Автор проводил опрос с целью выявления способа доступа к расписанию (сайт или приложение). Необходимо обосновать, почему именно мобильное приложение явилось объектом разработки.

Не ясно, при проектировании интерфейса каким образом Автор создавали его структуру? (выбор элементов, взаимное расположение, количество экранов). Необходимо отметить основные критерии в разделе Проектирование приложения.

Если автором выполнялось АВ-тестирование, необходимо привести его результаты (основные рассчитанные критерии). Крайне желательно отметить какие элементы вызвали трудности у пользователей, какие изменения были внесены.

Подписуночные надписи необходимо конкретизировать. На рис. 2а указаны фамилии преподавателей, необходимо проверить данные на соблюдение конфиденциальности.

Подписуночные надписи дублируются на английском языке, статья русскоязычная, поэтому если это не обусловлено требованиями журнала, необходимо убрать лишний текст.

На рис. 3а показано выплывающее меню, оно в значительной мере перекрывает ширину основного окна. Вместе с тем, оно показывает как минимум наличие двух уровней доступа (пользователь и администратор), что не отражено в тексте статьи.

Заключение необходимо переделать, отразив основное практическое значение разработки и возможность расширения функционала. Следует избегать формулировок «можно было бы» в научной статье.

Библиографию необходимо оформить в соответствии с требованиями ГОСТ и Журнала.

Статья нуждается в доработке, после которой может быть опубликована. В представленном виде может быть опубликована как краткое сообщение без повторного рецензирования с техническими правками, при устранении замечаний выше – как научная статья.