

Кафедра математической теории упругости и биомеханики

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Прохорова Евгения Сергеевича

Бессонов Л.В.

Коссович Л.Ю.

Саратов 2022

Введение

В наше время - в век технологий и инноваций, потребность общества в удобстве и комфорте растёт с каждым годом. У каждого человека, будь то студент или преподаватель - есть мобильный телефон, с помощью которого можно не только коммуницировать с обществом, но и оптимизировать работу или обучение.

Использование веб-сайтов теряет свою актуальность, ведь на смену им пришли мобильные приложения. Их удобства заключаются в том, что вся информация отобрана под конкретного пользователя, что ускоряет процесс получения данных.

Расписание занятий - то, в чем нуждаются студенты в первую очередь. Поэтому актуальной задачей для всех ВУЗов страны, является создание электронной доски с расписанием, которая всегда будет рядом - в смартфоне студента.

Именно поэтому, я решил создать более удобный способ получения расписания занятий в Саратовском Государственном Университете им. Н.Г. Чернышевского и сделал мобильное приложение под мобильную ОС IOS на языке Swift.

Язык программирования Swift создан прежде всего для разработки приложений на iOS и macOS. Он относится к языкам программирования общего назначения, то есть на нем можно разрабатывать не только мобильные приложения, но и программы для десктопных платформ.

Целью работы является разработка мобильного приложения СГУ для IOS.

Для реализации цели данной работы были выполнены следующие **задачи**:

1. Анализ сайта СГУ им. Н.Г. Чернышевского
2. Анализ API СГУ им. Н.Г. Чернышевского
3. Разработка IOS-приложения

Объектами исследования являются язык программирования Swift, расписание СГУ им. Н.Г. Чернышевского и IDE XCode.

Научная новизна. Впервые было реализовано приложение с расписанием для СГУ им. Н.Г. Чернышевского.

Практическая значимость. Студентам и преподавателям будет намного проще смотреть расписание занятий.

Структура и объем работы. Выпускная квалификационная работа состоит из введения, 3 глав, заключения, списка использованных источников.

Основное содержание

Во введении обосновывается актуальность создания приложения СГУ.

В первой главе Проводился анализ предметной области, знакомство с ЯП Swift и прототипирование экранов приложения.

Расписание Саратовского национального исследовательского университета поделено на факультеты, которые включают в себя группы. Номер группы состоит из трехзначного числа:

1. Первое число - номер курса.
2. Второе число - номер специальности.
3. Третье число - номер группы специальности.

Номера групп могут совпадать у некоторых факультетов, но на одном факультете номера групп индивидуальны. Так же расписание делится на отделения и системы подготовки. Отделения могут быть очные, очно-заочные, заочные и вечерние. Системы подготовки делятся на бакалавриат, магистратуру и аспирантуру. В приложении нет необходимости делить расписание на отделения и системы подготовки. Расписание будет выводиться в зависимости от того, какой факультет и группа выбраны.

Swift - это способ писать приложения для телефонов, для десктопных компьютеров, серверов, да и чего-либо еще, что запускает и работает при помощи кода. Swift - безопасный, быстрый и интерактивный язык программирования. Swift вобрал в себя лучшие идеи современных языков с мудростью инженерной культуры Apple. Компилятор оптимизирован для производительности, а язык оптимизирован для разработки, без компромиссов с одной или другой стороны.

Swift исключает большой пласт распространенных программных ошибок при помощи применения современных программных паттернов:

- Переменные всегда инициализированы до того, как будут использованы.
- Индексы массивов всегда проверяются на out-of-bounds ошибки.

- Целые числа проверяются на переполнение.
- Опционалы гарантируют, что значения `nil` будут явно обработаны.
- Автоматическое управление памятью
- Обработка ошибок позволяет осуществлять контролируемое восстановление от непредвиденных ошибок.

Код на Swift скомпилирован и оптимизирован, чтобы получать максимальную отдачу от современного оборудования. Синтаксис стандартной библиотеки спроектирован основываясь на руководстве, что самый очевидный и простой способ написания кода является лучшим вариантом. Комбинация безопасности и скорости делает Swift лучшим кандидатом для написания программ от уровня "Hello, World!" и до целой операционной системы.

Swift совмещает вывод типов и паттерн-матчинг с современным простым синтаксисом, позволяя сложным идеям быть выраженными просто и кратко. И в качестве результата не только становится проще писать код, но и читать его и поддерживать так же становится просто.

XCode — среда для разработки программного обеспечения под операционные системы Apple. С ее помощью можно писать и проверять код, запускать программы, описывать графические интерфейсы и их связь с кодом.

Среда предназначена для операционной системы macOS. Проекты, которые создаются с ее помощью, также можно запускать на iOS, tvOS и watchOS. XCode поддерживает языки программирования Swift, Objective-C, C, C++, а также AppleScript, Python, Ruby и Java. Сторонние разработчики также реализовали поддержку других языков, в том числе старых: Haskell, Pascal, Ada и других.

Применение XCode

- Быстрое и удобное написание кода с помощью специального текстового редактора с подсветкой и проверкой синтаксиса, подсчетом строк и другими функциями.
- Создание и организация крупных проектов, которые могут включать в себя много файлов и папок. Среда помогает структурировать работу.
- Компиляция или интерпретация кода, то есть его преобразование в понятный машине формат.
- Выполнение кода. С помощью IDE можно запускать программу и от-

слеживать результаты ее работы.

- Отладка и тестирование — работа по поиску и устранению ошибок в коде.
- Создание графических интерфейсов и их связь с кодом.
- Автоматизация сборки готового проекта.
- Упрощение архитектурных и конструкторских задач: их помогают решать вспомогательные инструменты, которые наглядно показывают структуру кода или проекта.

Во второй главе проводилась подготовка к реализации приложения и изучение API СГУ:

1. Архитектура MVVM и фреймворк RxSwift Подготовил классы, соответствующие архитектуре, от которых будут наследоваться модули приложения:
 - Router нужен для переходов между контроллерами
 - View Model для бизнес логики
 - MainView для верстки UI элементов экранов
 - View Controller для связи MainView и ViewModel.

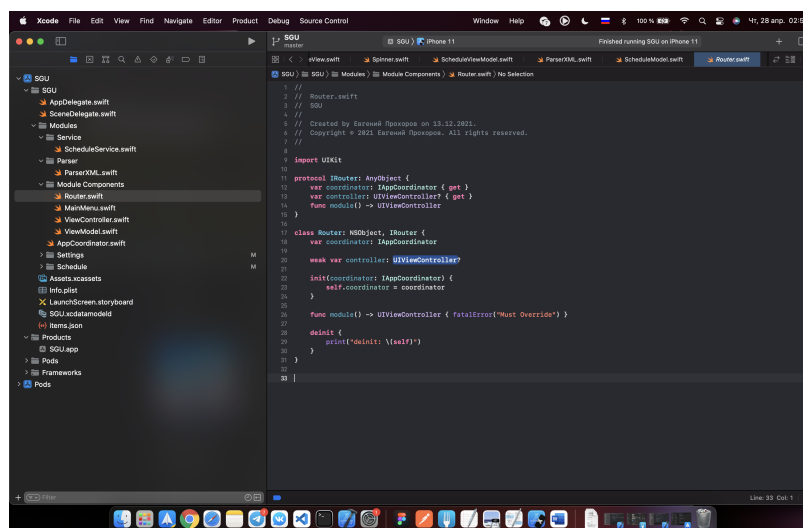


Рисунок 1 – Router

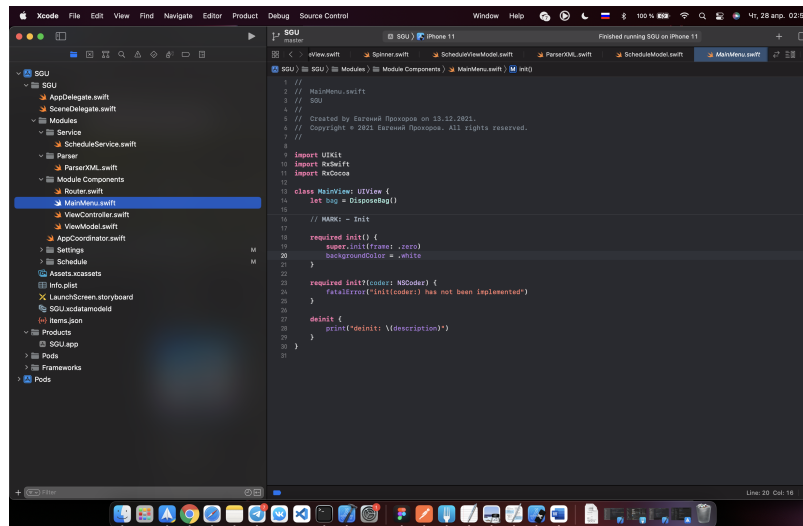


Рисунок 2 – MainView

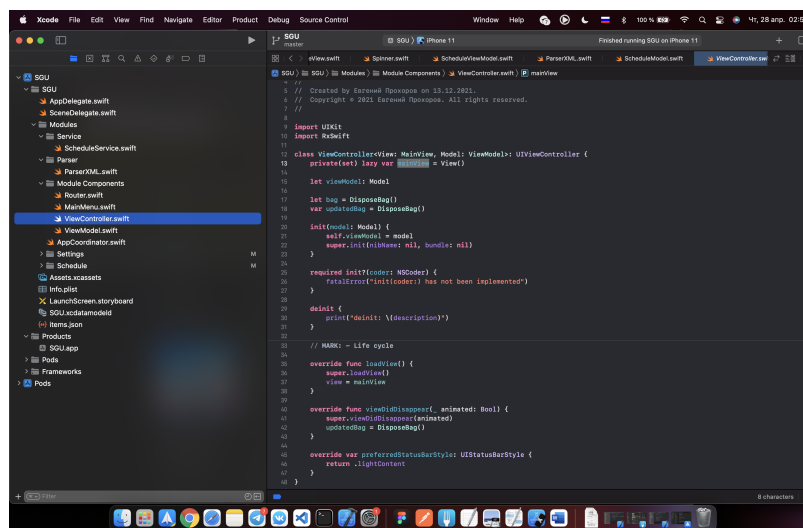


Рисунок 3 – View Controller

2. Способы передачи данных

3. Фреймворки

Используемые фреймворки: SnapKit, RxSwift, RxCocoa, RxDataSources, Alamofire. Подключение происходило через CocoaPods.

4. Модели данных

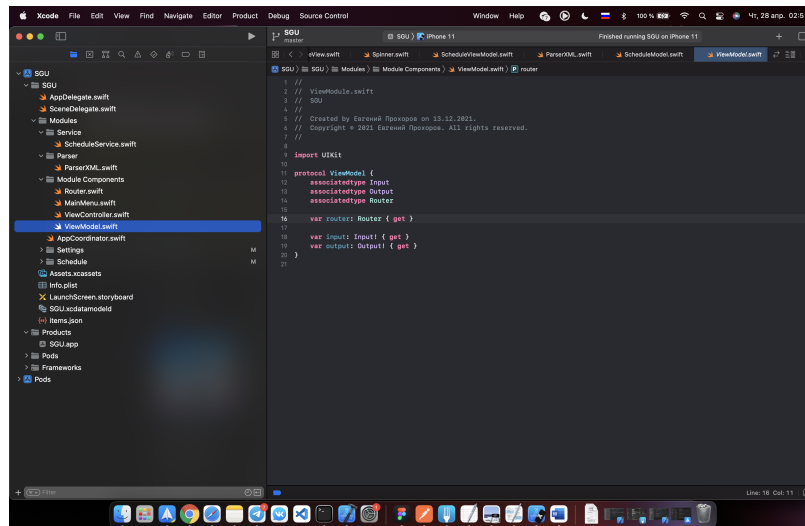


Рисунок 4 – View Model

Третья глава посвящена разработке приложения СГУ и разделена на несколько этапов:

На первом этапе третьей главы была разработка архитектуры MVVM и настройка

Model-View-ViewModel(MVVM) - это модель организации кода.

Расшифровка MVVM:

View Controller: выполняет только вещи, связанные с пользовательским интерфейсом - Показывать/получать информацию.

View Model: получает информацию от VC, обрабатывает всю эту информацию и отправляет ее обратно VC.

Model: представляет собой простые данные. Он просто хранит данные и не имеет ничего общего с какой-либо бизнес-логикой.

MVVM работает в связке с концепцией “Реактивного” пользовательского интерфейса. Реактивные привязки (Reactive Bindings) между View и ViewModel являются основной идеей архитектуры MVVM, где разработчики могут работать с кодом ViewModel, а дизайнеры могут работать с View в Interface Designer.

RxSwift — это библиотека для работы с асинхронным кодом, которая представляет события в виде потоков событий с возможностью подписаться на них, а также позволяет применять к ним подходы функционального программирования, что сильно упрощает работу со сложными последовательно-

```

import Foundation

struct Schedule: Codable {
    let type: TypeLesson
    let subject: String
    let time: LessonTime
    let teacher: String
    let auditorium: String
    let subgroup: String
    let dayOfWeek: DayOfWeek
}

enum DayOfWeek: String, Codable {
    case monday = "Пн"
    case tuesday = "Вт"
    case wednesday = "Ср"
    case thursday = "Чт"
    case friday = "Пт"
    case saturday = "Сб"
}

enum LessonTime: String, Codable {
    case first = "08:20 - 09:50"
    case second = "10:00 - 11:30"
    case third = "12:05 - 13:40"
    case fourth = "13:50 - 15:25"
    case fifth = "15:35 - 17:10"
    case sixth = "17:20 - 18:40"
    case seventh = "18:45 - 20:05"
    case eighth = "20:10 - 21:30"
}

enum TypeLesson: String, Codable {
    case lecture = "лек."
    case practice = "пр."
}

```

Рисунок 5 – Модели данных

стями асинхронных событий.

Предпосылкой появления подобных инструментов является то, что в процессе эволюции iOS SDK представил широкий выбор инструментов для работы с асинхронными событиями: GCD, KVO, Notification Center, шаблон делегирования и механизм замыканий. Проблема заключается в том, что каждый из этих инструментов требует индивидуального подхода и определенной кодовой базы вокруг них, при этом их сочетание может оказаться не слишком элегантным. API, описываемый проектом ReactiveX — это попытка сделать универсальный язык для работы с асинхронными событиями.

На втором этапе третьей главы была проведена верстка экранов с использованием фреймворка SnapKit. Верстка была проведена кодом, а дополнением для удобного размещения элементов на экране стал фреймворк SnapKit.

На третьем этапе третьей главы была настройка API СГУ и парсинг данных. Для доступа к API мне потребовалось запросить у ВУЗа логин и пароль. API СГУ представляет собой данные в формате XML. Поэтому для парсинга данных используется нативный инструмент ParseXML.

На четвертом этапе третьей главы была неотъемлемая часть разработки - тестирование приложения.

141 г.

Расписание

Пн	Вт	Ср	Чт	Пт	Сб
----	----	----	----	----	----

08:20 -

лек.

09:50

Шебалдин Вадим Рудольфович

9-203

Общая и неорганическая химия

лек.

10:00 -

Захарова Тамара Витальевна

1-13

Общая и неорганическая химия

лек.

10:00 -

Захарова Тамара Витальевна

1-13

Общая и неорганическая химия

лек.

12:05 -

Захарова Тамара Витальевна

1-13

Общая и неорганическая химия

лек.

12:05 -

Захарова Тамара Витальевна

1-13

Рисунок 6 – Экран приложения

Заключение

В заключении хочется подвести итог проделанной работы. Исходя из требований общества, а конкретнее, студентов Саратовского национального исследовательского государственного университета имени Н.Г. Чернышевского мною было разработано приложение для системы IOS с расписанием занятий.

Данный продукт поможет студентам оптимизировать свой процесс обучения. Сделает его более удобным и современным.

Для достижения цели, проделаны следующие этапы:

1. Проанализировать предметную область.
2. Выбрать инструменты реализации.
3. Спроектировать приложение.
4. Реализовать приложение.

В процессе работы над данной задачей, приобретены следующие знания в области разработки под ОС IOS:

1. Парсинг данных формата XML
2. Работа с архитектурой MVVM и RxSwift

Считаю, что приложение станет незаменимым в жизни каждого студента и принесёт положительные отзывы в адрес ВУЗа.