

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»

1223

регистрационный номер

ИУ «Информатика и системы управления»

название факультета

ИУ7 «Программное обеспечение ЭВМ и информационные технологии»

название кафедры

Микросервисное API в операционной деятельности учебного заведения.

Мобильное приложение «Мой МГТУ».

название работы

Автор:

Козодой Андрей Александрович

фамилия, имя, отчество

ГБОУ школа №1580, 11-Д класс

наименование учебного заведения, класс

Научный руководитель:

Митрофанов Михаил Сергеевич

фамилия, имя, отчество

ГБОУ школа №1580

место работы

учитель информатики

звание, должность

подпись научного руководителя

Москва – 2021

Микросервисное API в операционной деятельности учебного заведения. Мобильное приложение "Мой МГТУ".

Аннотация

EduApp API – наименование web-ориентированного программного интерфейса учебного заведения на основе микросервисной архитектуры, созданного с целью внедрения в учебный процесс собственных разрабатываемых технологий для его автоматизации; способствует внутреннему техническому развитию и обобщению знаний, внедрению этих знаний на практике. Цели проекта рассматривают возможность быстрого расширения и адаптации под нужды конкретной образовательной организации.

Проект создан для того, чтобы технологии стали открытыми и доступными для изучения и развития. Работа включает пример использования API в виде прототипа мобильного приложения «Мой МГТУ», которое представляет собой прообраз социальной сети и доступно для бета-тестирования в среде TestFlight.

Документ описывает архитектуру веб-приложения, а также дизайн и архитектуру мобильного приложения «Мой МГТУ».

В основу проектирования технического решения был заложен принцип согласования и обсуждения дизайна продукта и его функционала всеми участниками учебного процесса. Веб-приложение было разработано на языке программирования Python, мобильное приложение было разработано на языке программирования Swift. Для связи мобильного приложения с API был разработан модуль EduAppCore. Для реализации блока новостей и объявлений в виде «историй» был разработан модуль Warbler.

В текущей версии проекта, с помощью мобильного приложения, можно узнать расписание по классам, ученикам, учителям. Новости и объявления образовательного учреждения представлены в привычном по формату социальных сетей «историях».

Предусматривается дальнейшая работа над проектом, как дальнейшее исполнение плана, где глобальная миссия – предоставить / дополнить платформу высшего учебного заведения.

Оглавление

1	Введение.....	4
2	Философия продукта	5
3	Модель / Конструкторская часть	6
3.1	Применяемые тех решения	6
3.2	Формирование базы данных	9
3.2.1	Получение списка учителей.....	9
3.2.2	Получение списка учеников	10
3.2.3	Получение расписания	11
3.2.4	Преобразование собранных данных.....	12
3.3	Микросервисная архитектура.....	13
3.3.1	Возможные ошибки	14
3.3.2	Pagination.....	15
3.3.3	Фильтры	15
3.3.4	Создание ID пользователя.....	16
3.3.5	Статусы пользователей.....	16
3.3.6	Аутентификация.....	16
3.3.7	Версии	17
3.3.8	Ссылки	18
3.3.9	Другие проекты	19
3.3.10	Пользователи	20
3.3.11	Курсы.....	23
3.3.12	Stories.....	25
3.3.13	Группы	26
3.3.14	Расписание	26
3.4	Автоматизация исполнения скриптов	29
3.5	Библиотека EduAppCore	29
3.6	Библиотека pyEduApp	29
3.7	Библиотека Warbler	30
3.7.1	Описание реализации	30
3.7.2	Поддерживаемые жесты и управление	32
3.8	Push-уведомления	33
3.9	Наследники «UIViewController»	34
3.9.1	Контроллер «LoadingVC»	35
3.9.2	Контроллер «AdVC».....	35
3.9.3	Контроллер «LoginVC»	35
3.9.4	Контроллер «TimetableTVC»	36
3.9.5	Контроллер «LessonDetailsTVC».....	37

3.9.6	Контроллер «ProfileTVC»	37
3.9.7	Контроллер «SearchVC».....	38
3.9.8	Контроллер «UserTVC».....	39
3.9.9	Контроллер «SubgroupDetailsTVC»	41
3.10	Описание внутренних типов.....	41
3.10.1	Тип «User»	41
3.10.2	Тип «Status»	42
3.10.3	Тип «Class»	42
3.10.4	Тип «Subgroup»	42
3.10.5	Тип «Subject».....	42
3.10.6	Тип «Lesson».....	43
3.10.7	Тип «Day»	43
3.10.8	Тип «Week».....	43
3.10.9	Тип «LessonType»	43
3.10.10	Тип «Setting».....	43
3.10.11	Тип «OtherProject»	44
4	Технологическая и исследовательская часть	44
5	Заключение	45
6	Приложение: список литературы	45

Список иллюстраций

Рисунок 1. Системная архитектура «Мой МГТУ».....	7
Рисунок 2. Модель экранных форм в виде Storyboard.....	8
Рисунок 3. Информация о педагогическом составе	9
Рисунок 4. Информация об административном составе.....	10
Рисунок 5. Информация о расписании	11
Рисунок 6. Внутренняя структура расписания	12
Рисунок 7. Коммуникации в распределенной микросервисной архитектуре	13
Рисунок 8. Внешний вид контроллера Warbler	30
Рисунок 9. Управление уведомлениями.....	33
Рисунок 11. Задание времени напоминания	33
Рисунок 12. Внешний вид уведомления.....	33
Рисунок 13. Наследники «UIViewController»	34
Рисунок 13. Вид LoadingVC	35
Рисунок 14. Вид AdVC (темная тема)	35
Рисунок 15. Вид TimetableTVC	36
Рисунок 17. Вид LessonDetailsTVC	37
Рисунок 18. Вид LessonDetailsTVC (темная тема)	37
Рисунок 19. Вид ProfileTVC	38
Рисунок 20. Вид ProfileTVC (переход по ссылке).....	38
Рисунок 21. Вид SearchVC.....	39
Рисунок 22. Вид SearchVC (результаты).....	39
Рисунок 23. Вид UserTVC.....	40
Рисунок 24. Вид UserTVC (темная тема)	40
Рисунок 24. Вид SubgroupDetailsTVC	41

1 Введение

В процессе учебы в лицее 1580 у меня постоянно возникали мысли о том, что учебный процесс упускает возможность использования современных ИТ-технологий. Даже с введением электронного журнала, а также электронных учебников и электронных тетрадей (на основе планшета), постоянно оставалось ощущение «недостаточности».

Я поставил перед собой задачу сделать динамическую и универсальную платформу для образовательного учреждения, которую можно будет использовать в высших учебных заведениях, школах, кружках. После проведения опросов сверстников, анализа собранных материалов и сложившейся ситуации, была выявлена функциональность первой необходимости — это решение существующих проблем с динамически изменяемым расписанием занятий и поиск друзей из других классов / групп.

Для применения результатов проекта я начал рассматривать интересный для меня, как абитуриента, ВУЗ – МГТУ. Во время создания технических решений и продукта непосредственно, я принимал решения об адаптации платформы к нуждам образовательного учреждения совместно с его студентами.

Согласно их отзывам, действующие технологии, внедрённые в МГТУ, решают эти проблемы очень условно, где веб-сайт сервиса «Электронный университет» представляет информацию недостаточно современно и удобно для ежедневного использования и поиска. Возникла идея создания платформы, основанной на микросервисной архитектуре, которая бы обладала открытым API и позволяла масштабировать решение в дальнейшем, под быстроразвивающиеся нужды университета. В качестве демонстрации возможной функциональности возникла идея создания мобильного приложения «Мой МГТУ».

После завершения работ над веб-приложением и его применением, появились мысли о расширении – создание социальной сети на базе предоставляемых сервисов платформы. Однако, я понимал, что создание обычного клона социальной сети не представляется перспективным, так как необходимо сфокусироваться на «полезности», и я начал изучать интернет по теме моих рассуждений.

Моё изучение довело меня к теориям и практикам по части продвижения программных продуктов, формирования облика технического решения, синтеза программного продукта под требования, и т. д., где в самом итоге, исследование довело меня до источников – педагогике и проблемам воспитания абитуриентов. Тут я обнаружил, что современная педагогика не использует механизмы, действующие внутри виртуального социума, а также не учитывает современное противоречивое явление – социальные сети и виртуальное общение.

Мир меняется, меняется молодежь. Меняются взгляды молодежи на происходящую вокруг них реальность. Отсутствие достойных альтернатив социальным сетям, упускает возможность использования виртуальных коммуникаций в образовательном процессе.

Отсутствие достойной замены социальным сетям происходит по следующим причинам:

1. Социальные сети в жизни молодежи – состоявшийся факт. Более того, не учитывать этот факт в части образовательного процесса учебного заведения излишне расточительно
2. Клоны существующих социальных сетей не могут решить задачи педагогики, так как имеют иное (противоположное требуемому) назначение и задачи
3. Учебные заведения, обычно, не имеют достаточных ресурсов для создания собственной платформы социальных коммуникаций, а представляемые готовыми платформами возможности ограничены по функционалу и имеют ограниченную возможность расширения. Во всяком случае, имеющиеся возможности расширений, является достаточно сложными для использования в учебном процессе, так как требуют развитие конкретного продукта используемой платформы

Я уверен, что в рамках проекта, дополнительно, необходимо создать организационную структуру, которая позволила бы оперативно отслеживать любые ошибки у всех пользователей и незамедлительно вносить изменения. Для этого собираюсь использовать подходы *continuous delivery*¹ и *continuous integration*², которые обычно применяются для создания сервисов и крупных продуктов. Считаю, что с помощью подходов краудсорсинга³, именно они позволят тестировать внесенные исправления и реализовывать их буквально в течение одного часа.

2 Философия продукта

Проект ставит перед собой следующие общие задачи:

1. **Создать платформу учебного заведения.** Итоговый продукт, в первую очередь, должен обладать «полезной» для участников образовательного процесса функциональностью, т. е. использоваться многократно в течение дня. Также не следует упускать из виду современный дизайн и узнаваемость, присутствующих в популярных мобильных приложениях и социальных сетях.

¹ Непрерывная поставка (*continuous delivery*) — подход к разработке программного обеспечения, при котором все изменения, включая новые функции, изменения конфигурации, исправления ошибок и эксперименты, поставляются пользователям максимально быстро и безопасно

² Непрерывная интеграция (*continuous integration*) — практика разработки программного обеспечения, которая заключается в постоянном слиянии рабочих копий в общую основную ветвь разработки (до нескольких раз в день) и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем

³ Краудсорсинг (от англ. *crowdsourcing*, *crowd* — «толпа» и *sourcing* — «использование ресурсов») — это мобилизация на добровольных началах ресурсов большого количества людей с целью решения определенных задач

2. **Использовать коллективные ресурсы учебного заведения.** Web-API должен быть открыт для всех участников образовательного процесса. Таким образом, можно проводить образовательные практические занятия, результатом которых будет естественное расширение платформы учебного заведения. Студенты с опытом в области программирования могут воспользоваться личным идентификатором разработчика и начать работу над собственным продуктом на базе информационных ресурсов образовательного учреждения. Команды студентов смогут проводить самостоятельное тестирование, обсуждать инновации и дополнять конечную базу данных.

Для решения данных задач был выбран следующий план:

1. Создать Web-API с возможностью расширения функциональности и системой идентификаторов разработчиков
2. Создать библиотеки для «сторонних» разработчиков (Swift, Python)
3. Создать пример использования API в виде мобильного приложения «Мой МГТУ», решающего обозначенные проблемы (расписание и поиск)
4. Внутри университета обсудить необходимые модификации / надстройки над существующим функционалом, их последующее внедрение
5. Выпустить платформу для участников учебного процесса МГТУ
6. Расширять платформу с помощью краудсорсинга (портирование приложения на Android, создание чат-ботов в социальных сетях и мессенджерах, внедрение новой, требуемой функциональности)

Тем самым, я считаю, что философия продукта состоит в том, что вокруг ежедневно используемых и обязательно полезных функций, представленных в техническом и эстетическом совершенстве, выстроить среду краудсорсинга для расширения (фактического создания) специализированных учебных сервисов.

3 Модель / Конструкторская часть

3.1 Применяемые тех решения

Для проекта была выбрана классическая двухуровневая системная архитектура, с выделенными фронтэндом и бэкэндом (см. Рисунок 1).

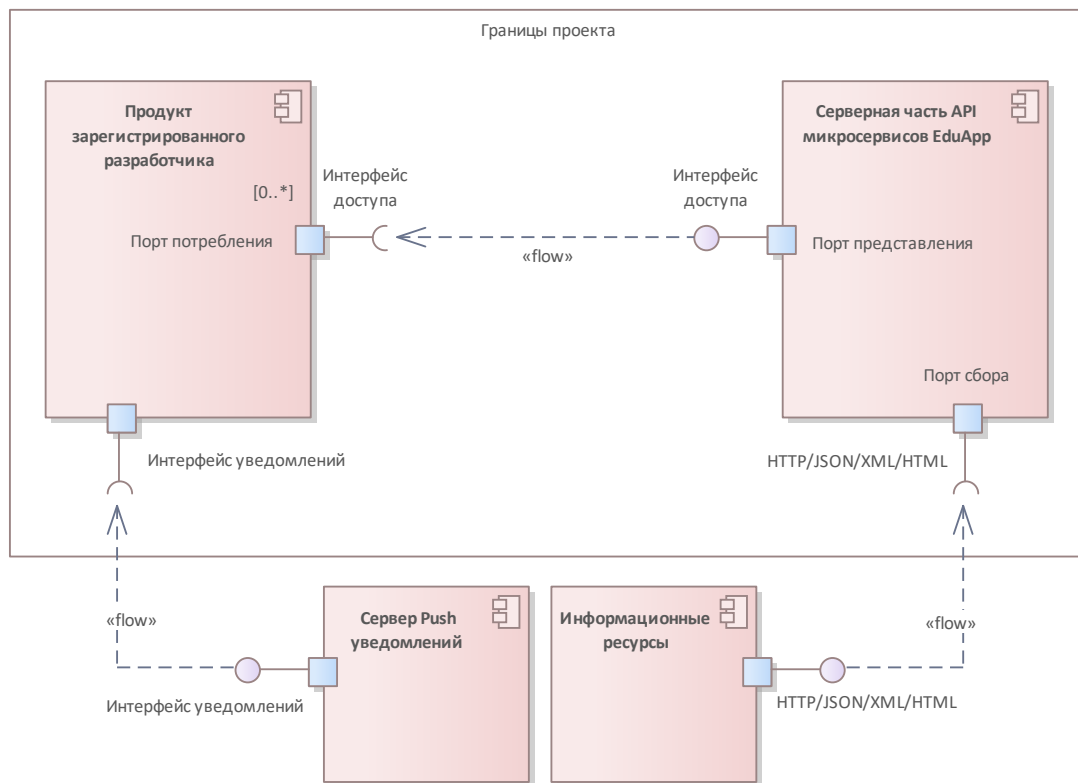


Рисунок 1. Системная архитектура «Мой МГТУ»¹

Данная архитектура рекомендуется для мобильных приложений, работающих с данными, где в качестве фронтэнда выступает мобильное приложение «Мой МГТУ» непосредственно, а бэкэнд выполняет функции хранилища данных и исполняет скрипты автоматизации.

В качестве хранилища данных была выбрана SQL СУБД PostgreSQL (возможно использование любой SQL ориентированной СУБД).

Метрика «Мой МГТУ»	Мобильное приложение (фронтэнд)	Микросервисное API (бэкэнд)
Язык программирования	Swift 5	Python 3.6
Среда разработки	XCode	PyCharm CE
СУБД	iOS UserDefaults	SQL (PostgreSQL)
Доступ к данным	JSON-клиент	JSON сервер по HTTPS
Защита от атак	Криптографическая, с аутентификацией пользователя по паролю и уникальному ключу. Обеспечивается аутентичность данных при помощи криптографической подписи	
Операционная система	iOS 11.0+	Linux\Windows
Занимаемый размер	7,6 МБ - исполнимый код 0,4 МБ - хранимые данные	Зависит от размера учебного заведения
Поддержка «темной темы»	Да, на выбор пользователя	-

Язык программирования Swift, для построения иерархии отношений между внешним видом, данными и бизнес-логикой, использует реализацию паттерна

¹ Используется среда моделирования на языке UML2 Enterprise Architect

«Модель-вид-контроллер», где «модель» выполняет функции работы с данными, «вид» – отвечает за отображение, а вся бизнес-логика сосредоточена в связующем их элементе - «контроллер». Причем, достаточно наследовать от UINavigationController и переопределить или добавить требуемые свойства.

Для мобильного приложения «Мой МГТУ» экранные формы-наследники UINavigationController были созданы с помощью Storyboard - встроенного инструмента среды разработки XCode. Данная функция помогает визуализировать связи контроллеров и внутренние графические элементы между собой (см. Рисунок 2). Далее, техническое и функциональное наполнение каждой экранной формы было описано программно, на языке программирования Swift.

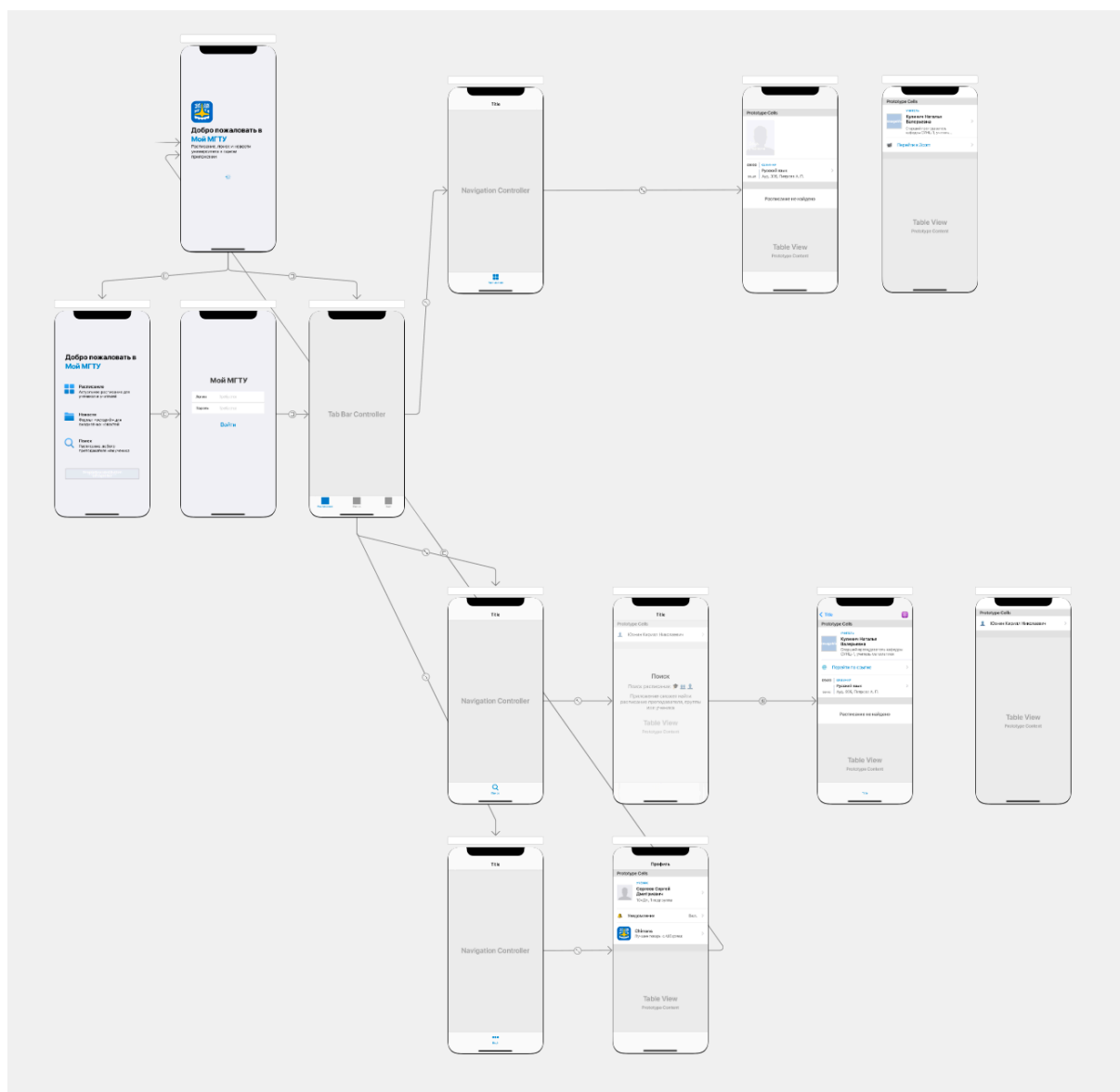


Рисунок 2. Модель экранных форм в виде Storyboard

Ввиду отсутствия готовой библиотеки показа (просмотра) Stories¹ в узнаваемом формате, т. е. который используется известными социальными сетями, было

¹ Характерная функциональная черта Instagram, WhatsApp и тд, позволяющая публиковать короткие видео или фото с теми событиями, которые происходят прямо сейчас. Дополнительно, можно указать ссылку, которая откроется при «свайпе» вверх

принято решение создать данную библиотеку самостоятельно. Данной библиотеке было дано название Warbler.

3.2 Формирование базы данных

В связи с тем, что во время разработки платформы, у меня отсутствовал доступ к существующим информационным ресурсам «Электронного университета», локальная задача формирования базы данных была наиболее острой.

Я принял решение о создании макетной базы данных, основанной на имеющейся информации об учениках, учителях и расписании в лицее 1580, с последующим переименованием существующих подгрупп в аналогичные по профилю в МГТУ.

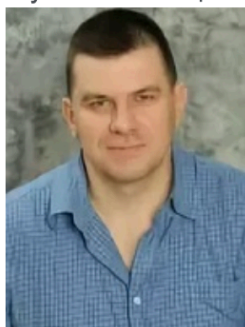
3.2.1 Получение списка учителей

Для получения списка учителей, должностей и групп с официального сайта школы, предоставленного департаментом образования, был разработан скрипт на языке программирования Python 3.6.

▼ Методическое объединение учителей общественных наук



Иванова Елена Ивановна
Председатель методического объединения, учитель истории



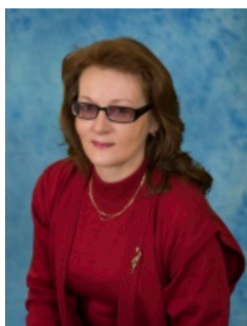
Богатин Андрей Аркадьевич
Учитель истории



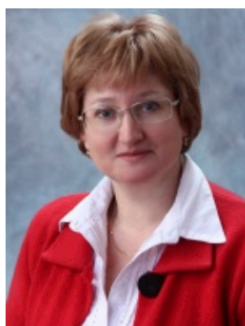
Карпец Федор Владимирович
Учитель истории



Ларина Марина Валентиновна
Учитель истории



Мухина Марина Георгиевна
Учитель обществознания



Порфирьева Ирина Анатольевна
Учитель истории



Преображенский Сергей Антонович
Учитель истории



Старостина Аннета Леонидовна
Учитель истории

Рисунок 3. Информация о педагогическом составе

Проектирование подпрограммы началось с анализа источника информации. Все школы города Москвы имеют сайт с одинаковой структурой и доменным расположением списка учителей, руководства и иных педагогических работников.

Данный список уже разделён на группы, для каждой из которых можно получить список учителей и их параметры: группа, ФИО, должность, ссылка на изображение, ссылка на страницу на официальном сайте, строковый латинский идентификатор.

Руководство. Педагогические и иные работники



Директор:
Граськин Сергей Сергеевич
Телефон: +7 (495) 316-50-22
Дополнительный телефон: +7 (905) 520-35-02
Адрес электронной почты: GraskinSS@edu.mos.ru

[➤ Развернуть все](#)

- Руководство
- Кафедра "Основы математики и информатики" (СУНЦ-1)
- Кафедра "Основы физики" (СУНЦ-2)
- Методическое объединение учителей начальной школы
- Методическое объединение учителей общественных наук
- Методическое объединение учителей русского языка и литературы
- Методическое объединение учителей естественных наук
- Методическое объединение учителей иностранных языков
- Методическое объединение учителей физической культуры и ОБЖ
- Иные педагогические работники
- Иные работники (1-й корпус)
- Иные работники (2-й корпус)
- Иные работники (3-й корпус)

Рисунок 4. Информация об административном составе

Для разработки скрипта была использована библиотека для web-crawl – BeautifulSoup.

Результатом работы подпрограммы является JSON файл, содержащий информацию об учителях, приведённую для работы с типами, описанными выше.

3.2.2 Получение списка учеников

Для получения списка учеников, подгрупп и содержащих их классов был разработан скрипт на языке программирования Python 3.6.

Проектирование скрипта началось с анализа источников информации. Так как списки учеников недоступны с сайта образовательного учреждения, были использованы альтернативные источники. Полученная информация в формате .pdf содержала: списки классов, разделение на подгруппы с указанием профиля каждой подгруппы, ФИО учеников, ФИО классного руководителя.

Для разработки подпрограммы была использована библиотека для перевода PDF в строку – PDFMiner.

Результатом работы подпрограммы является JSON файл, содержащий информацию о классах, подгруппах и учениках, приведённую для работы с типами, описанными выше.

3.2.3 Получение расписания

Для получения расписания был разработан скрипт на языке программирования Python 3.6.

Проектирование подпрограммы началось с анализа источника информации. Все школы города Москвы имеют сайт с одинаковой структурой и доменным расположением расписания уроков.

[Главная](#) / Расписание уроков.

Расписание уроков.

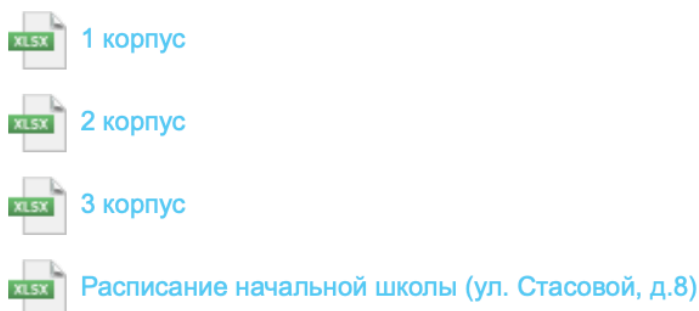


Рисунок 5. Информация о расписании

Дни	Уроки	10д1		10д2		10е1		10е2
Понедельник	1			Физика Шавгуддзе И.В.	311	Физика Калашникова А.А.	312	Английский язык Соловьева Д.А.
	2	Математика А Кулинич Н.В.	106	Физика Шавгуддзе И.В.	311	Физика Калашникова А.А.	312	Английский язык Соловьева Д.А.
	3	Русский язык Петрова А.П.	305	Физика Шавгуддзе И.В.	311	История Иванова Е.И.	304	История Иванова Е.И.
	4	История Иванова Е.И.	304	История Иванова Е.И.	304	Литература Якшина Т.С.	305	Литература Якшина Т.С.
	5	Английский язык Батурина М.И.	316	Английский язык Моцелкова И.А.	313	Теоретические основы физики Калашникова А.А.	317	Теоретические основы Калашникова А.А.
	6	Литература Петрова А.П.	305	Литература Петрова А.П.	305	Физическая культура физ-ра 4	сп.зал	Физическая культура физ-ра 4
	7	Литература Петрова А.П.	305	Литература Петрова А.П.	305	Физика Калашникова А.А.	209	Русский язык Якшина Т.С.
	8							
Дни	Уроки	10д1		10д2		10е1		10е2
Июль	1	Физика Шевцова Г.Ю.	309	Информатика и ИКТ Митрофанов М.С.	инф.	Математика Г Миканева О.Г.	210	
	2	Физика Шевцова Г.Ю.	310	Информатика и ИКТ Митрофанов М.С.	инф.	Математика Г Миканева О.Г.	210	ООП Партанский М.С.
	3	Информатика и ИКТ Митрофанов М.С.	инф.	Математика Г Миканева О.Г.	210	Основы схемотехники Соколовский К.А.	2к.210	Информатика и ИКТ Партанский М.С.
	4	Информатика и ИКТ Митрофанов М.С.	инф.	Математика Г Миканева О.Г.	210	Основы схемотехники Соколовский К.А.	2к.210	Информатика и ИКТ Партанский М.С.

Рисунок 6. Внутренняя структура расписания

Каждый файл имеет характерный шаблон оформления, откуда можно получить расписание для каждой подгруппы, включая информацию о преподавателе и месте проведения занятия.

Для разработки подпрограммы была использована библиотека для работы с Excel файлами – OpenPyxl.

Результатом работы подпрограммы является JSON файл, содержащий информацию о расписании подгрупп, приведённую для работы с типами, описанными выше.

3.2.4 Преобразование собранных данных

В лицее классы имеют разделение по профилям: физико-математический, физико-химический, химико-биологический, социально-экономический, информационно-технологический. В связи с данной классификацией было проведено преобразование, описанное в таблице ниже.

ID профиля	Полное наименование	Кафедры
it	информационно-технологический	ИУ7, ИУ3, ИУ8

ID профиля	Полное наименование	Кафедры
se	социально-экономический	ЮР1
fm	физико-математический	Э1, Э3, РЛ1, МТ8, ФН1, СМ3, СМ4, СМ6
fh	физико-химический	ФН5
hb	химико-биологический	ФН5

Лицейским подгруппам были выданы случайные университетские группы по данным кафедрам (курс и группа).

3.3 Микросервисная архитектура

При создании технического задания я осознавал, что реальные нужды образовательной организации могут отличаться от потенциально реализованного функционала и требуют тщательной проработки и анализа. Для того, чтобы оставить возможность быстрой адаптации и расширения, было принято решение реализовать Web-API на базе микросервисной архитектуры.

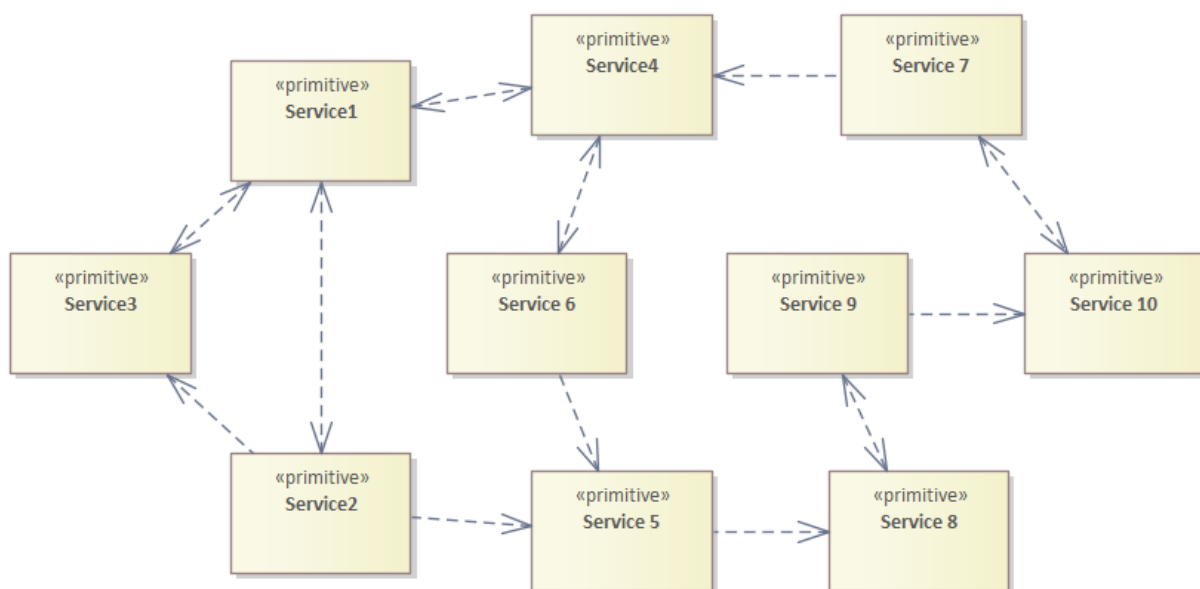


Рисунок 7. Коммуникации в распределенной микросервисной архитектуре

Микросервисная архитектура — принципиальная организация распределенной системы на основе микросервисов и их взаимодействия друг с другом и со средой по сети, а также принципов, направляющих проектирование архитектуры, её создание и эволюцию. Где микросервис определяется как сущность, обладающая функциональностью с комбинацией следующих свойств:

1. Он небольшой
2. Он независимый
3. Он строится вокруг бизнес-потребности и использует ограниченный контекст

4. Он взаимодействует с другими микросервисами по сети, где они обычно распределённые
5. Его распределённая суть обязывает использовать подход, учитывающий отказы в распределённой системе
6. Централизация ограничена сверху на минимуме
7. Процессы его разработки и поддержки требуют автоматизации
8. Его развитие итерационное

3.3.1 Возможные ошибки

HTTP Status Code

Status Code	Значение	Описание
200	OK	Всё сработало, как ожидалось.
400	Bad Request	Запрос был неправильный, зачастую ввиду отсутствия обязательного параметра.
401	Unauthorized	Вход не был осуществлён.
402	Request Failed	Параметры были верны, но запрос провалился.
403	Forbidden	Недостаточно прав для осуществления запроса.
404	Not Found	Запрашиваемый ресурс не был найден.
422	Unprocessable Entity	Неверный ввод.
429	Too Many Requests	Очень много поступающих API запросов. Постарайтесь остановить все действующие.
500, 502, 503, 504	Server Errors	Что-то случилось с серверами EduApp.

Виды ошибок

В случае, если поступаемый от сервера заголовок *Content-Type* равен *application/json*, а HTTP Status Code отличен от 200, можно получить вид ошибки по ключу *error*.

```
{
  "error": "missing_request_id"
}
```

Error Type	HTTP Status Code	Description
missing_token	401	В заголовке отсутствует X-ACCESS-TOKEN.
invalid_token	401	X-ACCESS-TOKEN неверен или его срок действия истёк.
missing_request_id	403	В заголовке отсутствует X-REQUEST-ID.
invalid_request_id	403	X-ACCESS-TOKEN недействителен.

Error Type	HTTP Status Code	Description
missing_auth_credentials	401	Недостаточно аргументов для Basic Auth.
invalid_auth_credentials	401	Нельзя осуществить авторизацию по предоставленным Basic Auth аргументам.
invalid_auth	401	Ошибка авторизации.
invalid_filter	422	Неверный фильтр.
invalid_id	404	Неверный id для поиска.
invalid_filter	422	Неверный фильтр.
missing_permissions	403	Недостаточно прав для осуществления запроса.

3.3.2 Pagination

В связи с возможным большим объемом информации и потребностью в её отображении на мобильных устройствах, необходимо реализовать Pagination, алгоритм, под которым понимают постраничный вывод информации с возможностью перехода на другие страницы.

Параметры пагинатора доступны в ответе:

```
{
  "meta": {
    "count": 20,
    "page": 1,
    "per_page": 20,
    "total_found": 214
  },
  ...
}
```

Описание

Variable	Type	Description
count	Int	Количество элементов на текущей странице
page	Int	Номер выбранной страницы (default: 1)
per_page	Int	Количество элементов на страницу (default: 20)
total_found	Int	Всего элементов найдено (сумма count всех страниц)

Управление

URL параметры `page` и `per_page` могут быть установлены вручную.

3.3.3 Фильтры

Некоторым микросервисам требуется алгоритм фильтров для продвинутого поиска. Для этого необходимо добавить в Body HTTP запроса массив из фильтров следующего вида:

```
[
  {"field": "name", "op": "like", "value": "%Михаил%"},
  ...
]
```

```
[{"field": "status", "op": "==", "value": "2"}]
```

Подобный запрос для списка пользователей вернёт всех учителей (status = 2) с именем Михаил («Михаил» содержится в поле name).

Описание

Variable	Type	Description
field	String	Название поля (для разных запросов названия полей уникальны, учитывайте индивидуально)
op	String	Оператор. Возможные варианты: == (равно), != (не равно), >= (больше либо равно), > (больше), <= (меньше либо равно), < (меньше), like (содержит схему)
value	String	Значение для фильтра

Like-схемы

- %мих% - содержит "мих" в поле
- Ко% - содержит "Ко" в поле, при этом "Ко" являются первыми символами
- %ич - содержит "ич" в поле, при этом "ич" являются последними символами

3.3.4 Создание ID пользователя

ID пользователя – транслитерация ФИО пользователя. Для создания ID была использована библиотека «transliterate» на языке программирования Python.

Пример использования: ivanov-ivan-ivanovich

3.3.5 Статусы пользователей

Status, Int	Description
0	Пользователь заблокирован
1	Ученик
2	Учитель

3.3.6 Аутентификация

Для каждого запроса к API необходимо иметь в заголовке Request ID (X-REQUEST-ID), который выдаётся индивидуально каждому разработчику и является уникальным. Для запросов, содержащих конфиденциальную информацию

(пользователи, истории и тд), необходимо иметь в заголовке Access token (X-ACCESS-TOKEN). Access Token представляет собой JWT (JSON Web Token).

Все запросы осуществляются по HTTPS. Остальные запросы будут проигнорированы.

HTTP Method: GET

Path: /login

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{
  "token": "<X-ACCESS-TOKEN>"
}
```

3.3.7 Версии

SQL-схема микросервиса Versions

Столбец	Тип	Описание
type	String	Идентификатор сервиса
version	String	Номер версии
build	String	Номер сборки

3.3.7.1 Получить список актуальных версий

HTTP Method: GET

Path: /versions

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{
  "versions": {
    "<type>": {
      "build": ...,
      "version": ...
    },
    ...
  }
}
```

```
}  
}
```

3.3.7.2 Получить актуальную версию для type

HTTP Method: GET

Path: /versions/<type>

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{  
  "<type>": {  
    "build": ...,  
    "version": ...  
  }  
}
```

3.3.8 Ссылки

SQL-схема микросервиса Links

Столбец	Тип	Описание
type	String	Идентификатор ссылки
value	String	Ссылка

3.3.8.1 Получить список ссылок

HTTP Method: GET

Path: /links

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{  
  "links": {  
    "<type>": {  
      "value": ...  
    },  
    ...  
  }  
}
```

3.3.8.2 Получить ссылку по type

HTTP Method: GET

Path: /links/<type>

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{
  "<type>": {
    "value": ...
  }
}
```

3.3.9 Другие проекты

Во время разработки мобильного приложения я понял, что для распространения новых продуктов на основе API будет полезен отдельный блок для их продвижения. Зарегистрированные разработчики имеют право добавить свои проекты.

SQL-схема микросервиса Other_Projects

Столбец	Тип	Описание
id	String	Идентификатор проекта
name	String	Название проекта
photo	String	Ссылка на изображение-иконку
link	String	Ссылка на сторонний проект
description	String	Краткое рекламное описание проекта

3.3.9.1 Получить список партнёрских проектов

HTTP Method: GET

Path: /other_projects

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```
{
  "other_projects": {
    "<id>": {
      "description": ...,
      "link": ...,
    }
  }
}
```

```

    "name": ...,
    "photo": ...
  },
  ...
}
}

```

3.3.9.2 Получить партнёрский проект по id

HTTP Method: GET

Path: /other_projects/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID

Пример ответа:

```

{
  "<id>": {
    "description": ...,
    "link": ...,
    "name": ...,
    "photo": ...
  }
}

```

3.3.10 Пользователи

SQL-схема микросервиса Users

Столбец	Тип	Описание
id	String	Идентификатор пользователя
study_group	String	Для студентов – кафедра (ИУ7), для преподавателей – направление/название кафедры (Кафедра "Основы математики и информатики" (СУНЦ-1))
link	String?	Ссылка на внешний источник (для преподавателей изначальное значение – профиль на сайте образовательного учреждения)
name	String	ФИО пользователя
photo	String?	Ссылка на фотографию
status	Int	Статус пользователя
subject	String	Для студентов – группа (41Б), для преподавателей – должность (Ассистент кафедры СУНЦ-1, учитель информатики)
zoom	String?	Идентификатор комнаты в дистанционной среде обучения
request_id	String?	Идентификатор разработчика

3.3.10.1 Получить всех пользователей

Присутствует Pagination

HTTP Method: GET

Path: /users

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "meta": {
    "count": ...,
    "page": ...,
    "per_page": ...,
    "totalFound": ...
  },
  "users": {
    "<id>": {
      "link": ...,
      "name": ...,
      "photo": ...,
      "status": ...,
      "study_group": ...,
      "subject": ...,
      "zoom": ...
    },
    ...
  }
}
```

3.3.10.2 Получить всех пользователей, с фильтрами

Присутствует Pagination

HTTP Method: POST

Path: /users

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Тело запроса содержит массив необходимых фильтров. Пример:

```
[
  {"field": "name", "op": "like", "value": "%Михаил%"},
  {"field": "status", "op": "==", "value": "2"}
]
```

Пример ответа:

```
{
  "meta": {
    "count": ...,
    "page": ...,
    "per_page": ...,
    "totalFound": ...
  },
  "users": {
    "<id>": {
      "link": ...,
      "name": ...,
      "photo": ...,
      "status": ...,
      "study_group": ...,
      "subject": ...,
      "zoom": ...
    },
    ...
  }
}
```

3.3.10.3 Получить пользователя по id

HTTP Method: GET

Path: /users/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "<id>": {
    "link": ...,
    "name": ...,
    "photo": ...,
    "status": ...,
    "study_group": ...,
    "subject": ...,
    "zoom": ...
  }
}
```

3.3.10.4 Изменить пользователя по id

HTTP Method: PUT

Path: /users/<id>

Query Parameters

Необходимые для изменения поля передаются в качестве параметров. Для изменения доступны только необязательные поля. Для обнуления поля необходимо передать пустое значение для параметра.

Пример Path для изменения zoom и photo:

`/users/<id>?zoom=<zoom_id>&photo=<photo_url>`

Пример Path для обнуления photo:

`/users/<id>?photo=`

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа включает данные изменённого пользователя:

```
{
  "<id>": {
    "link": ...,
    "name": ...,
    "photo": ...,
    "status": ...,
    "study_group": ...,
    "subject": ...,
    "zoom": ...
  }
}
```

3.3.11 Курсы

Микросервис предоставляет доступ к учебным курсам и их деталям.

SQL-схема микросервиса Courses

Столбец	Тип	Описание
id	String	Идентификатор учебного курса
name	String	Название учебного курса (предмета)
teacher_id	String	Идентификатор преподавателя (id пользователя)

3.3.11.1 Получить все курсы

Присутствует Pagination

HTTP Method: GET

Path: /courses

Headers

Key	Value
X-REQUEST-ID	Request ID

Key	Value
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "meta": {
    "count": ...,
    "page": ...,
    "per_page": ...,
    "totalFound": ...
  },
  "courses": {
    "<course_id>": {
      "name": ...,
      "teacher_id": ...
    },
    ...
  }
}
```

3.3.11.2 Получить все курсы, с фильтрами

Присутствует Pagination

HTTP Method: POST

Path: /courses

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Тело запроса содержит массив необходимых фильтров. Пример:

```
[
  {"field": "teacher_id", "op": "==", "value": "lesnichenko_valentina_aleksandrovna"}
]
```

Пример ответа:

```
{
  "meta": {
    "count": ...,
    "page": ...,
    "per_page": ...,
    "totalFound": ...
  },
  "courses": {
    "<course_id>": {
      "name": ...,
      "teacher_id": ...
    },
    ...
  }
}
```

```
}
```

3.3.11.3 Получить курс по id

HTTP Method: GET

Path: /courses/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "<course_id>": {
    "name": ...,
    "teacher_id": ...
  }
}
```

3.3.12 Stories

SQL-схема микросервиса Stories

Столбец	Тип	Описание
Stories		
id	String	Идентификатор истории
info_id	String	Идентификатор информации об истории
Stories_Infos		
id	String	Идентификатор информации об истории
hex	String	Средний цвет photo в формате HEX
photo	String	Ссылка на изображение
title	String	Короткий заголовок
Stories_Snaps		
id	String	Идентификатор снимка
story_id	String	Идентификатор истории-родителя
mime_type	String	Mime Type для photo
redirect_url	String?	Ссылка для переадресации
text	String	Текст над снимком
photo	String	Ссылка на медиаконтент

3.3.12.1 Получить все истории

HTTP Method: GET

Path: /stories

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "story_cards": {
    "count": ...,
    "stories": [
      {
        "id": ...,
        "info": {
          "hex": ...,
          "id": ...,
          "photo": ...,
          "title": ...
        },
        "snaps": [
          {
            "id": ...,
            "mime_type": ...,
            "redirect_url": ...,
            "text": ...,
            "url": ...
          },
          ...
        ],
        "snaps_count": ...
      },
      ...
    ]
  }
}
```

3.3.13 Группы

SQL-схема микросервиса Subgroups

Столбец	Тип	Описание
id	String	Идентификатор учебного курса
subject	String	Идентификатор профиля
teacher_id	String	Идентификатор руководителя / куратора (id пользователя)
head_id	String	Идентификатор старосты (id пользователя)
parent_id	String	Идентификатор кафедры

3.3.14 Расписание

SQL-схема микросервиса Lessons

Столбец	Тип	Описание
id	String	Идентификатор учебного курса
subgroup	String	Идентификатор подгруппы
room	String	Место проведения занятия
number	Int	Номер занятия
day	Int	Номер дня занятия

3.3.14.1 Получить расписание по id курса

HTTP Method: GET

Path: /timetable/course/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "timetable": {
    "<day>": {
      "<lesson>": {
        "id": ...,
        "name": ...,
        "room": ...,
        "subgroup": [
          ...
        ],
        "teacher_id": ...
      },
      ...
    },
    ...
  }
}
```

3.3.14.2 Получить расписание по id подгруппы

HTTP Method: GET

Path: /timetable/subgroup/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "timetable": {
    "<day>": {
      "<lesson>": {
        "id": ...,
        "name": ...,
        "room": ...,
        "subgroup": [
          ...
        ],
        "teacher_id": ...
      },
      ...
    },
    ...
  }
}
```

3.3.14.3 Получить расписание по id пользователя

HTTP Method: GET

Path: /timetable/user/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "timetable": {
    "<day>": {
      "<lesson>": {
        "id": ...,
        "name": ...,
        "room": ...,
        "subgroup": [
          ...
        ],
        "teacher_id": ...
      },
      ...
    },
    ...
  }
}
```

3.3.14.4 Получить расписание по id комнаты

HTTP Method: GET

Path: /timetable/room/<id>

Headers

Key	Value
X-REQUEST-ID	Request ID
X-ACCESS-TOKEN	Access token

Пример ответа:

```
{
  "timetable": {
    "<day>": {
      "<lesson>": {
        "id": ...,
        "name": ...,
        "room": ...,
        "subgroup": [
          ...
        ],
        "teacher_id": ...
      },
      ...
    },
    ...
  }
}
```

3.4 Автоматизация исполнения скриптов

Чтобы избежать неполадок в связи с обновившейся информацией и последующей неактуальностью базы данных, было принято решение назначить регулярное исполнение crawl-скриптов, описанных выше.

На бэкенд-сервере запущен CRON – программа, выполняющая crawl-скрипты по расписанию, ежедневно в 0:00.

3.5 Библиотека EduAppCore

EduAppCore – библиотека на языке программирования Swift для связи с API, специально разработанная для целей проекта и распространяющаяся в открытом доступе для участников учебного процесса.

3.6 Библиотека pyEduApp

pyEduApp – библиотека на языке программирования Python для связи с API, специально разработанная для целей проекта и распространяющаяся в открытом доступе для участников учебного процесса.

Ниже представлен пример использования для получения расписания.

```
import pyEduApp as edu
edu.request_id = "<REQUEST_ID>"
user = edu.User(username="<username>", password="<password>")
print(user.get_timetable_user("<user_id>"))
```

3.7 Библиотека Warbler

Warbler - библиотека, разработанная для воссоздания функционала Stories который присутствует во всех приложениях-социальных сетях.

Было проведено сравнение функционала Warbler и аналогичных модулей из социальных сетей. Результаты сравнения сведены в таблицу:

Параметр сравнения	Warbler	Instagram
Язык программирования	Swift	Objective-C
Управление жестами	Да	Да
Формат контента	Видео+Фото	Видео+Фото
Поддержка планшетных устройств	Да	Нет
Версия программного обеспечения	iOS 11.0+	iOS 11.0+
Адаптирован для новостей и объявлений	Да	Нет
Адаптирован для Notch-устройств	Да	Да

Warbler – первая библиотека с подобным функционалом на языке программирования Swift, специально разработанная для целей проекта и распространяющаяся в открытом доступе с помощью GitHub и CocoaPods для установки в среду разработки.

3.7.1 Описание реализации

Для инициализации и работы библиотеки необходимо ознакомиться со следующими структурами:

- **WarblerStoriesLoader** - служит для загрузки данных о Stories из любого (расширяемого в дальнейшем) источника. Основан на паттерне «Стратегия», поэтому наследование от данного протокола даст возможность создать свой загрузчик и использовать его в отличающихся проектах.

Предлагаемые загрузчики «из коробки»:

1. **WarblerJSONURLLoader** - загрузка JSON данных по указанному URL и приведение ко внутреннему типу.
2. **WarblerJSONFileLoader** - загрузка JSON данных из файла по указанному пути и приведение ко внутреннему типу.
3. **WarblerAPIResponseLoader** - приведение ко внутреннему типу данного на вход словаря, со строковым типом ключа и любым значением по ключу.

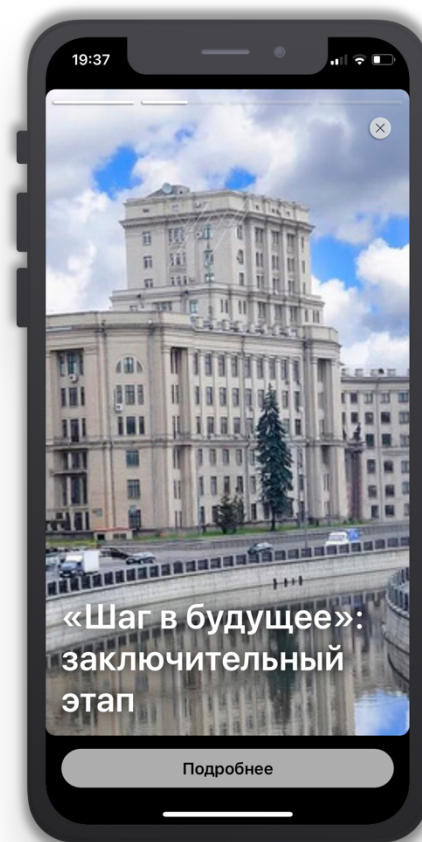


Рисунок 8. Внешний вид контроллера Warbler

- **WarblerDataModel** - служит для работы с полученными загрузчиком данными и содержит привычные для разработчиков по встроенным UITableView и UICollectionView методы: количество элементов в секции (numberOfItems(in section:Int) -> Int), получить элемент для ячейки (itemForCell(at indexPath: IndexPath) -> WRStory?), получить Stories (getStories(forceUpdate : Bool = false, completionHandler: (() -> ())? = nil) -> WRStories?).

Для инициализации требуется загрузчик — наследник типа WarblerStoriesLoader.

- **WarblerManager** - служит для управлением экранами Stories и имеет два основных метода: показать (showStories(above presentingVC: UIViewController, at pickedStoryIndex : Int = 0)) и скрыть (dismissStories()).

Для инициализации требуется загрузчик — наследник типа WarblerStoriesLoader или WarblerDataModel.

- **WRInfo** - служит для передачи информации о ячейке. Параметры: id - строковый уникальный идентификатор, title - строковый заголовок ячейки, picture - строка, содержащая ссылку на картинку ячейки, hex - строковый код информации среднего цвета картинки ячейки.

Hex был выделен в отдельный параметр во время разработки мобильного приложения. Алгоритм расчёта среднего цвета на старых версиях iOS требует много ресурсов центрального процессора, и при частом скролле содержащей блок историй UITableView негативно влияет на потребление электроэнергии. Данная проблема была решена.

- **MimeType** - вспомогательный перечисляемый тип для обозначения типа контента. Инициализируется через значение строки. Принимает два значения: «video» и «image», в противном случае «unknown».
- **WRSnap** - является репрезентацией данных об одной истории ячейки (снимка) и имеет параметры: id - строковый уникальный идентификатор снимка, redirectUrl - строка-ссылка для переадресации, text - строка-описание в нижней части снимка (накладывается на снимок), url - строка-ссылка на файл снимка, mimeType - строка, определяющая тип данных mime по ссылке url (image, video). На этапе инициализации класса создается параметр kind типа MimeType.
- **WRStory** - является репрезентацией данных для одной ячейки и имеет несколько параметров. Среди них: id - строковый уникальный идентификатор ячейки, snaps и snapsCount - истории ячейки (типа WRSnap) и их количество, info - дополнительная информация о ячейке (типа WRInfo).

- **WRStories** - является репрезентацией данных для всех ячеек и имеет два параметра: count - переменная с количеством историй, служит для отображения макетных изображений в интерфейсе до полного скачивания информации; stories - массив из элементов типа WRStory.

3.7.2 Поддерживаемые жесты и управление

Поддерживается управление жестами. Согласно рекомендациям Apple Human Guidelines следует использовать знакомые для многих (80%) пользователей жесты. В связи с этим социальные сети имеют сходное управление жестами для показа историй.

1. «Свайп¹» вниз или крестик - закрытие экрана со Stories
2. «Свайп» влево/вправо - переключение на предыдущий/следующий экран с Story
3. «Тап²» по левому/правому краю - переключение историй в пределах одной ячейки
4. «Долгий тап³» - пауза

Работает с фото и видео материалами, при ошибках появляется кнопка для повторного запроса.

¹ Смахивающее движение рукой по экрану

² Прикосновение рукой (пальцем) к экрану

³ Прикосновение рукой (пальцем) к экрану и удерживать

3.8 Push-уведомления

В приложение внедрены оповещения о начале урока, основанные на загруженном при последнем посещении приложения расписании.

При первом открытии приложения, в наборе всплывающих сообщений-инструкций, пользователь может разрешить отправку приложением уведомлений или отложить принятие решения.

Пользователь всегда может включить/отключить уведомления или настроить их доставку в секции UITabBarController «Ещё».

Доставка уведомлений о начале урока будет выполняться за 5 минут (или другое указанное пользователем вручную число минут).

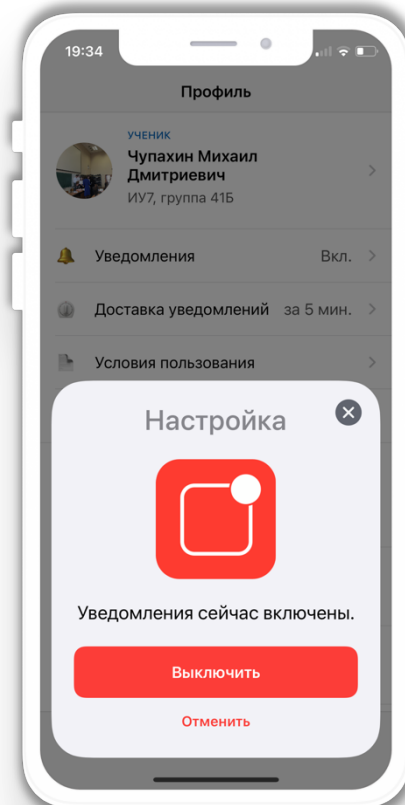


Рисунок 9. Управление уведомлениями

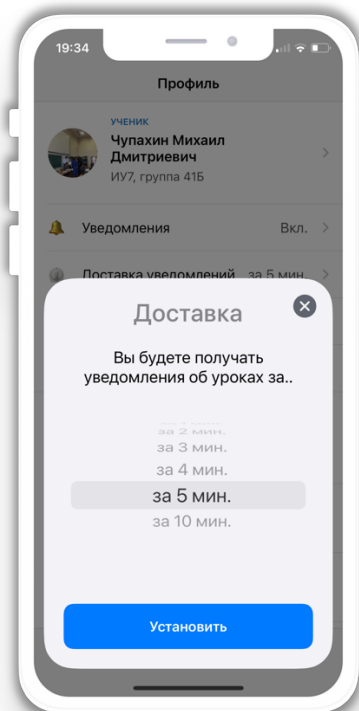


Рисунок 10. Задание времени напоминания

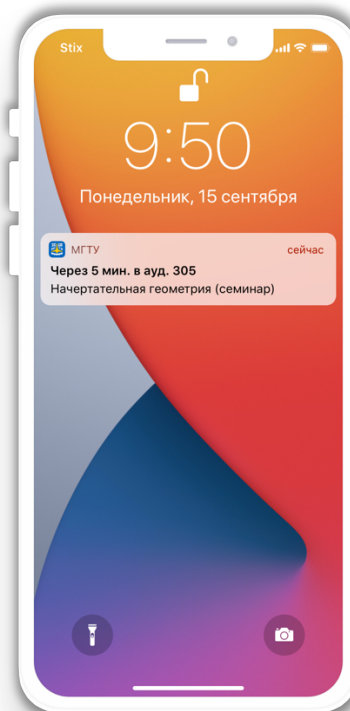


Рисунок 11. Внешний вид уведомления

3.9 Наследники «UIViewController»

Архитектура мобильного приложения «Мой Лицей» легко описывается иерархией его контроллеров. Каждый контроллер является наследником общего класса «UIViewController», наследует все функции родителя, переопределяет и добавляет дополнительные.

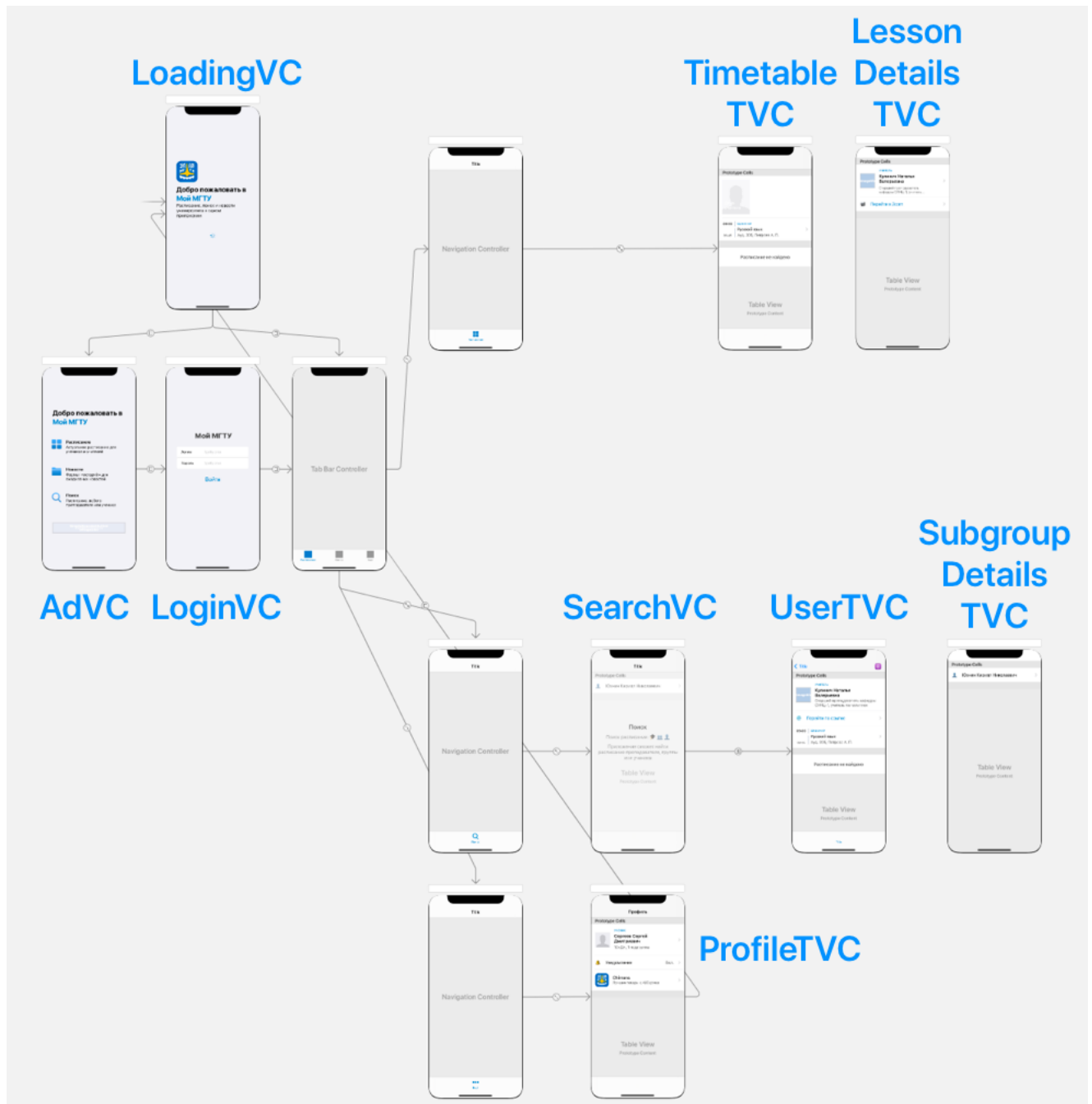


Рисунок 12. Наследники «UIViewController»

3.9.1 Контроллер «LoadingVC»

Описание: Цикл пользовательского интерфейса начинается с LoadingVC. Экран приветствует пользователя и вкратце описывает функционал приложения.

Функция: показывает анимированный UIActivityIndicatorView и загружает данные о пользователе и поддерживаемой версии клиента. Если версия запущенного приложения больше не поддерживается, то пользователь будет оповещён всплывающим сообщением. Если приложение было открыто впервые или пользователь не авторизован, то происходит анимированный переход к AdVC, иначе проверяется текущий статус пользователя – если аккаунт заблокирован, то пользователь будет оповещён всплывающим сообщением, иначе происходит анимированный переход к TimetableTVC.

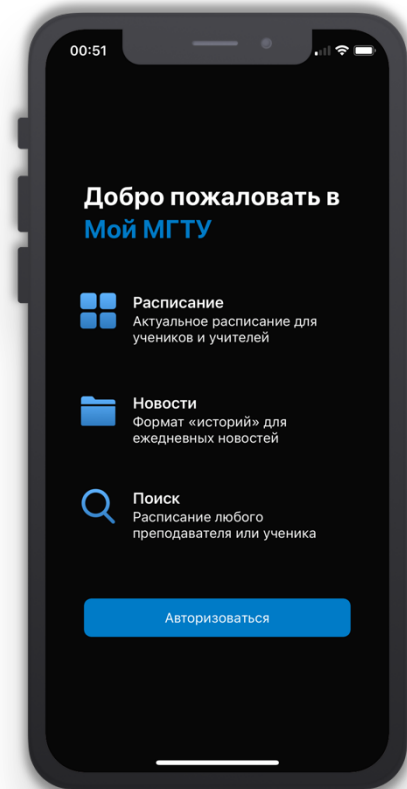


Рисунок 14. Вид AdVC (темная тема)

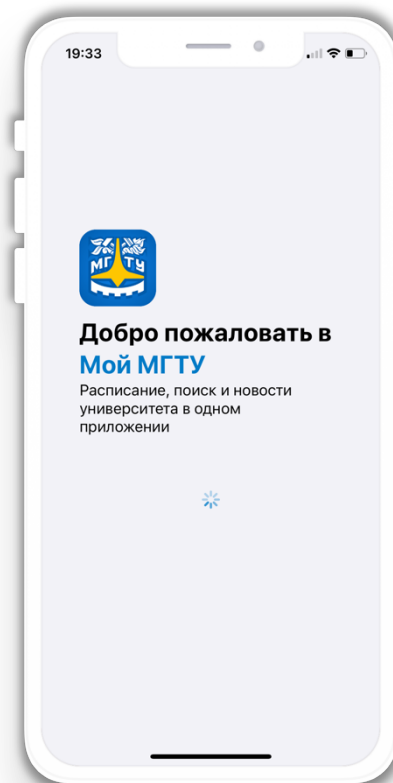


Рисунок 13. Вид LoadingVC

В случае возникновения любой ошибки пользователь будет оповещён всплывающим сообщением.

3.9.2 Контроллер «AdVC»

Описание: Декоративный экран, подробнее описывающий текущий функционал или нововведения.

Функция: Единственный интерактивный элемент – кнопка авторизации. При нажатии происходит анимированный переход к LoginVC.

3.9.3 Контроллер «LoginVC»

Описание: Экран является формой для авторизации.

Функция: Два поля типа UITextField служат для ввода логина и пароля. Как только данных, введенных в форму, будет достаточно для запроса авторизации, кнопка “Войти” типа UIButton будет доступна для взаимодействия с пользователем.

При нажатии на кнопку авторизации формируется запрос. При успешной авторизации мобильное приложение получает информацию о пользователе, проверяет его статус на блокировку – если аккаунт не заблокирован, то происходит анимированный переход к UITabBarController, являющемуся родительским для UINavigationController, основным вид-контроллером которого является TimetableTVC.

В случае возникновения любой ошибки пользователь будет оповещён всплывающим сообщением.

3.9.4 Контроллер «TimetableTVC»

Описание: Первый экран в UITabBarController доступный после авторизации. Содержит персональное расписание на неделю, новости и объявления в формате Stories.

Функция: Большую часть экрана занимает таблица типа UITableView. В данной таблице есть ячейки нескольких классов: StoriesTableViewCell (первая ячейка для Stories) и LessonTableViewCell (ячейка с уроком).

Перед появлением экрана, происходит инициализация – вызывается метод setup(), который получает информацию для показа в таблице и инициализирует Warbler для работы. Если пользователь использует приложение впервые, то открывается инструкция (набор всплывающих сообщений). После появления экрана происходит удаление полученных уведомлений приложения “Мой Лицей” из центра уведомлений устройства.

Для каждого учебного дня имеется заголовок с наименованием дня недели.

Если расписание отсутствует в базе данных, то пользователь увидит соответствующее сообщение в таблице.

При нажатии на ячейку типа LessonTableViewCell происходит анимированный переход к экрану LessonDetailsTableViewController.

Для ячейки типа StoriesTableViewCell таблица передаёт delegate опционального типа WarblerStoriesHandler. StoriesTableViewCell содержит в себе UICollectionView, содержащая ячейки типа StoriesCollectionViewCell, где обращается к WarblerManager и при нажатии на которую вызывается метод делегата didSelectStory(at row: Int).

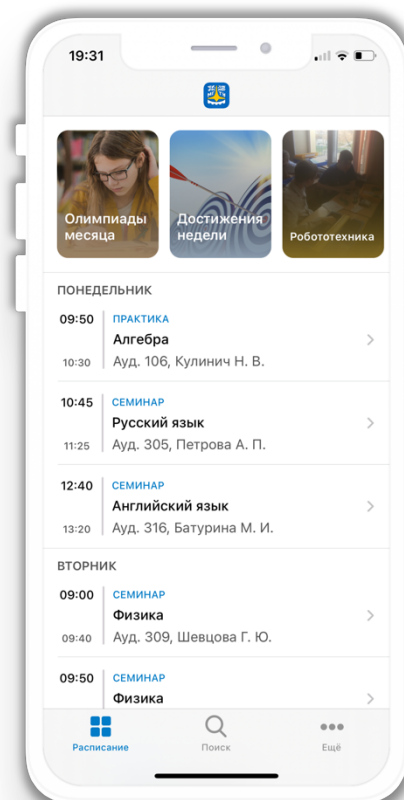


Рисунок 15. Вид TimetableTVC

3.9.5 Контроллер «LessonDetailsTVC»

Описание: Данный экран открывается с логикой pop-up и содержит подробную информацию об уроке: ссылка на учителя, номер урока по счёту, аудитория, вид занятия, время начала и завершения. В заголовке указано название урока.

Функция: принимает при переходе от предыдущего вид-контроллера информацию об уроке типа Lesson и delegate типа UserCellDelegate. Основную площадь экрана занимает таблица, первая ячейка типа TeacherTableViewCell содержит информацию об учителе, остальные ячейки типа UITableViewCell показывают справочную информацию по алгоритму словаря (ключ-значение).

Если при нажатии на ячейку типа TeacherTableViewController, указанный учитель существует в системе, происходит закрытие текущего pop-up, вызывается метод делегата didSelect(_ user: User), далее переход к UserTVC.

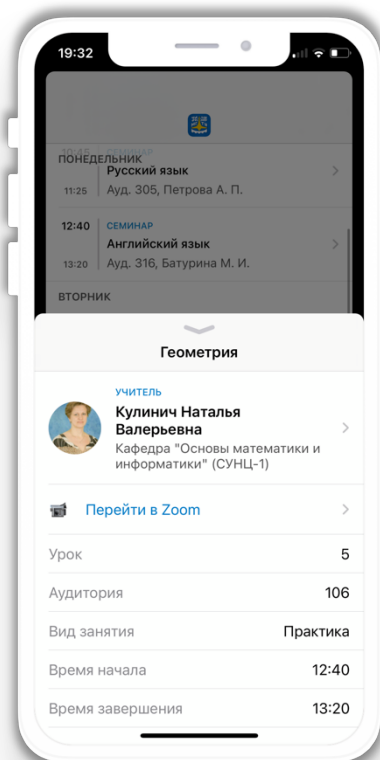


Рисунок 16. Вид LessonDetailsTVC

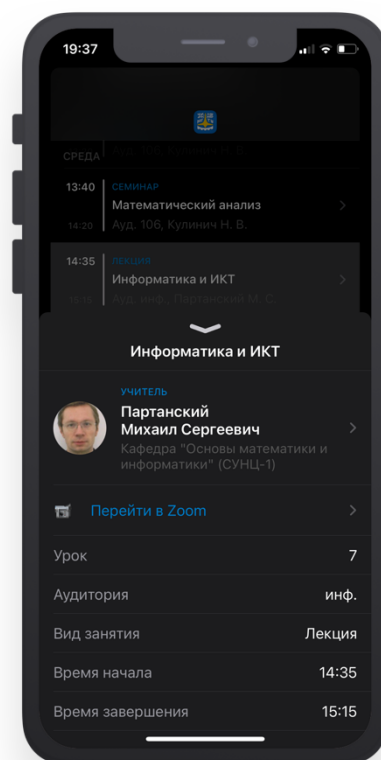


Рисунок 17. Вид LessonDetailsTVC
(темная тема)

3.9.6 Контроллер «ProfileTVC»

Описание: Последний экран в UITabBarController. Содержит информацию о личном аккаунте, условиях пользования, даёт возможность настройки профиля и уведомлений, обратную связь с разработчиком, предлагает другие приложения разработчика.

Функция: Данный экран наследуется от UITableViewController и содержит в себе ячейки трех различных классов: ProfileTableViewCell (ячейка профиля), SettingTableViewCell (ячейки настроек), OtherProjectTableViewCell (ячейки “других проектов”).

При нажатии на ячейку типа ProfileTableViewCell, вызывается всплывающее меню выбора дальнейшего действия типа UIAlertController. Пользователь может сменить фото (выбор из фотоплёнки), выйти из аккаунта, узнать версию используемого приложения и покинуть меню.

При нажатии на ячейку типа SettingTableViewCell возможно включить/выключить уведомления и настроить их отправку, получить и перейти по текущим ссылкам для условия пользования или обратной связи.

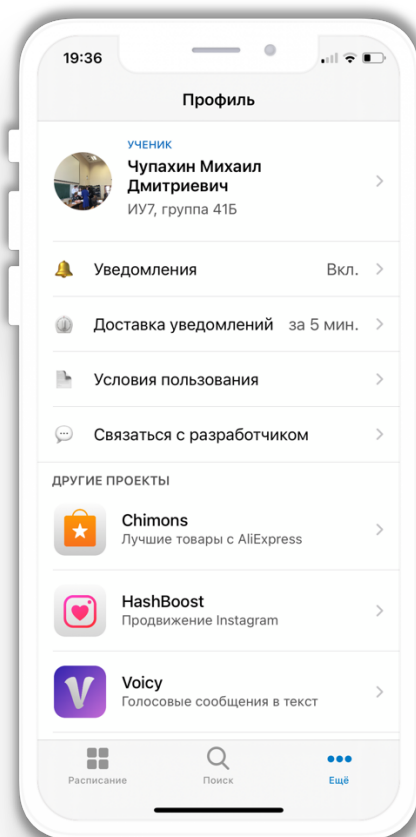


Рисунок 18. Вид ProfileTVC

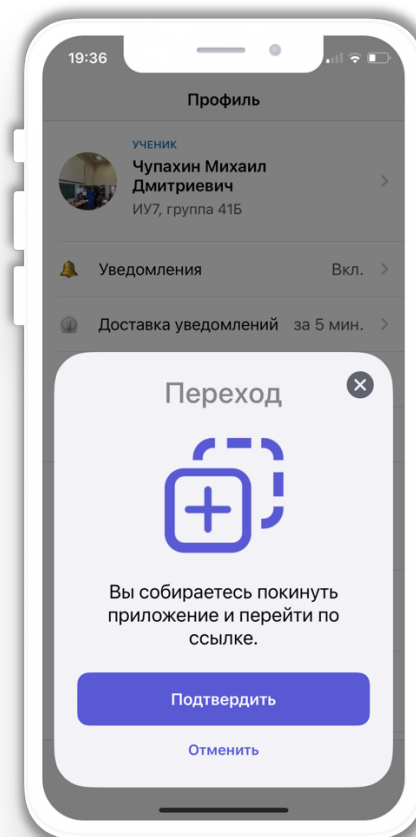


Рисунок 19. Вид ProfileTVC
(переход по ссылке)

3.9.7 Контроллер «SearchVC»

Описание: Второй экран в UITabBarController. Содержит поисковую строку и справку для поиска. Результат поискового запроса отображается в таблице.

Функция: После первичной инициализации поиска, поле поискового запроса становится доступным для пользовательского взаимодействия. При вводе символов в поисковую строку, происходит поисковый запрос, и появляется скрытая до этого таблица, в которой отображены результаты поиска – ученики, учителя, классы с подгруппами. Результаты поиска представлены ячейками типа `SearchResultTableViewCell`.

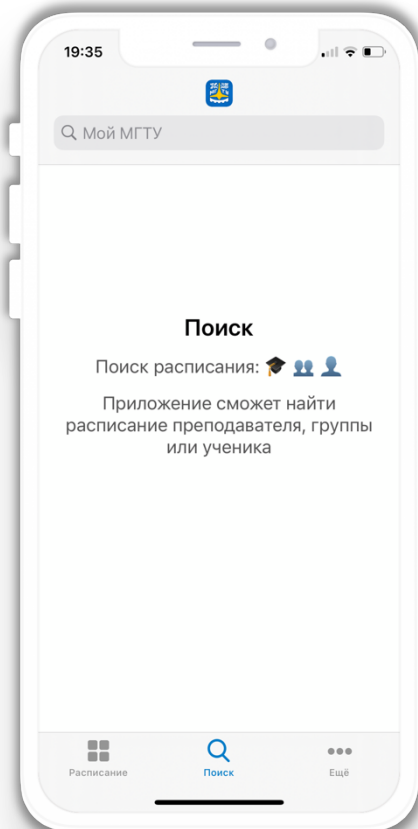


Рисунок 20. Вид SearchVC

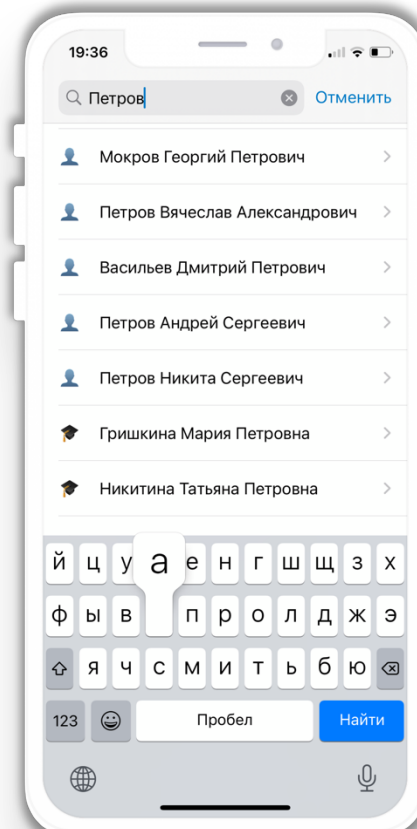


Рисунок 21. Вид SearchVC
(результаты)

При нажатии на ячейку поискового результата, происходит анимированный переход к UserTVC.

3.9.8 Контроллер «UserTVC»

Описание: Экран, представляющий собой профиль пользователя или класса, содержащий его расписание и дополнительную информацию (списки, ссылка, фото).

Функция: Принимает при переходе от предыдущего вид-контроллера пользователя типа `User` или класс типа `Class`. Основную площадь экрана занимает таблица, первая ячейка типа `UserTableViewCell` содержит информацию о пользователе, вторая (необязательная) ячейка типа `LinkTableViewCell` содержит кнопку

для перехода, остальные ячейки типа LessonTableViewCell содержат информацию об уроках.

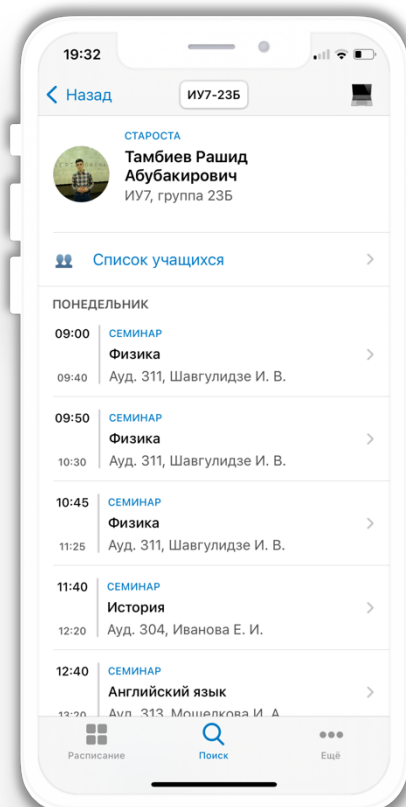


Рисунок 22. Вид UserTVC

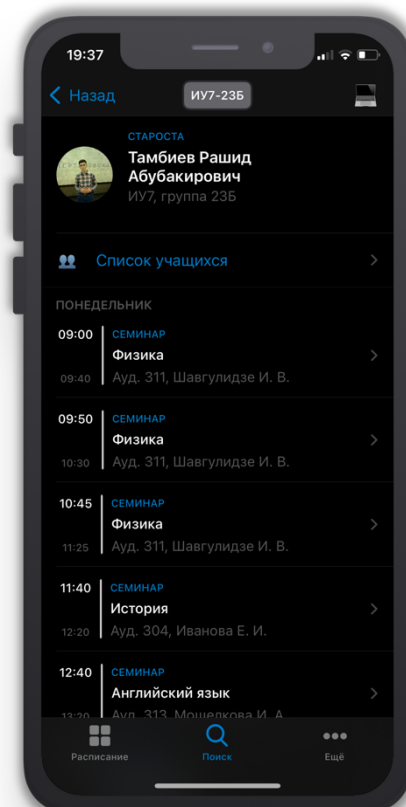


Рисунок 23. Вид UserTVC
(темная тема)

При нажатии на ячейку типа UserTableViewCell происходит переход к пользователю-классному руководителю для учебных классов и открытие фото профиля в противном случае.

Поведение при нажатии на LessonTableViewCell является аналогичным подобному в TimetableTVC.

Если у переданного контроллеру пользователя имеется ссылка, то она будет отображена в ячейке типа LinkTableViewCell, при нажатии на которую будет произведён переход в системный браузер.

Для учебных классов в UINavigationController доступен UISegmentedControl – выбор подгруппы. Ячейка типа LinkTableViewCell служит для вызова списка учеников выбранной подгруппы, при нажатии на которую происходит анимированный pop-up переход к SubgroupDetailsTVC.

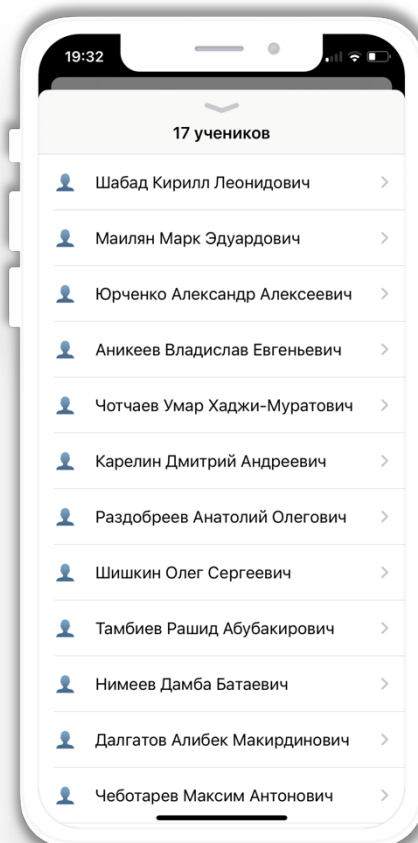
Если запрашиваемое расписание отсутствует в базе данных, то пользователь увидит соответствующее сообщение в таблице.

3.9.9 Контроллер «SubgroupDetailsTVC»

Описание: Данный экран открывается с логикой pop-up и содержит список учеников выбранной подгруппы. В заголовке указано количество учеников.

Функция: Принимает при переходе от предыдущего вид-контроллера список пользователей и delegate типа `UserCellDelegate`. Основную площадь экрана занимает таблица, каждая ячейка типа `SearchResultTableViewCell` содержит фамилию, имя и отчество пользователя.

При нажатии на ячейку типа `SearchResultTableViewCell`, происходит закрытие текущего pop-up, вызывается метод делегата `didSelect(_ user: User)`, далее переход к `UserTVC`.



3.10 Описание внутренних типов

3.10.1 Тип «User»

Описание: Репрезентация пользователя социальной сети.

Параметры: имя пользователя (`name : String`), группа (`group : String`), уникальный идентификатор пользователя (`id : String`), субъект или предмет (`subject : String`), необязательная ссылка на фото пользователя (`photo : String?`), необязательная личная ссылка пользователя (`link : String?`), статус пользователя (`status : Status`).

Инициализация: с помощью идентификатора пользователя (`uid : String`) и строкового словаря параметров пользователя (`row : [String:String]`).

Имя пользователя содержит ФИО пользователя.

Для учеников `group` используется для хранения класса (например: “10-Д”), а `subject` содержит номер подгруппы (например: “1”).

Для учителей `group` используется для хранения названия кафедры или группы преподавателей (например: “Кафедра "Основы математики и информатики" (СУНЦ-1)”), а `subject` содержит наименование должности (например: “Доцент кафедры СУНЦ-1, учитель математики”).

Типы `Teacher` и `Student` являются наследниками типа `User`.

Рисунок 24. Вид `SubgroupDetailsTVC`

3.10.2 Тип «Status»

Описание: Перечисляемый строковый тип для представления статуса пользователя.

Принимаемые значения: ученик (student), учитель (teacher), заблокирован (blocked).

Инициализация: blocked = “0”, student = “1”, teacher = “2”

Доступные переменные: Строковое описание статуса пользователя (title : String, например: “Доступ закрыт”, “Ученик”, “Учитель”), цветовое описание статуса пользователя (color : UIColor, для заблокированного – красный, для остальных – основной цвет приложения), директория в базе данных для данного статуса пользователя (group : String).

3.10.3 Тип «Class»

Описание: Репрезентация учебного класса в социальной сети.

Параметры: имя класса (name : String), уникальный идентификатор классного руководителя (leaderTeacherID : String), список подгрупп (subgroup : [Subgroup])

Инициализация: с помощью имени класса (name : String) и словаря параметров (info : Dictionary<String, Any>).

Пример имени класса: “10-Д”

3.10.4 Тип «Subgroup»

Описание: Репрезентация подгруппы в социальной сети.

Параметры: номер подгруппы (number : Subgroup.number), профиль класса (subject : Subject), список уникальных идентификаторов учеников (students : [String]).

Инициализация: повторяет каждый параметр структуры.

Subgroup.number – внутренний перечисляемый строковый тип, принимающий значения first, second, none.

3.10.5 Тип «Subject»

Описание: Перечисляемый строковый тип для представления профиля класса.

Принимаемые значения: физико-математический (fm), информационно-технологический (it), физико-химический (fh), социально-экономический (se), химико-биологический (hb), иной или отсутствует (none).

Инициализация: строковое значение повторяет наименование принимаемых значений.

3.10.6 Тип «Lesson»

Описание: Репрезентация урока в расписании пользователя социальной сети.

Параметры: наименование урока (`name : String`), номер урока (`number : Int`), вид урока (`type : LessonType`), место проведения занятия (`room : String`), уникальный идентификатор учителя (`teacher : String`), есть урок / нет урока (`flag : Bool`).

Инициализация: с помощью номера урока (`number : Int`), вида урока (`type : LessonType`) и строкового словаря параметров (`dictionary : Dictionary<String, String>`).

3.10.7 Тип «Day»

Описание: Является списком элементов типа `Lesson`.

3.10.8 Тип «Week»

Описание: Является списком элементов типа `Day`.

3.10.9 Тип «LessonType»

Описание: Перечисляемый строковый тип для представления вида урока.

Принимаемые значения: практика (`practice`), семинар (`seminar`), лекция (`lecture`).

Инициализация: `practice = "0"`, `seminar = "1"`, `lecture = "2"`

Доступные переменные: Строковое описание вида урока (`name : String`, например: "Практика", "Семинар", "Лекция").

3.10.10 Тип «Setting»

Описание: Репрезентация элемента настроек приложения.

Параметры: заголовок (`title : String`), строка перехода (`segue : String`), эмодзи-символ перед заголовком (`emoji : String`), флаг содержания ссылки (`isLink : Bool`).

Инициализация: повторяет каждый параметр структуры.

Строка перехода содержит наименование перехода в пользовательском интерфейсе если `isLink = false`, иначе – ссылку.

3.10.11 Тип «OtherProject»

Описание: Репрезентация “других проектов” в настройках.

Параметры: название (name : String), ссылка на изображение с иконкой (image : String), описание (description : String), ссылка для перехода (link : String).

Инициализация: с помощью строкового словаря параметров (dictionary : Dictionary<String, String>).

4 Технологическая и исследовательская часть

После анализа популярных мобильных платформ, их операционных систем, сред разработки на этих операционных системах и используемых языках программирования, было принято решение делать приложения индивидуально на каждую платформу. Аргументом для данного решения явился тот факт, что максимальный комфорт с точки зрения пользователя, включая производительность и узнаваемое для данной платформы поведение, возможно достичь только создавая приложения индивидуально под каждую платформу. Кроссплатформенные (мультиплатформенные) подходы выглядят как некоторый общий срез, где эксплуатационные характеристики унифицированы, что в данном случае отрицательно влияет на «полезность» и «удобство», которые я стремился достичь в разработке.

Для реализации в первую очередь, были выбраны платформы iPhone + iPad и операционная система iOS, так как у меня был опыт мобильной разработки на них. Для данных платформ высокопроизводительным и современным подходом к созданию приложений является язык Swift. Согласно Apple Human Interface Guidelines, конечный программный продукт должен покрывать 80% аудитории. В связи с этим была выбрана целевая iOS версии 11.0 и выше, так как по статистике более 80% пользователей имеют версию старше.

Для реализации хранилища данных была выбрана реляционная СУБД PostgreSQL, где основой логики работы является реляционная алгебра. Если посмотреть книги, посвященные базам данных (особенно университетские), то они в обязательном порядке дают её, и иногда в больших количествах, т. е. популярность и бесплатность сделали первый шаг выбора.

Второй шаг заключается в следующем: существует множество разных способов представить одни и те же данные. Один из первых широко используемых способов — иерархическая модель. В такой модели данные представлены в виде дерева, где дочерние элементы находятся в зависимости от родительских. С помощью иерархической модели можно представить и бизнес-области. К примеру, в предметной области учебного заведения есть такие понятия, как преподаватель и лекция. Причём, у одного преподавателя может быть множество лекций, что сразу определяет структуру дерева, где вершиной становится преподаватель, а его детьми — лекции. Подобным образом структурируются и все остальные части.

Проблемы начинаются, когда у одного ученика может быть несколько родителей. В современных сервисах такси есть возможность оплачивать счёт за такси совместно — это значит, что у одного заказа сразу несколько клиентов. Как быть в такой ситуации? К сожалению, иерархическая модель не может предложить хорошего решения данной задачи. Придётся создавать параллельные деревья, в которых появится дублирование данных. Тем не менее, СУБД PostgreSQL является хорошим выбором в данном применении.

Был проведен устный опрос среди учащихся 9-11 классов нашего лицея, где предметом исследования являлось выяснение «полезного» функционала для платформы и приложения. Итогом исследования являются имеющиеся, реализованные функции.

5 Заключение

Поставленные цели были достигнуты в полном объёме:

- **Разработана платформа учебного заведения.** На базе данной платформы было разработано мобильное приложение «Мой МГТУ», обладающее, согласно результатам опроса, функционалом первой необходимости: расписание, профили, подгруппы, поиск с фильтрами, уведомления о занятиях. Для разработчиков доступны библиотеки на языках программирования Swift и Python.
- **Расширение платформы благодаря краудсорсингу.** Ученики-разработчики из лицея 1580 получили доступ к EduApp API и создали свои производные сервисы и продукты, которые в дальнейшем могут быть использованы в инфраструктуре МГТУ без дополнительных изменений в коде.

6 Приложение: список литературы

1. Арлоу Д., Нейштадт И. «UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование», 2-е издание. — Пер. с англ. — СПб: Символ Плюс, 2007. — 624 с., ил.
2. Руководство по Enterprise Architect, англ. - <https://sparxsystems.com/resources/tutorials/uml2/index.html>
3. iOS Version Market Share, англ. - <https://gs.statcounter.com/ios-version-market-share/>
4. Саммерфилд М. «Программирование на Python 3. Подробное руководство». — Пер. с англ. — СПб.: Символ Плюс, 2009. — 608 с., ил.

5. Ганс-Юрген Шениг: «PostgreSQL 11. Мастерство разработки». - Пер. с англ. Слинкина А. А. - М.: ДМК Пресс, 2019. - 352с.: ил.
6. Солдатова Г. В., Зотова Е. Ю., Чекалина А. И., Гостимская О. С. «Пойманные одной сетью: социально-психологическое исследование представлений детей и взрослых об интернете», МГУ - Под ред. Г. В. Солдатовой. — М., 2011. — 176 с.