

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»

98

регистрационный номер

ИУ «Информатика и системы управления»

название факультета

ИУ7 «Программное обеспечение ЭВМ и информационные технологии»

название кафедры

Мобильное приложение «Мой Лицей».

Социальная сеть образовательного учреждения.

название работы

Автор:

Козодой Андрей Александрович

фамилия, имя, отчество

ГБОУ школа №1580, 10-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

Мобильное приложение «Мой Лицей».
Социальная сеть образовательного учреждения.

Аннотация

Мобильное приложение «Мой Лицей» – социальная сеть образовательного учреждения, созданная с целью объединения участников учебного процесса, внедрения собственных технологий для автоматизации; способствует внутреннему техническому развитию и сообщению. Цели проекта рассматривают мобильное приложение «Мой Лицей» как один из промежуточных шагов более общего проекта.

Документ описывает лишь промежуточный шаг - дизайн и архитектуру мобильного приложения «Мой Лицей». Это действующее приложение, которое доступно для школьного бета-тестирования в среде TestFlight.

В основу проектирования технического решения был заложен принцип согласования и обсуждения дизайна продукта и его функционала со всеми участниками учебного процесса. Клиентская часть клиент-серверного приложения была разработана на языке программирования Swift. Для реализации блока новостей и объявлений в виде «историй» был разработан модуль Warbler.

На текущем этапе проекта, с помощью мобильного приложения, можно узнать расписание классов, учеников, учителей. Новости и объявления образовательного учреждения представлены в привычном по формату социальных сетей «историях».

Это первый опыт в сфере автоматизации учебных процессов внутри 1580, не имеющий конкурентов. Проект создан для того, чтобы технологии стали открытыми и доступными для развития.

Предусматривается дальнейшая работа над проектом, как дальнейшее исполнение плана, где глобальная миссия - предоставить платформу социальной сети для ВСЕХ образовательных учреждений.

Оглавление

1	Введение	4
2	Философия продукта	6
3	Модель / Конструкторская часть.....	7
3.1	Применяемые тех решения	7
3.2	Формирование базы данных	10
3.2.1	Получение списка учителей.....	10
3.2.2	Получение списка учеников	11
3.2.3	Получение расписания	12
3.3	Схема данных	13
3.4	Автоматизация исполнения скриптов.....	16
3.5	Библиотека Warbler.....	16
3.5.1	Описание реализации	17
3.5.2	Поддерживаемые жесты и управление	19
3.6	Push-уведомления	20
3.7	Наследники «UIViewController».....	21
3.7.1	Контроллер «LoadingVC»	22
3.7.2	Контроллер «AdVC».....	22
3.7.3	Контроллер «LoginVC»	23
3.7.4	Контроллер «TimetableTVC».....	23
3.7.5	Контроллер «LessonDetailsTVC»	24
3.7.6	Контроллер «ProfileTVC»	25
3.7.7	Контроллер «SearchVC».....	26
3.7.8	Контроллер «UserTVC».....	27
3.7.9	Контроллер «SubgroupDetailsTVC»	29
3.8	Описание внутренних типов.....	29
3.8.1	Тип «User»	29

3.8.2	Тип «Status»	30
3.8.3	Тип «Class»	30
3.8.4	Тип «Subgroup»	30
3.8.5	Тип «Subject»	31
3.8.6	Тип «Lesson»	31
3.8.7	Тип «Day»	31
3.8.8	Тип «Week»	31
3.8.9	Тип «LessonType»	32
3.8.10	Тип «Setting»	32
3.8.11	Тип «OtherProject»	32
4	Технологическая и исследовательская часть	32
5	Заключение	34
6	Приложение: список литературы	35

Список иллюстраций

Рисунок 1. Системная архитектура «Мой Лицей».....	7
Рисунок 2. Модель экранных форм в виде Storyboard	9
Рисунок 3. Информация о педагогическом составе	10
Рисунок 4. Информация об административном составе	11
Рисунок 5. Информация о расписании.....	12
Рисунок 6. Внутренняя структура расписания.....	13
Рисунок 7. Диаграмма разворачивания базы данных.....	14
Рисунок 8. Внешний вид контроллера Warbler.....	17
Рисунок 9. Управление уведомлениями	20
Рисунок 10. Задание времени напоминания.....	20
Рисунок 11. Внешний вид уведомления	20
Рисунок 12. Наследники «UIViewController»	21
Рисунок 13. Вид LoadingVC.....	22
Рисунок 14. Вид AdVC (темная тема).....	22
Рисунок 15. Вид TimetableTVC	23
Рисунок 16. Вид LessonDetailsTVC	25
Рисунок 17. Вид LessonDetailsTVC (темная тема).....	25
Рисунок 18. Вид ProfileTVC.....	26
Рисунок 19. Вид ProfileTVC (переход по ссылке)	26
Рисунок 20. Вид SearchVC	27
Рисунок 21. Вид SearchVC (результаты)	27
Рисунок 22. Вид UserTVC	28
Рисунок 23. Вид UserTVC (темная тема).....	28
Рисунок 24. Вид SubgroupDetailsTVC.....	29

1 Введение

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

В самом начале учебного года, я поставил перед собой задачу сделать «полезное» приложение для нашей школы. После проведения опросов сверстников, анализа собранных материалов и сложившейся ситуации, был выявлен срез «полезности» — это решение существующих проблем с динамически изменяемым расписанием занятий и поиск друзей из параллельных классов.

Технологии, внедрённые департаментом образования, представляют решение этой проблемы очень условно, где веб-сайт лицея представляет информацию недостаточно современно и удобно для ежедневного использования и поиска. Возникла идея создания мобильного приложения «Мой Лицей», решающего эту задачу.

После начала работ над проектом мобильного приложения «Мой Лицей», появились мысли его расширения – создание социальной сети. 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 школьников. 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. Школы не имеют достаточных ресурсов для создания собственной платформы социальных коммуникаций, а представляемые школам города Москвы платформы ограничены по функционалу и не имеют возможность расширения

Я уверен, что также в рамках проекта 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.

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 **Философия продукта**

Проект «Мой Лицей» ставит перед собой следующие общие задачи:

1. **Создать платформу социальной сети учебного заведения.** Чтобы получить подобную социальную сеть, итоговый продукт должен обладать «полезностью», в первую очередь. Т.е. он должен быть полезен всем участникам – в идеале много раз ежедневно. Так же не следует упускать из виду современный дизайн и узнаваемость, присутствующих в известных социальных сетях
2. **Использовать коллективные ресурсы учебного заведения.** С помощью краудсорсинга создать финальную версию продукта - платформу социальной сети учебного заведения, где коллективом участников (в идеале - весь лицей) выполняются следующие задачи проекта: Производство тестирования и сбор информации, обсуждение инноваций

Для решения данных задач был выбран следующий план:

1. Создать мобильное приложение «Мой Лицей» (описывается ниже), обладающего «полезностью» (т.е. проблема расписания и поиска информации является актуальной и полезной, а самое главное - нужной)
2. Создать описание API для целей развития
3. Усилиями краудсорсинга портировать приложение на мобильную платформу Android
4. Усилиями краудсорсинга добавить элементы социальных сетей
5. Выпустить платформу для всех школ

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) специализированной социальной сети.

3 Модель / Конструкторская часть

3.1 Применяемые тех решения

Так как основное использование приложения «Мой Лицей» предусматривается в режиме «онлайн», то была выбрана классическая двухуровневая системная архитектура, с выделенными фронтендом и бэкендом (см. Рисунок 1).

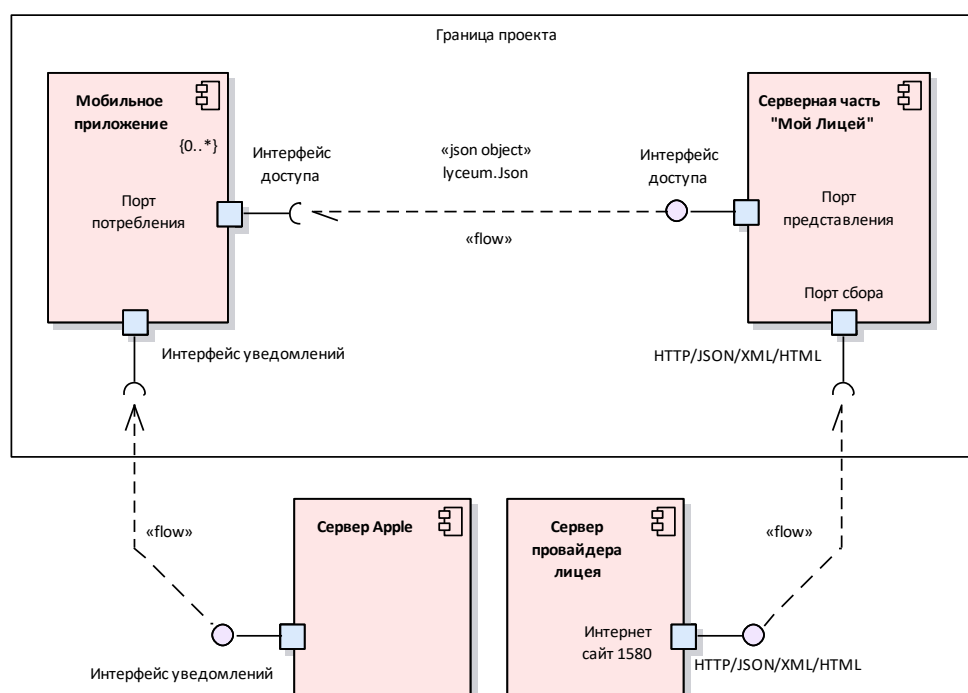


Рисунок 1. Системная архитектура «Мой Лицей»¹

Данная архитектура рекомендуется для мобильных приложений, работающих с данными, где в качестве фронтэнда выступает мобильное приложение «Мой Лицей» непосредственно, а бэкенд выполняет функции хранилища данных и исполняет скрипты автоматизации по импорту (обновлению) данных.

В качестве хранилища данных была выбрана NoSQL СУБД MongoDB, с доступом к данным в формате JSON (возможно использование любой JSON ориентированной СУБД).

Метрика «Мой Лицей»	Мобильное приложение (фронтэнд)	Серверная часть (бэкенд)
Язык программирования	Swift 5	Python 3.6
Среда разработки	XCode	PyCharm CE

¹ Используется среда моделирования на языке UML2 Enterprise Architect

Метрика «Мой Лицей»	Мобильное приложение (фронтэнд)	Серверная часть (бэкенд)
СУБД	iOS UserDefaults	NoSQL, JSON (MongoDB/Firebase)
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	7,9 МБ - исполнимый код 1.1 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 «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 UIViewController и переопределить или добавить требуемые свойства.

Для мобильного приложения “Мой Лицей”, экранные формы-наследники UIViewController 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.

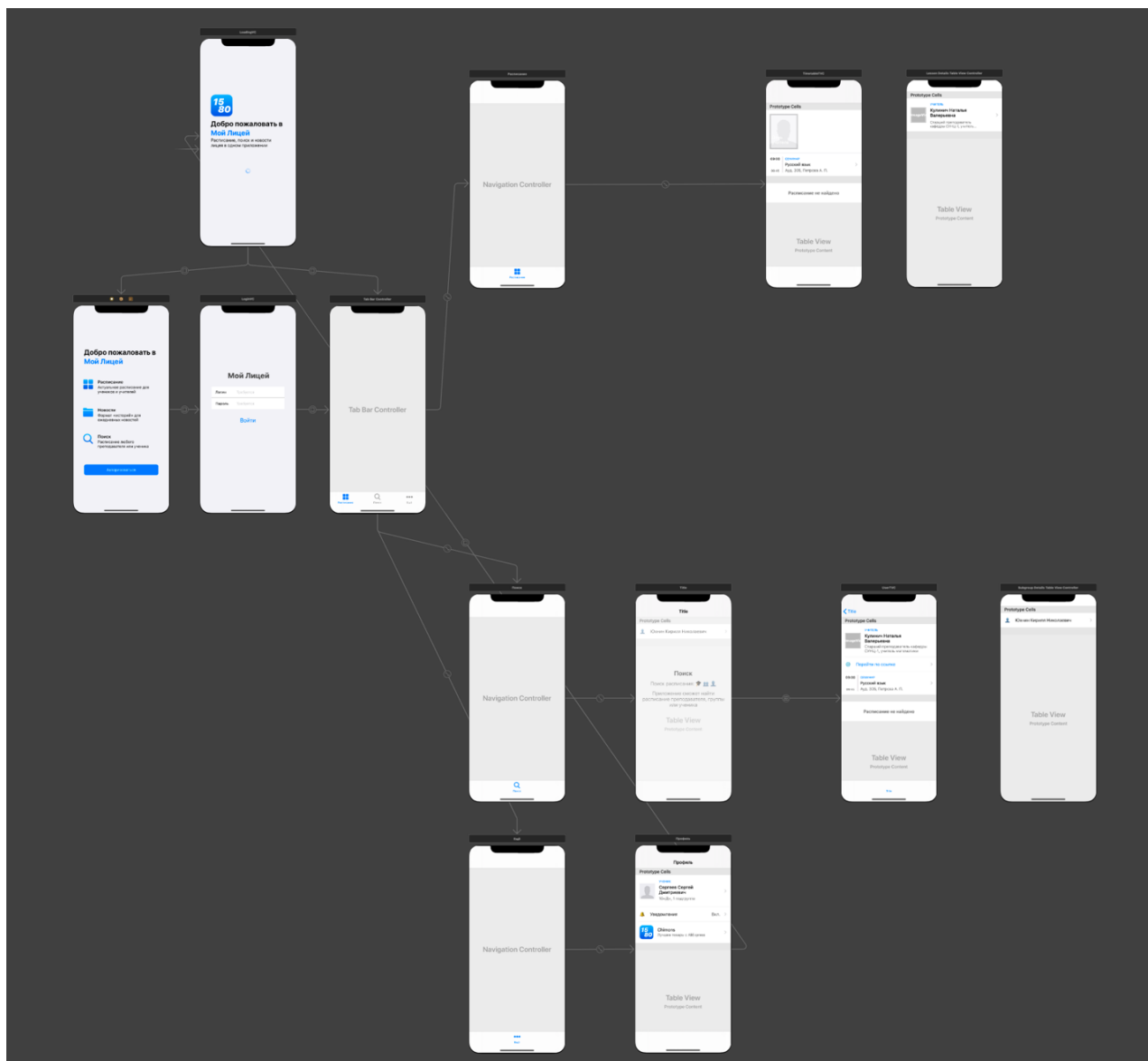


Рисунок 2. Модель экранных форм в виде Storyboard

Ввиду отсутствия готовой библиотеки показа (просмотра) Stories¹ в узнаваемом формате, т.е. который используется известными социальными сетями, было принято решение создать данную библиотеку самостоятельно. Данной библиотеке было дано название Warbler.

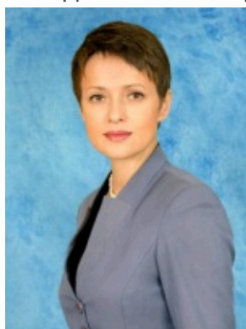
¹ Характерная функциональная черта Instagramи, WhatsUp и тд, позволяющая публиковать короткие видео или фото с теми событиями, которые происходят прямо сейчас. Дополнительно, можно указать ссылку, которая откроется при «свайпе» вверх

3.2 Формирование базы данных

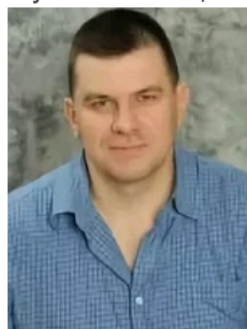
3.2.1 Получение списка учителей

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



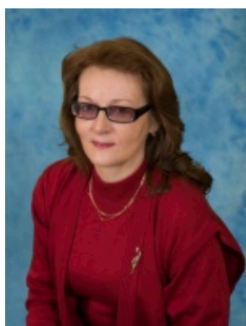
Богатин Андрей Аркадьевич
Учитель истории



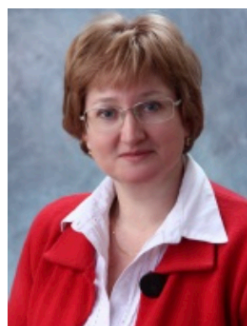
Карпец Федор Владимирович
Учитель истории



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.

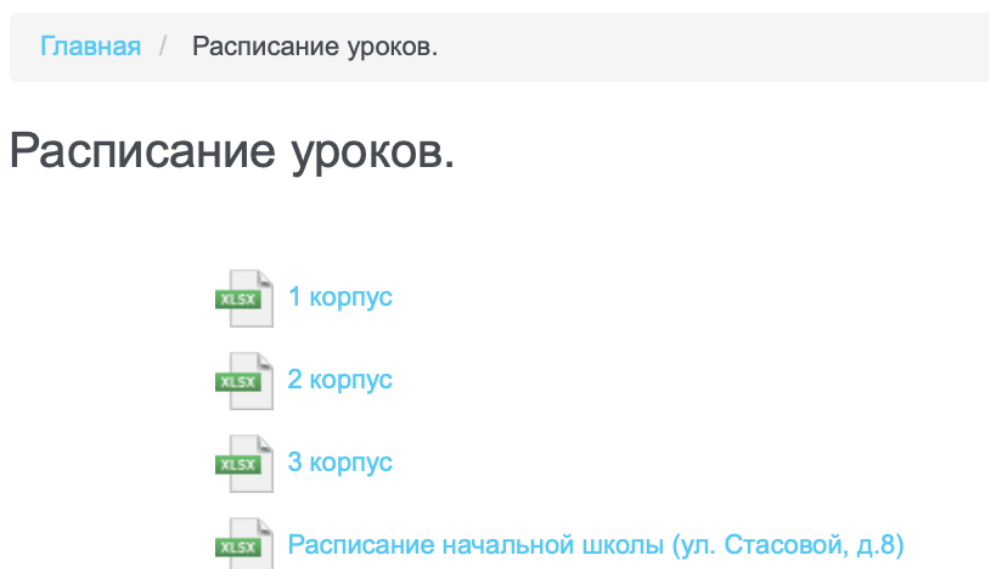


Рисунок 5. Информация о расписании

Дни	Уроки	10д1		10д2		10е1		10е2
Понедельник	1			Физика Шавгулдзе И.В.	311	Физика Калашникова А.А.	312	Английский язык Соловьева Д.А.
	2	Математика А Кулинич Н.В.	106	Физика Шавгулдзе И.В.	311	Физика Калашникова А.А.	312	Английский язык Соловьева Д.А.
	3	Русский язык Петрова А.П.	305	Физика Шавгулдзе 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 Мошелкова 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
ИКТ	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 – OpenPxl.

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

3.3 Схема данных

Итоговое решение было построено на NoSQL СУБД Firebase. Предпочтение данной СУБД было отдано потому, что простота ее использования, а также ее доступность в режиме «онлайн» экстремально высоки. Более того, для подобных применений её использование бесплатно.

Выбранное решение не требует поддержки выделенного (обычно - платного) сервера. А разработанные скрипты по экспорту данных могут быть размещены непосредственно на сервере провайдера лица. Тем самым, техническое развитие приложения сводится к расширению его функций.

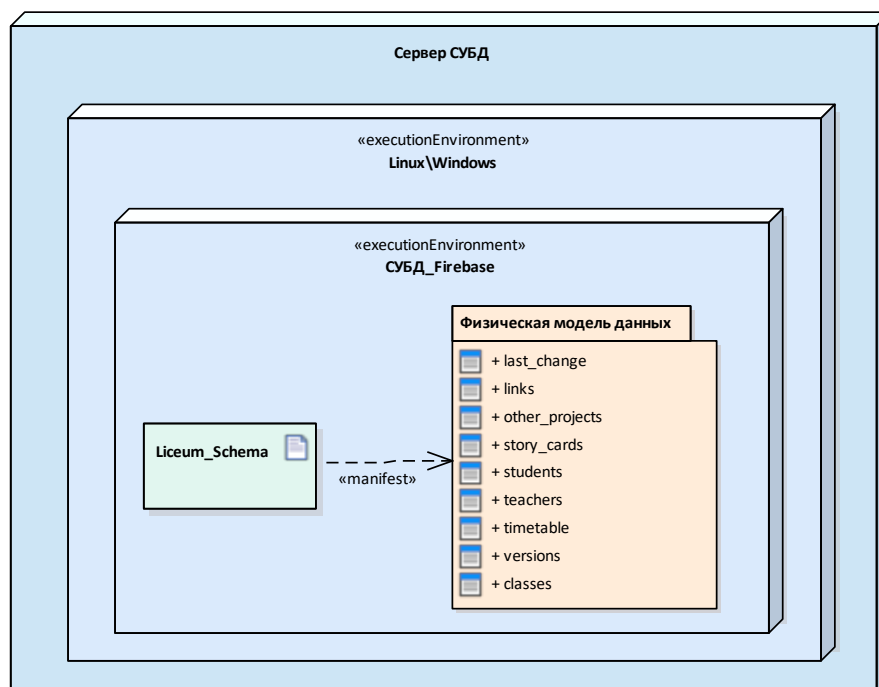


Рисунок 7. Диаграмма разворачивания базы данных¹

Элемент	Описание	Структура JSON данных
versions	Используется для проверки версии при запуске	<pre>"versions" : { "client" : { "build" : "version_num", "version" : "version_num" } }</pre>
last_change	Содержит временные метки изменения основных коллекций, так как они кэшируются на стороне мобильного приложения	<pre>"last_change" : { "classes" : timestamp, "students" : timestamp, "teachers" : timestamp }</pre>
links	Вспомогательная структура для вывода адреса технической поддержки в интернете	<pre>"links": { "support": "https://vk.com/stixirl", "terms": "https://stix.cc" }</pre>
other_projects	Вспомогательная структура для вывода адресов других проектов разработчика	<pre>"other_projects": [{ "description": "описание проекта", "image": "url", "link": " url ", "name": "наименование проекта" }, ...]</pre>
Коллекция classes	Основная коллекция классов	<pre>"classes": { "class_id": { "subgroup": { "first": {</pre>

¹ Используется среда моделирования на языке UML2 Enterprise Architect

Элемент	Описание	Структура JSON данных
		<pre> "students": ["student_id", ...], "subject": "subject_id" }, "second": { "students": ["student_id", ...], "subject": "subject_id" } }, "teacher_id": "teacher_id" }, ... </pre>
Коллекция students	Основная коллекция учеников	<pre> "students" : { "student_id" : { "group" : "group_id", "link" : "link_id", "name" : "ФИО", "photo" : "photo_url", "status" : "status_id", "subject" : "subject_id" } } ... </pre>
Коллекция teachers	Основная коллекция учителей	<pre> "teachers" : { "teacher_id" : { "group" : "наименование группы", "link" : "url", "name" : "ФИО", "photo" : "photo_url", "status" : "status_id", "subject" : "наименование предмета" } } ... </pre>
Коллекция timetable	Коллекция расписания занятий	<pre> "timetable": { "class_id": [[{ "flag": "flag_id", "name": "наименование предмета", "room": "номер аудитории", "teacher": "teacher_id" },],], ... </pre>
Коллекция Stories	Коллекция социальных элементов (данные библиотеки Warbler)	<pre> "story_cards": { "count": количество элементов, "stories": [{ "id": "story_id", "info": { "hex": "цвет фона", "id": "идентификатор", "picture": "picture_url", "title": "заголовок" } }] } </pre>

Элемент	Описание	Структура JSON данных
		<pre> }, "snap": [{ "id": "snap_id", "mime_type": "тип содержимого", "redirect_url": "url", "text": "заголовок", "url": "url" },], "snap_count": количество snaps },] } </pre>

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 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 Swift, специально разработанная для целей проекта и распространяющаяся в открытом доступе с помощью GitHub и CocoaPods для установки в среду разработки.

3.5.1 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.
- **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

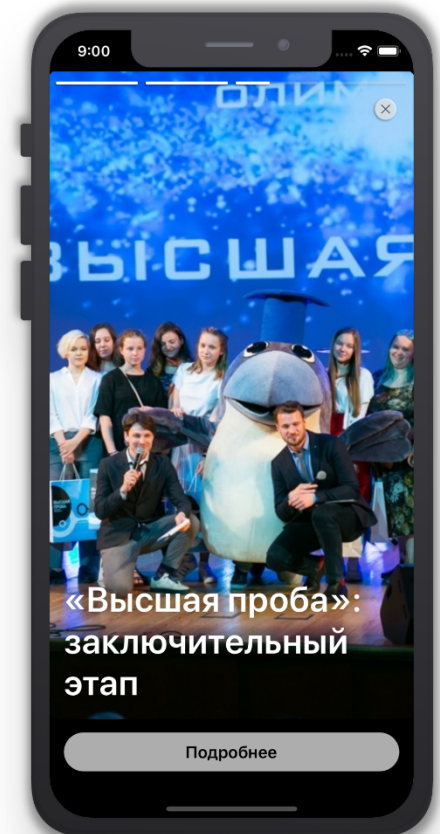


Рисунок 8. Внешний вид контроллера Warbler

(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.

3.5.2 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

3.6 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).

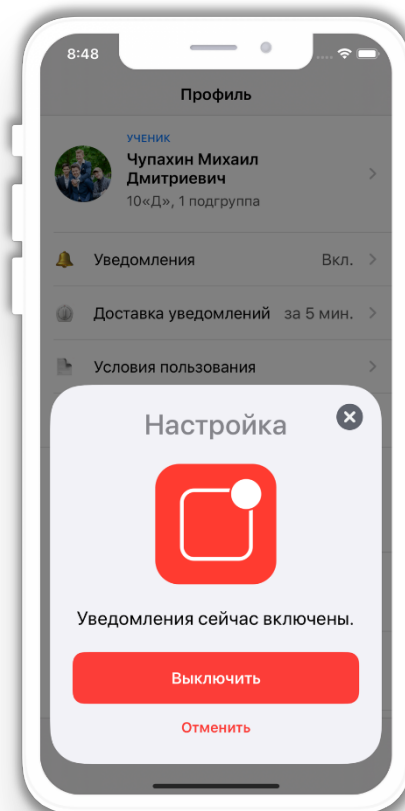


Рисунок 9. Управление уведомлениями

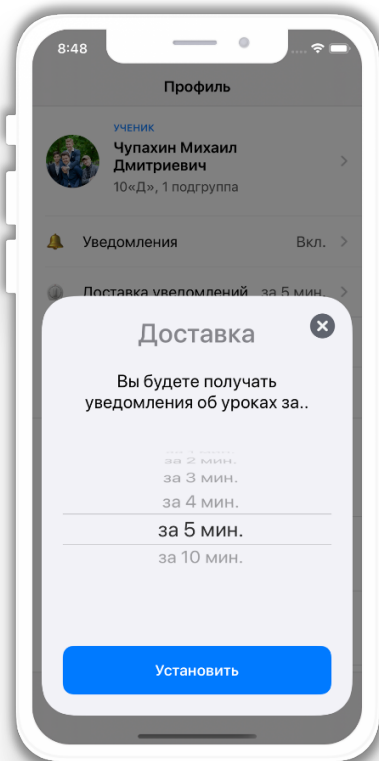


Рисунок 10. Задание времени напоминания

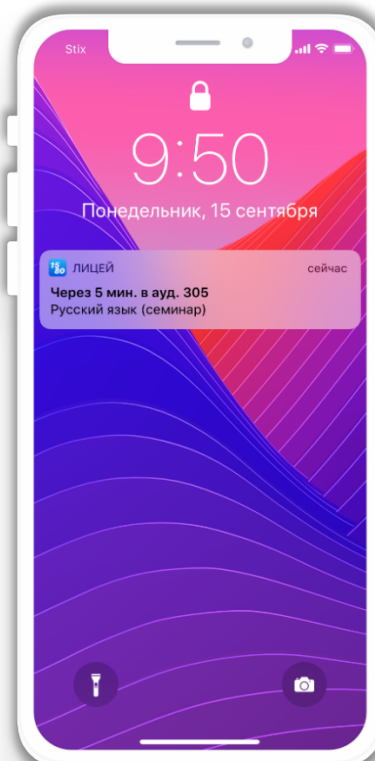


Рисунок 11. Внешний вид уведомления

3.7 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.

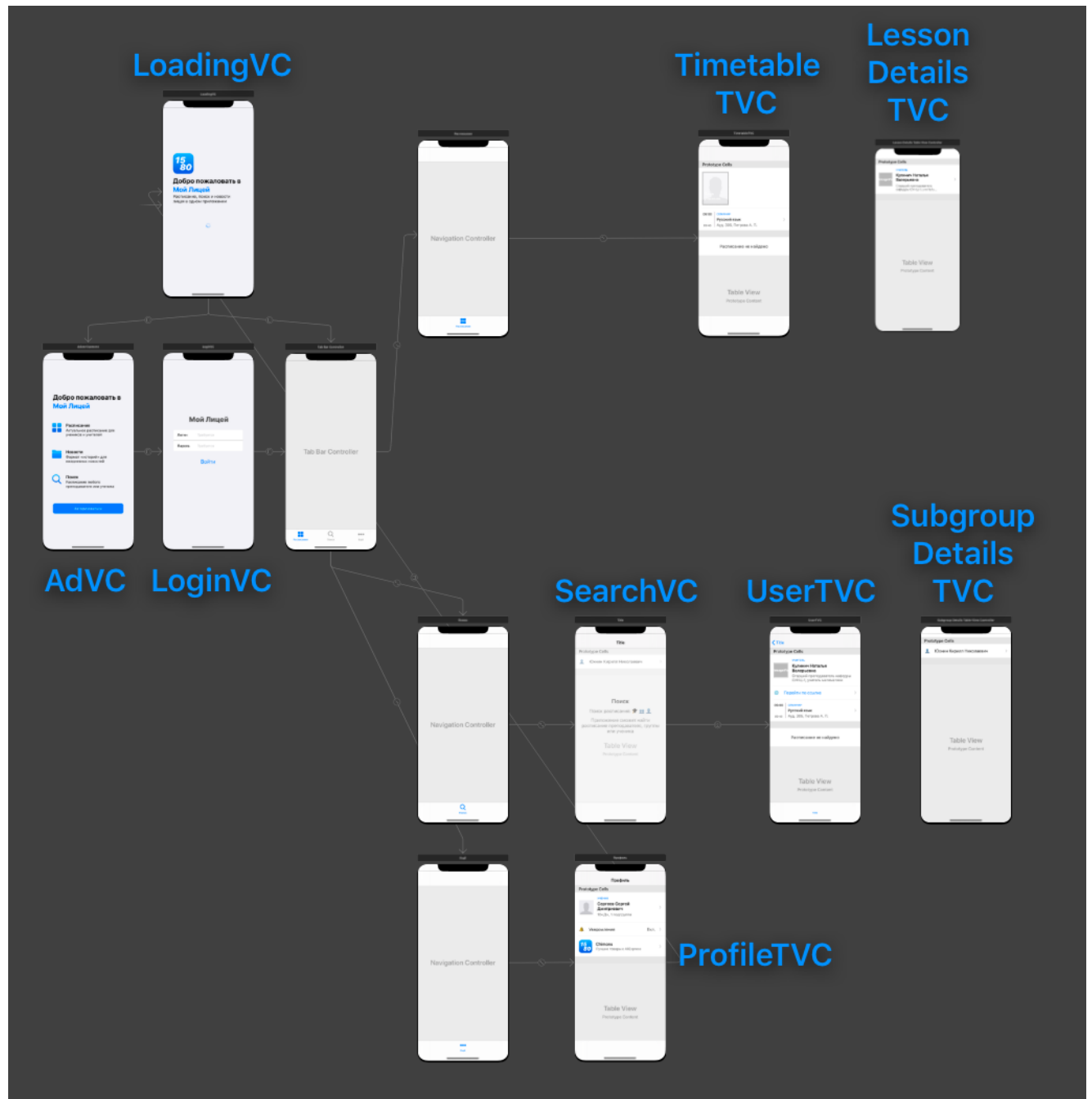
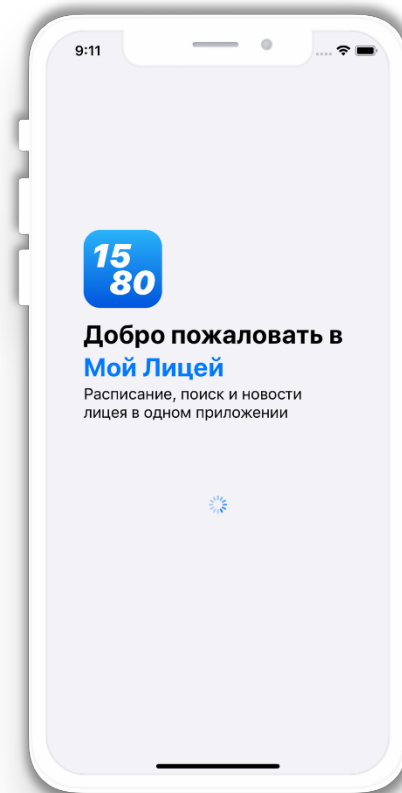


Рисунок 12. Наследники «UIViewController»

3.7.1 Контроллер «LoadingVC»

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

Функция: показывает анимированный UIActivityIndicator 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 ани-



мированный переход к TimetableTVC.

Рисунок 13. Вид LoadingVC

В случае возникновения любой ошибки, пользователь будет оповещён всплывающим сообщением.

3.7.2 Контроллер «AdVC»

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

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

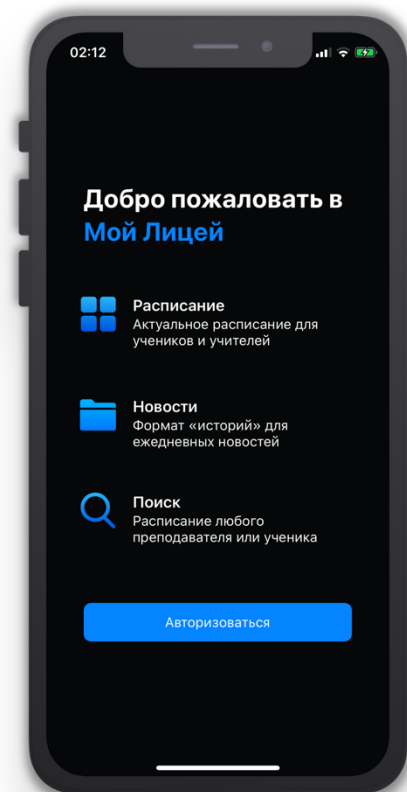


Рисунок 14. Вид AdVC (темная тема)

3.7.3 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.

3.7.4 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

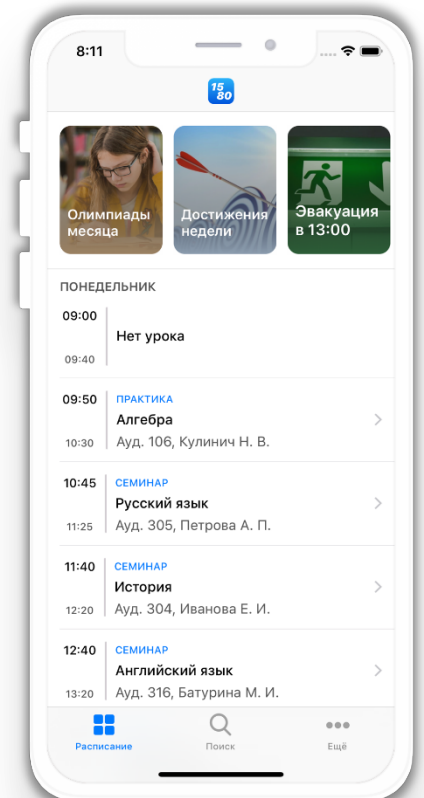


Рисунок 15. Вид TimetableTVC

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).

3.7.5 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.

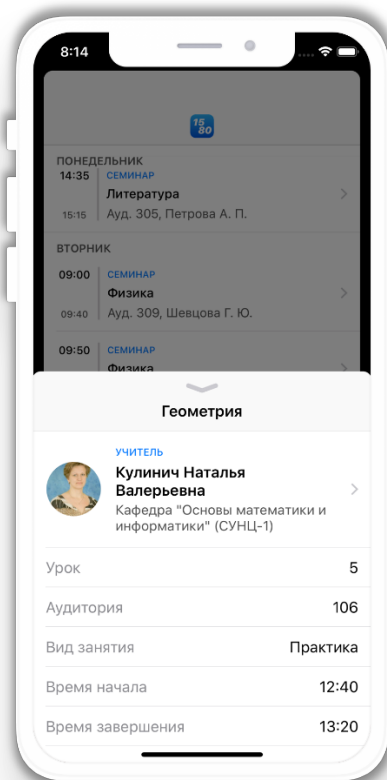


Рисунок 16. Вид LessonDetailsTVC

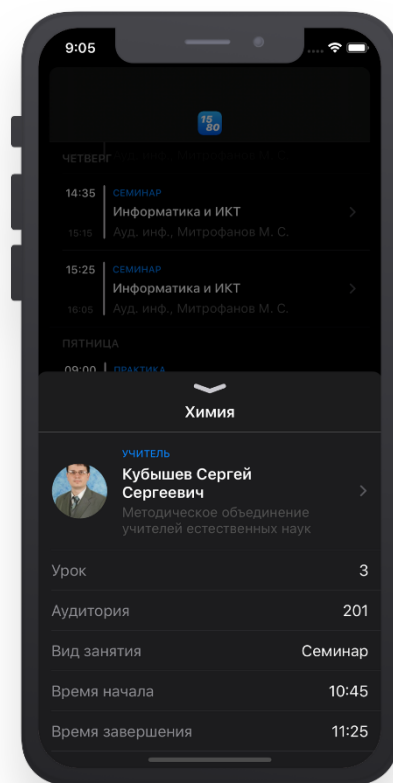


Рисунок 17. Вид LessonDetailsTVC
(темная тема)

3.7.6 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.

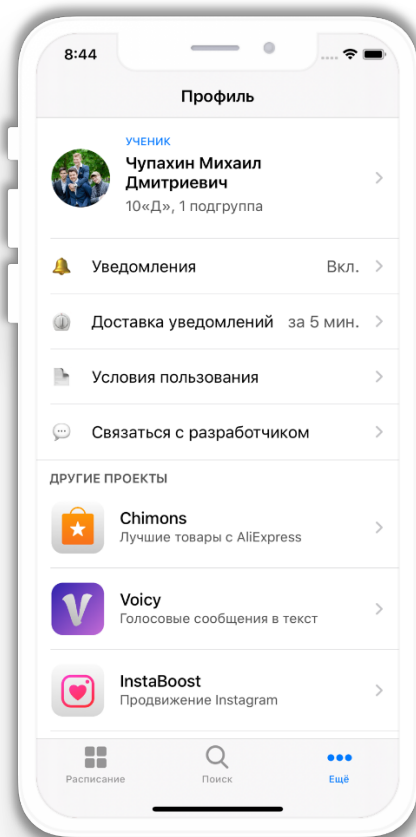


Рисунок 18. Вид ProfileTVC

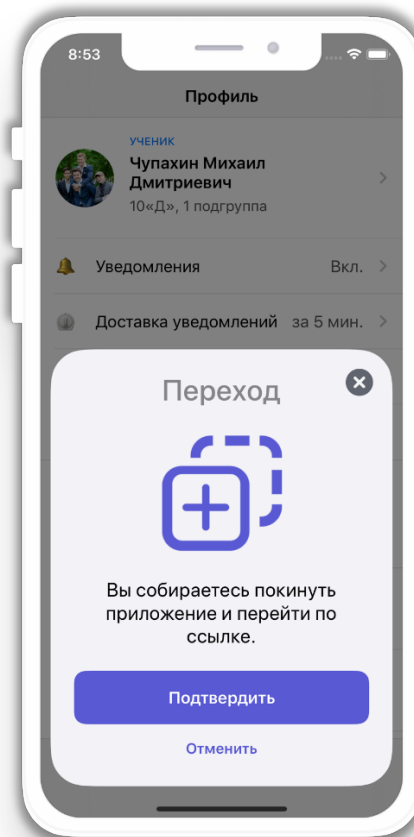


Рисунок 19. Вид ProfileTVC
(переход по ссылке)

3.7.7 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`.

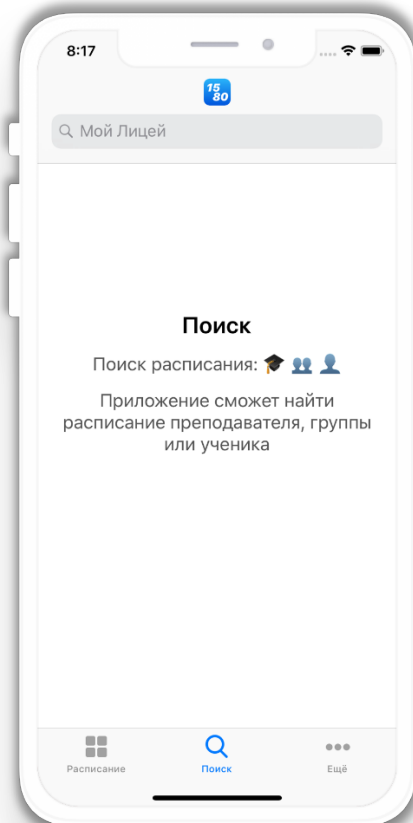


Рисунок 20. Вид SearchVC

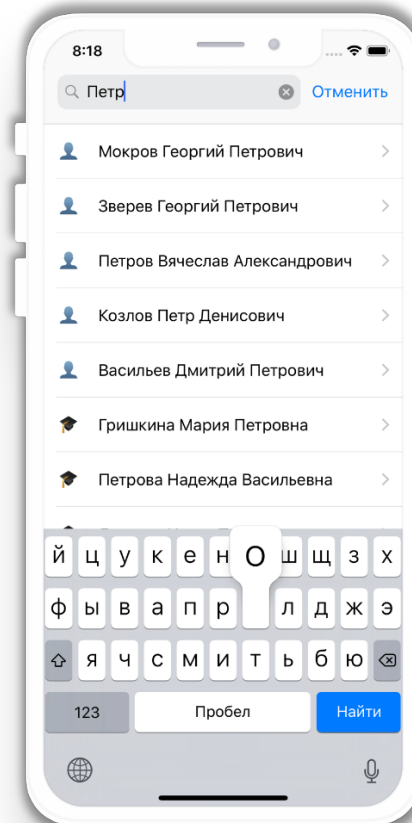


Рисунок 21. Вид SearchVC (результаты)

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

3.7.8 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.

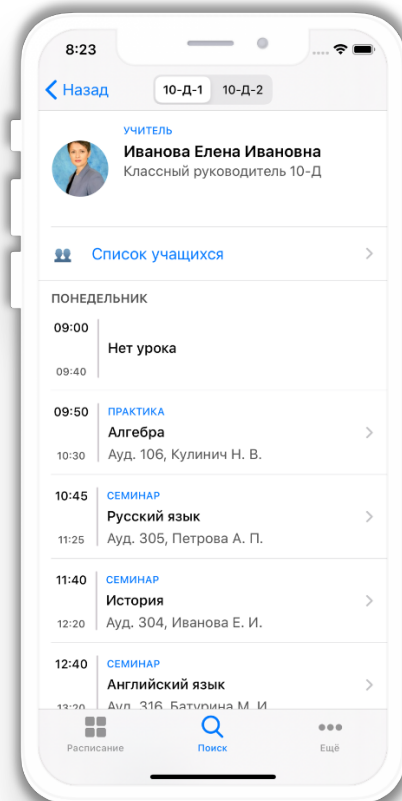


Рисунок 22. Вид UserTVC

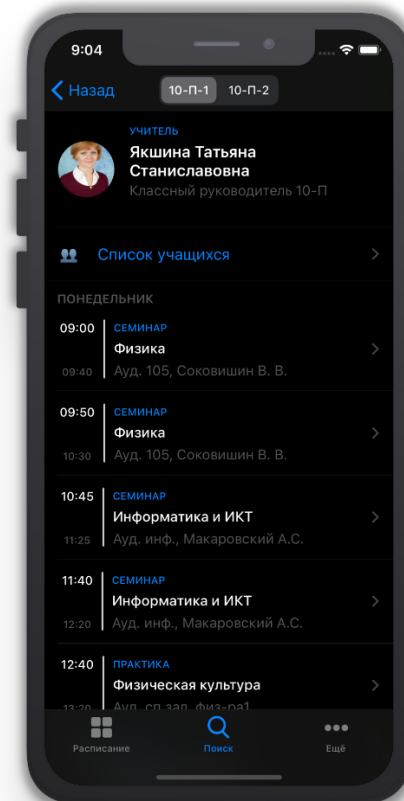


Рисунок 23. Вид UserTVC
(темная тема)

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 пор-уп переход к `SubgroupDetailsTVC`.

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

3.7.9 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`.

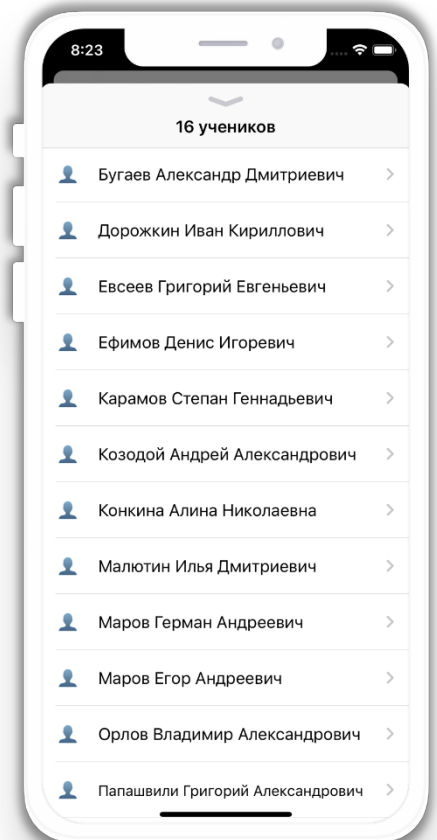


Рисунок 24. Вид SubgroupDetailsTVC

3.8 Описание внутренних типов

3.8.1 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.

3.8.2 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).

3.8.3 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”

3.8.4 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`.

3.8.5 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.

3.8.6 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>`).

3.8.7 Тип «Day»

Описание: Является списком элементов типа `Lesson`.

3.8.8 Тип «Week»

Описание: Является списком элементов типа `Day`.

3.8.9 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”).

3.8.10 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.

3.8.11 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 были навыки мобильной разработки. 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 на базе JSON документов была выбрана СУБД MongoDB (первая реализация выполнена на СУБД Firebase из-за присутствия «онлайн», доступности и бесплатности). Основа данных на базе JSON документов, во-первых, решает поставленные задачи, а во-вторых, позволит в будущем прямую интеграцию с JSON сервисами сайта, где от администратора сайта требуется только включить на требуемых страницах программные модули JavaScript. Эти страницы можно будет интегрировать в приложение напрямую.

На данном этапе реализации проекта, где целью является создание мобильного приложения «Мой Лицей», были проведены дополнительные исследования:

1. Проведен устный опрос среди учащихся 9-11 классов нашего лицея, где предметом исследования являлось выяснение «полезного» функционала для приложения. Итогом исследования являются имеющиеся, реализованные функции приложения
2. Проведено бета-тестирование созданного мобильного приложения «Мой Лицей», среди учащихся 10-х классов в среде TestFlight. Собрана обратная связь, исправлены ошибки и введён новый функционал. Общий индекс удовлетворённости составил 82%.

5 Заключение

Поставленные цели проекта по разработке мобильного приложения «Мой Лицей» (первый шаг плана) были достигнуты в полном объеме и включают:

- Функции «Расписания»
 - Подробная информация о каждом уроке: день недели, аудитория, время начала и завершения урока, вид занятия
 - Формат Stories для новостей и объявлений
- Функции «Профиля»
 - Статусы профилей: ученик (+староста), учитель, «заблокирован»
 - Расписание каждого участника или группы учебного процесса доступно через поиск
- Функции «Классы и Подгруппы»
 - Ссылка на профиль классного руководителя
 - Списки учеников и расписание каждой подгруппы
 - Рор-уп и управление в стиле встроенных в iOS приложений
- Функции «Поиск»
 - Поиск расписания учителей, подгрупп, учеников
 - Решает проблему поиска друзей, учителей, места сдачи материалов, времени пребывания участников учебного процесса
- Функции «Уведомлений и напоминаний»
- Функции «Настройки»
 - Смена фото профиля
 - Обратная форма в один клик
 - Настройка уведомлений
 - Продвижение других приложений

По результатам проведенного бета-тестирования мобильного приложения «Мой Лицей» был достигнут индекс удовлетворенности 82%, что показывает высокий уровень достижения ожиданий учащихся нашего лицея.

Для дальнейшего развития выбран следующий план шагов:

1. Создать описание API для целей развития и расширения команды разработки
2. Усилиями краудсорсинга портировать мобильное приложение «Мой Лицей» на мобильную платформу Android
3. Усилиями краудсорсинга добавить элементы социальных сетей
4. Выпустить платформу для всех школ

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. Кайл Бэнкер «MongoDB в действии». - Пер. с англ. Слинкина А. А. - М.: ДМК Пресс, 2012. - 394с.: 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.