**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**Кафедра веб-технологий и компьютерного моделирования**

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПОД ANDROID НА ТЕМУ: “СПРАВОЧНИК АБИТУРИЕНТА”**

Курсовая работа

Станкевича Александра Олеговича

студента 2 курса, специальность 1-31 03 08-02 Математическое и программное обеспечение мобильных устройств

Научный руководитель:  
Старший преподаватель кафедры веб-технологий

Вельченко С. А.

Минск, 2018

Оглавление

[введение 3](#_Toc514337819)

[Глава 1 ВЫБОР ТЕХНОЛОГИЙ И МЕТОДОВ РАЗРАБОТКи 4](#_Toc514337820)

[1.1 Сравнение с аналогами. Место данной работы среди других приложений. 4](#_Toc514337821)

[1.2 Обзор существующих технологий разработки приложений. 4](#_Toc514337822)

[1.3 Структура и проектирование приложения 8](#_Toc514337823)

[Глава 2 РаЗРАБОТКА ПРИЛОЖЕНИЯ 11](#_Toc514337824)

[2.1 Постановка задачи. 11](#_Toc514337825)

[2.2 Описание архитектуры информационной модели 11](#_Toc514337826)

[2.3 Описание функционала приложения. 12](#_Toc514337827)

[2.4 Выбор технологий для разработки 13](#_Toc514337828)

[2.5 Выбор метода управления проектом 15](#_Toc514337829)

[2.6 Этапы разработки приложения 17](#_Toc514337830)

[2.6.1 Разработка дизайна 17](#_Toc514337831)

[2.6.2 Сбор необходимой информации и изучение технологии 17](#_Toc514337832)

[2.6.3 Работа с окнами приложения 18](#_Toc514337833)

[2.6.4 Тестирование приложения 20](#_Toc514337834)

[Приложение было протестировано на 5 смартфонах на операционной системе Android и одном на операционной системе IOS, с разными разрешениями экрана и разными процессорами, все работает отлично, и это заслуга фреймворка Ionic. 20](#_Toc514337835)

[2.7 Перспективы развития проекта 20](#_Toc514337836)

[Заключение 21](#_Toc514337837)

[Список использованной литературы 22](#_Toc514337838)

[Приложения 23](#_Toc514337839)

[5.1 Скриншоты 23](#_Toc514337840)

[5.1.1 Главное окно приложения и работа с ним. 23](#_Toc514337841)

[5.1.2 Окно Calculator приложения и работа с ним. 24](#_Toc514337842)

[5.1.3 Окно Progress и работа с ним 25](#_Toc514337843)

[5.2 Вставки кода 26](#_Toc514337844)

[5.2.1 ionViewWillLoad() 26](#_Toc514337845)

[5.2.2 toggleCh() 26](#_Toc514337846)

введение

В современном мире мы все чаще и чаще используем мобильные телефоны в своей жизни. Каждый из нас по несколько десятков раз в день заходит в приложения на своем смартфоне. Обычный пользователь понимает что оптимизированные приложения для работы с веб сервисами гораздо удобнее, по сравнению с обыкновенным использованием веб-версий данных сервисов. Платформа Android занимает большую долю на рынке смартфонов. Разработка приложения велась именно под неё. Интерес пользователей к различным приложениям растет огромными темпами. Поэтому ежедневно создаются все новые и новые виды, которые стараются отвечать запросам рынка. Современные приложения сопоставимы по своим возможностям с классическими приложениями (desktop applications), но при этом могут быть доступны в любом месте и в любое время на всех современных устройствах. Эти особенности делают разработку приложений очень привлекательными для решения широкого спектра задач. Отличительной особенностью веб-приложения является масштаб: одновременно им может пользоваться большое количество человек. Одним из достоинств выбора разработки приложения для Android является тот факт, что база пользователей огромна. От выбора платформы и фреймворков для разработки зависит как сама по себе реализуемость необходимого функционала, так и возможности дальнейшего развития проекта. Целью моей курсовой работы является создание приложения Справочник абитуриента.

# ВЫБОР ТЕХНОЛОГИЙ И МЕТОДОВ РАЗРАБОТКи

## Сравнение с аналогами. Место данной работы среди других приложений.

Приложение справочник абитуриента является чем-то новым для магазина приложений Google Play Market хотя бы потому, что в магазине нет аналогов. Есть неисчислимое множество приложений “Абитуриент”, но каждое из них относится к отдельному факультету различных вузов, и, несложно догадаться, что на месте абитуриента выбирать себе специальность не очень удобно, как минимум потому, что придется скачивать кучу приложений и заходить в каждое. Такой способ использования не далек от обычного скитания по сайтам факультетов. К тому же, каждое приложение спроектировано по-своему, что добавляет сложности в поиске специальности.

## Обзор существующих технологий разработки приложений.

В настоящее время существует хороший выбор языков программирования для разработки мобильных приложений. Это связанно с тем, что для различных мобильных устройств приходится использовать различные языки программирования. Обычно это связано с тем, что мобильные устройства имеют различные ОС.

Ниже будут рассмотрены такие технологи как Java, Qt (основанный на библиотеках C++), Windows Phone SDK (написание на языке XAML), iPhone SDK (основной язык - Objective-C), Android SDK (основной язык - Java) и Symdian (основной язык - C++).

**1. Java 2 Micro Edition (J2ME).** В первую очередь J2ME это набор спецификаций и технологий, предназначенных для различных типов портативных устройств. Существуют два основных направления: Connected Device Configuration (CDC) и Connected Limited Device Configuration (CLDC). Направление определяет тип конфигурации центральных библиотек Java, а так же параметров виртуальной машины Java (в которой будут исполняться приложения). Логично предположить, что устройства CDC будут более «развитыми», в качестве примера можно привести коммуникаторы. К устройствам CLDC относятся обычные мобильные телефоны, аппаратно обладающие более скромными возможностями (ресурсами). Специальные режимы позволяют определять функциональность конфигураций для различных типов устройств. Режим Mobile Information Device Profile (MIDP) предназначен для CLDC портативных устройств с возможностью коммуницировать. Режим MIDP определяет функциональность - работу пользовательского интерфейса, сохранение настроек, работу в сети и модель приложения. CLDC и MIDP закладывают основу реализации J2ME [1].

Java-код интерпретируется непосредственно самим устройством при помощи так называемой Java Virtual Machine. Этот механизм делает возможным свободное распространение Java-приложений, так как они работают на всех устройствах с аналогичной Java-платформой [2].

Программирование Java-приложений и на сегодняшний день занимает большую часть, так как большинство мобильных устройств (в основном мобильные телефоны) в мире имеют уже предустановленную Java-машину.

**2. Qt.** Среда разработки Qt была приобретена Nokia в 2008 г. у норвежской Trolltech за 150 миллионов долларов. Qt в основном используется в качестве кросс-платформенной среды, которая позволяет использовать написанные с ее помощью приложения на различных устройствах и операционных системах, в том числе Windows, Mac OS X, Linux, Symbian, Android и других [3]. Начиная с версии Qt 4.0 появилась возможность программировать для мобильных устройств. С растущей пользовательской базой Qt, растёт потребность во встроенных, мобильных приложениях и UI-разработчиках.

Qt является одной из самых удачных библиотек для С++. Отладка приложений, разработанных для мобильных устройств, происходит с помощью эмулятора, который содержится в среде разработки. Таким образом, мы можем писать сложные приложения для мобильных устройств с использованием библиотек C++ и поддержкой кроссплатформенности.

В настоящее время последняя версия - Qt 5 бета. Финальный релиз планируется на 2012 год. Для работы Qt на мобильных устройствах необходима установка соответствующего Фреймворка.

**3.** **Windows Phone SDK.**На момент написания этой статьи, последняя версия инструментария доступна в версии Windows Phone SDK 7.1 Release Candidate в лицензии «Go Live» с возможностью разрабатывать свои приложения и публиковать их в Windows Phone Marketplace. Windows Phone SDK 7.1 Release Candidate содержит следующие компоненты [4]:

* Windows Phone SDK 7.1;
* Windows Phone Emulator;
* Windows Phone SDK 7.1 Assemblies;
* Silverlight 4 SDK and DRT;
* Windows Phone SDK 7.1 Extensions for XNA Game Studio 4.0;
* Expression Blend SDK for Windows Phone 7;
* Expression Blend SDK for Windows Phone OS 7.1;
* WCF Data Services Client for Windows Phone;
* Microsoft Advertising SDK for Windows Phone.

Код разрабатываемого приложения описывается на языке XAML. На самом деле - это просто XML файлы с языком разметки XAML.

Платформа Windows Phone не просто очередная платформа для мобильных устройств. Она содержит в себе не только технологическую составляющую, но и полностью проработанную концепцию дизайна интерфейса и взаимодействия с пользователем под названием Metro-дизайн или стиль Metro [4].

Вся разработка под Windows Phone ведется в среде Visual Studio. Среда является очень удобной для разработки и отладки приложений. Для мобильных приложений под Windows Phone отладка происходит с помощью эмулятора Windows Phone с помощью среды разработки Windows Phone.

**4. iPhone SDK.** Разработка под iPhone под операционную систему iOS возможна только под Mac OS X. Но в Интернете можно найти статьи, как можно программировать и на Macintosh и даже на VM. Стоит заметить, что Apple предоставляет инструменты бесплатно, платить придется за подписку разработчика [5].

Для написания программ под iPhone предлагается использовать Objective?C. При этом есть возможность писать так же и на C и на C++ (для этого необходимо изменять расширения файлов с .m на .mm). Правда при этом полностью уйти от Obj?C не удастся, почти весь API рассчитан именно на Obj?C, исключения составляют например OpenGL (хотя для его инициализации придется использовать несколько строк кода на Obj?C), так же полностью доступны стандартные библиотеки C/C++ (так, например, с файловой системой можно работать как средствами SDK на Obj?C, так и используя стандартную библиотеку С для ввода/вывода (fopen(), fgetc(), etc)) [5].

Отладка приложения происходит с помощью среды XCode и эмулятора iPhone установленного в ней.

**5. Android SDK.** Для разработки под Android можно использовать среду Eclipse с установленным плагином ADT. Разработка ведется на языке программирования Java. Есть возможность отладки с использованием эмулятора встроенного в ADT или непосредственно на мобильном устройстве с ОС Android.

Существует различные версии SDK, которые используются для написания кода для различных версий Android. В настоящее время большое распространения получили версии 2.2 и 2.3. Поддерживается почти полная обратная совместимость версий.

Кроме разработки на языке Java поддерживается возможность более низкоуровневая разработка с использованием Android NDK (Native Development Kit) на языке C/C++.

**6. Symbian и C++.** Для написания приложений под Symbian можно использовать язык программирования C++. В основном данный подход используется для Symbian OS v6.1, 7.0, 7.0s и 8.0 [6].

Разработка для Symbian OS (если говорить о C++) обычно ведется на ПК. Среда разработки - привычная многим программистам Visual Studio, это также могут быть IDE Metrowerks CodeWarrior Development Studio, Borland C++BuilderX Mobile Edition, Carbide.C++ (относительно новая IDE, созданная компанией Nokia на базе Eclipse), снабженная дополнительными инструментальными пакетами (SDK). Разработчику доступны практически все привычные возможности в отношении как создания ПО, так и отладки (трассировка, просмотр переменных, стека вызовов, структур классов и др.).

Отлаживаемая программа запускается в эмуляторе Symbian OS. Отметим, что эту подсистему правильнее было бы назвать симулятором, поскольку имитируются не аппаратные средства, а лишь программное окружение (соответствующие API операционной системы, реализованные поверх API Win32). При этом программные модули, которые загружаются в эмулятор, представляют собой исполнимые файлы для архитектуры x86 (не ARM, на базе которой построены смартфоны), соответствующее ПО для целевой платформы формируется после итоговой компиляции. Это предполагает определенную специфику (скажем, ранее была довольно распространена ситуация, когда программа, нормально функционировавшая в среде эмулятора, отказывалась работать на реальном устройстве), но сегодня эмулятор обеспечивает достаточно высокую степень сходства и проблемы возникают лишь при создании программ, нестандартно использующих API [7].

**Выводы по результатам обзора.**С появлением новых технологий, ранее использовавшиеся уходят в историю. Сейчас разработчиков, которые используют такие технологии как Symbian с использованием C++ и/или J2ME, становятся все меньше и лидирующую позицию занимают технологии, использующие различные SDK (Windows Phone SDK, iPhone SDK, Android SDK). Но недостаток всех существующих SDK в том, что разрабатываются нативные приложения, т.е. приложения, функционирующие под управлением только одной ОС.

## Структура и проектирование приложения

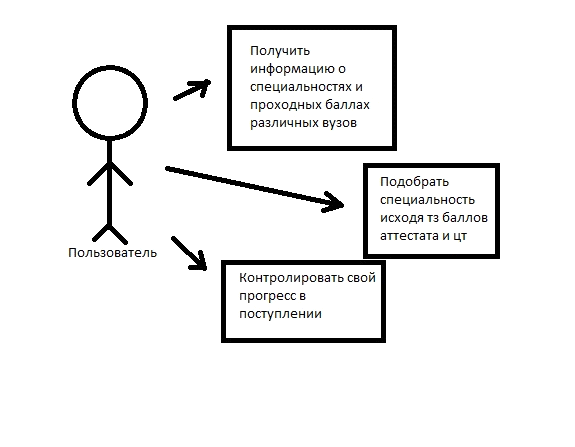


Рисунок 1.1 Возможности пользователя

Как видно из данной схемы, у пользователя есть возможность получить информацию о специальностях и проходных баллах на различные факультеты всех вузов, подобрать специальность в калькуляторе исходя из своих баллов централизованного тестирования и посмотреть все, доступные ему специальности из всех факультетов, а также возможность контролировать свой прогресс в поступлении не выходя из приложения. Данная схема в общих чертах отображает пользовательский интерфейс.

Далее определим главные компоненты приложения:

* ***HOME*** является главным окном, видным пользователю при запуске приложения. На странице имеется список факультетов разных университетов, нажав на определенный факультет отображается список всех специальностей этого факультет, список централизованого тестирования или репетиционного тестирования с минимальными баллами, необходимыми для поступления на факультет, а так же баллы для каждый специальности отдельно, так же пользователь может подробнее почитать о специальности нажав на соответствующую кнопку "подробнее". Кроме того, пользователь может воспользоваться поиском по ключевым словам (пример: если написать "прог...", осуществляется поиск специальностей связанных с программированием)

***CALCULATOR*** является окном, на котором пользователь может подобрать специальность исходя из баллов, набранных на централизованном тестировании или репетиционного тестирования и балла аттестата. Выбрав нужные предметы, он вводит свои баллы и получает список специальностей всех факультетов, которые ему доступны с такими баллами.

* ***PROGRESS*** является заключительным окном приложения, на нем абитуриент имеет возможность помечать свой прогресс в поступлении

# РаЗРАБОТКА ПРИЛОЖЕНИЯ

## Постановка задачи.

Главной целью работы является создание приложения-помощника абитуриенту, которое сможет объединить в себе весь тот набор приложений, что есть в Google Play Market, то есть, главная задача – создать справочник который хранит в себе информацию сразу всех факультетов всех университетов Беларуси.

## Описание архитектуры информационной модели

На следующем этапе составим диаграмму классов логики, которая будет отображать методы, необходимые для функционирования приложения:

****

Рисунок 2.2.1 Классы логики

Таким образом, основными сервисами являются HomePage, Progress и Calculator.

Сервис HomePage содержит методы getData(), который получает всю доступную информацию о университетах, факультетах и специальностях, getItems() для получения специальностей(items) от определенных факультетов и метод openModal() для работы с поиском.

Сервис Progress содержит метод showItems() для вывода задач, которые нужны для поступления.

Сервис Calculator. Он содержит четыре метода:

1. getData() – такой же, как у homePage метод, который получает всю доступную информацию о университетах, факультетах и специальностях.

2. getItems() – опять же, такой же, как у HomePage метод, для получения специальностей(items) от определенных факультетов.

3.calc() - основной для данной страницы метод, необходимый для подбора специальностей изходя из введенных данных.

4.showDescription() - метод для отображения подробной информации о специальности.

## Описание функционала приложения.

Теперь, когда продуманы все необходимые компоненты, можно описать работу приложения. Для удобства используем блок-схему:



Рисунок 2.3.2 Работа приложения

На данной схеме видно, что пользователь может зайти в приложение не только с целью найти специальность, но и с целью записать свой собственный прогресс в вышеописанном окне PROGRESS, в котором он имеет возможность поставить галочки (checkboxes) на необходимых для поступления вещах при их выполнении, тем самым, контролирую свой успех на пути к поступлению.

Также мы видим, что если пользователь не смог сам выбрать специальность либо не хочет этого делать самостоятельно, он имеет возможность подобрать специальность основываясь на своих баллах централизованного тестирования или репетиционного тестирования и баллах аттестата в окне CALCULATOR.

## Выбор технологий для разработки

Следующим этапом после проектирования приложений является этап выбора технологий. Чтобы наш выбор был наиболее точным, необходимо четко сформулировать критерии, по которым он будет осуществляться. После проведенного анализа я смог сформулировать основные критерии, которыми пользовался в ходе своей работы:

* Удобство в использовании.
* Оптимизированность под различные виды устройств.
* Простота в изучении.
* Технология, знания которой могут понадобиться в будущей карьере.

После тщательного их анализа мой выбор пал на ionic cordova - мощная платформа, созданная для создания красивых и функциональных приложений. Интегрирован мощный набором инструментов и услуг Ionic Pro, разработанный для всего жизненного цикла приложения.

Ionic framework — один из самых широко обсуждаемых фреймворков. Как гласит официальный сайт, Ionic — это SDK для создания гибридных мобильных приложений, набор CSS и JS компонент, созданный на основе AngularJS, SASS, Apache Cordova.

По своей сути, Ionic Framework — это дополнение над очень популярным фреймворком Apache Cordova, но со своим мощным CLI (Command Line Interface) и объемной документацией. Следуя принципам Apache Cordova, приложения на Ionic Framework — это гибридные HTML приложения. Такие приложения на телефоне выполняются в специальной оболочке (UIWebView для iOS и WebView для Android), которая позволяет показывать HTML и выполнять JavaScript. Соответственно при работе в приложении пользователь работает как бы в веб-браузере.

Apache Cordova является своеобразной прослойкой между пользовательским интерфейсом и ресурсами устройства. Какие-либо виджеты пользовательского интерфейса или MV\* (Model-View) фреймворков не входят в его состав. Если необходимо использовать UI-виджеты и/или MV\* фреймворк, то нужно выбрать и включить их в приложение самостоятельно, как ресурсы третьей стороны. Ionic 2 — это один из фреймворков, который предоставляет UI виджеты. MVС модель для него предоставляет Angular 2. Нативные функции телефона (например, камера, хранилище ключей, GPS координаты) недоступны из веб-браузера. Поэтому для работы с ними используются плагины Apache Cordova. Кроме официальных плагинов есть ряд сторонних открытых плагинов. Ionic cli - по сути, это обёртка над Cordova CLI, предоставляющая дополнительные возможности:

* создание базового приложения с выбором шаблонов (например приложение с боковым меню, приложение с табами, картами Google Maps, пустое приложение)
* сборка и запуск в эмуляторе, на реальном устройстве, в браузере
* live reload в браузере и на устройстве
* генерацию иконок, сплешскринов и др.

## Выбор метода управления проектом

Далее нужно было выбрать метод управления проектом, и здесь выбор пал на agile, хоть компоненты и не так сильно зависимы друг от друга, я привык работать по данному методу, вот его описание:

Не все проекты могут быть структурированы таким образом, чтобы быть реализованными по классическому проектному подходу. К примеру: приготовление одного блюда идеально ложится на «водопадный» подход, а вот вовремя приготовить и подать ужин из четырёх блюд будет практически невозможно, если придётся каждый раз ждать окончания приготовления одного блюда, чтобы приступить к приготовлению другого.

И тут в игру вступает Agile – семейство гибких итеративно-инкрементальных методов к управлению проектами и продуктами. Согласно данному подходу, проект разбивается не на последовательные фазы, а на маленькие подпроекты, которые затем «собираются» в готовый продукт. Схема работы приведена на

Таким образом, инициация и верхнеуровневое планирование проводятся для всего проекта, а последующие этапы: разработка, тестирование и прочие проводятся для каждого мини-проекта отдельно. Это позволяет передавать результаты этих мини-проектов, так называемые, инкременты, быстрее, а приступая к новому подпроекту (итарации) в него можно внести изменения без больших затрат и влияния на остальные части проекта.

Несмотря на то, что Agile вошёл в моду относительно недавно, идея итеративной разработки не нова. Своё нынешнее название семейство гибких методологий получило в 2001 с публикации Манифеста Agile (Agile Manifesto), закрепившем основные ценности и принципы гибкой разработки программного обеспечения, в основе которых – командная работа и адаптация, даже «любовь» к изменениям.

Сам по себе Agile – не метод управления проектами. Это скорее набор идей и принципов того, как нужно реализовывать проекты. Уже на основе этих принципов и лучших практик были разработаны отдельные гибкие методы или, как их иногда называют, фреймворки (frameworks): Scrum, Kanban, Crystal, и многие другие. Эти методы могут достаточно сильно отличаться друг от друга, но они следуют одним и тем же принципам.

**Сильные стороны Agile**

Самое главное достоинство Agile – его гибкость и адаптивность. Он может подстроиться под практически любые условия и процессы организации. Именно это обуславливает его нынешнюю популярность и то, сколько систем для различных областей было создано на его основе.

Один из принципов Agile: «Реакция на изменения важнее следования плану». Именно быстрая и относительно безболезненная реакция на изменения является причиной тому, что многие крупные компании стремятся сделать свои процессы более гибкими. Кроме того, Agile отлично подходит для проектов с «открытым концом» — например, запуску сервиса или блога.

Вотчина Agile – разработка новых, инновационных продуктов. В проектах по разработке таких продуктов высока доля неопределённости, а информация о продукте раскрывается по ходу проекта. В таких условиях реализовывать проект по «водопаду» становится невозможно– нет информации для планирования.

**Слабые стороны Agile**

В отличие от PRINCE2 и PMBOK Agile – не является ни методологией, ни стандартом. Agile — это набор принципов и ценностей. Слабая сторона состоит в том, что каждой команде придётся самостоятельно составлять свою систему управления, руководствуясь принципами Agile. Это непростой и длительный процесс, который потребует изменений всей организации, начиная процедурами и заканчивая базовыми ценностями. Это тернистый путь и не всем организациям он под силу.

Этот путь потребует от лидера изменений не только знаний и упорства, но и серьёзных административных ресурсов, а также затрат. К счастью, существуют готовые наборы практик, которые облегчают Agile-трансформацию организации. К таким наборам относятся фреймворк Scrum, метод Kanban и многие другие – Crystal, LeSS, SAFe, Nexus.

## Этапы разработки приложения

### Разработка дизайна

Для начала нужно определить стиль исполнения всего проекта. Хороший дизайн, как хороший офис или машина, всегда влияет на первое впечатление о приложении. Затем особое внимание уделяется главной странице, на которых пользователь должен совершить какое-то целевое действие. Но, я решил, что данное приложение является лишь инструментом-помощником в поиске специальности и поэтому не стал вносить сильных изменений с строгий выдержанный дизайн стандартных табов фреймворка ionic.

### Сбор необходимой информации и изучение технологии

Для сбора необходимой информации в основном я использовал видео уроки из сервиса YouTube, в изучении технологии ionic мне помог официальный сайт ionicframework.com, а так же тот же самый YouTube. Особое внимание было уделено компонентам ionic, таких как tabs, modal-pages, ion-item, ion-card и др. Так же было уделено внимание освоению typescript и angular.

### Работа с окнами приложения

#### Работа с HomePage

Скриншот окна см. приложение 5.1.1

В разработке главного окна приложения основную роль сыграл метод:

ionViewWillLoad(){см. приложение 5.2.1}

Данный метод позволяет получить информацию из json-файла для последующей работы с ним.

Далее нужно было написать элемент-поиск, который является одним из главных на странице. В написании кода роль сыграло знание typescript и angular.

#### Работа с Calculator

Скриншот окна см. приложение 5.1.2

В разработке второго окна приложения основную роль сыграли метод:

toggleCh(ch:**string**){ см. приложение 5.2.2}

#### Работа с Progress

Скриншот окна см. приложение 5.1.3

Это самое простое окно. Сюда пользователь заходит с целью записать свой собственный прогресс, окно обладает возможностью запоминать выбранные элементы, в котором он имеет возможность поставить галочки (checkboxes) на необходимых для поступления вещах при их выполнении, тем самым, контролирую свой успех на пути к поступлению.

#### Рефакторинг кода

Была произведена работа по рефакторингу кода. Рефакторинг играет особую роль в качестве дополнения к проектированию. Если заранее подумать об архитектуре программы, то можно избежать последующей дорогостоящей переработки. Многие считают, что проектирование важнее всего, а программирование представляет собой механический процесс. Аналогией проекта служит технический чертеж, а аналогией кода — изготовление узла. Но программа весьма отличается от физического механизма. Она значительно более податлива и целиком связана с обдумыванием. Как говорит Элистер Кокберн:  
«При наличии готового дизайна я думаю очень быстро, но в моем мышлении полно пробелов».

### Тестирование приложения

## Приложение было протестировано на 5 смартфонах на операционной системе Android и одном на операционной системе IOS, с разными разрешениями экрана и разными процессорами, все работает отлично, и это заслуга фреймворка Ionic.

## Перспективы развития проекта

В дальнейшем планируется доработка приложения и добавление нового функционала. Планируется выгрузка приложения в Google PlayMarket после дополнения приложения информацией о всех университетах страны. Я считаю приложение перспективным, потому что аналогов ему для нашей страны на данный момент нет, существуют только приложения отдельных вузов работающие некорректно. Приложение, имеющее в себе информацию о всех вузах одновременно точно будет пользоваться популярностью. Я считаю, что при должной поддержке приложение точно будет жить.

# Заключение

Как было написано в пункте 1 гл.2, главной целью работы было создание приложения-помощника абитуриенту, которое сможет объединить в себе весь тот набор приложений, что есть в Google Play Market, то есть, главной задачей являлось создание справочника который хранит в себе информацию сразу всех факультетов всех университетов Беларуси. Приложение справочник абитуриента является чем-то новым для магазина приложений Google Play Market хотя бы потому, что в магазине нет аналогов. Есть неисчислимое множество приложений “Абитуриент”, но каждое из них относится к отдельному факультету различных вузов. Данное приложение было представляет из себя помощника, который включает в себя весь набор возможностей, которые требует от него абитуриент, и, на мой взгляд, приложение будет пользоваться популярностью среди поступающих нашей страны.

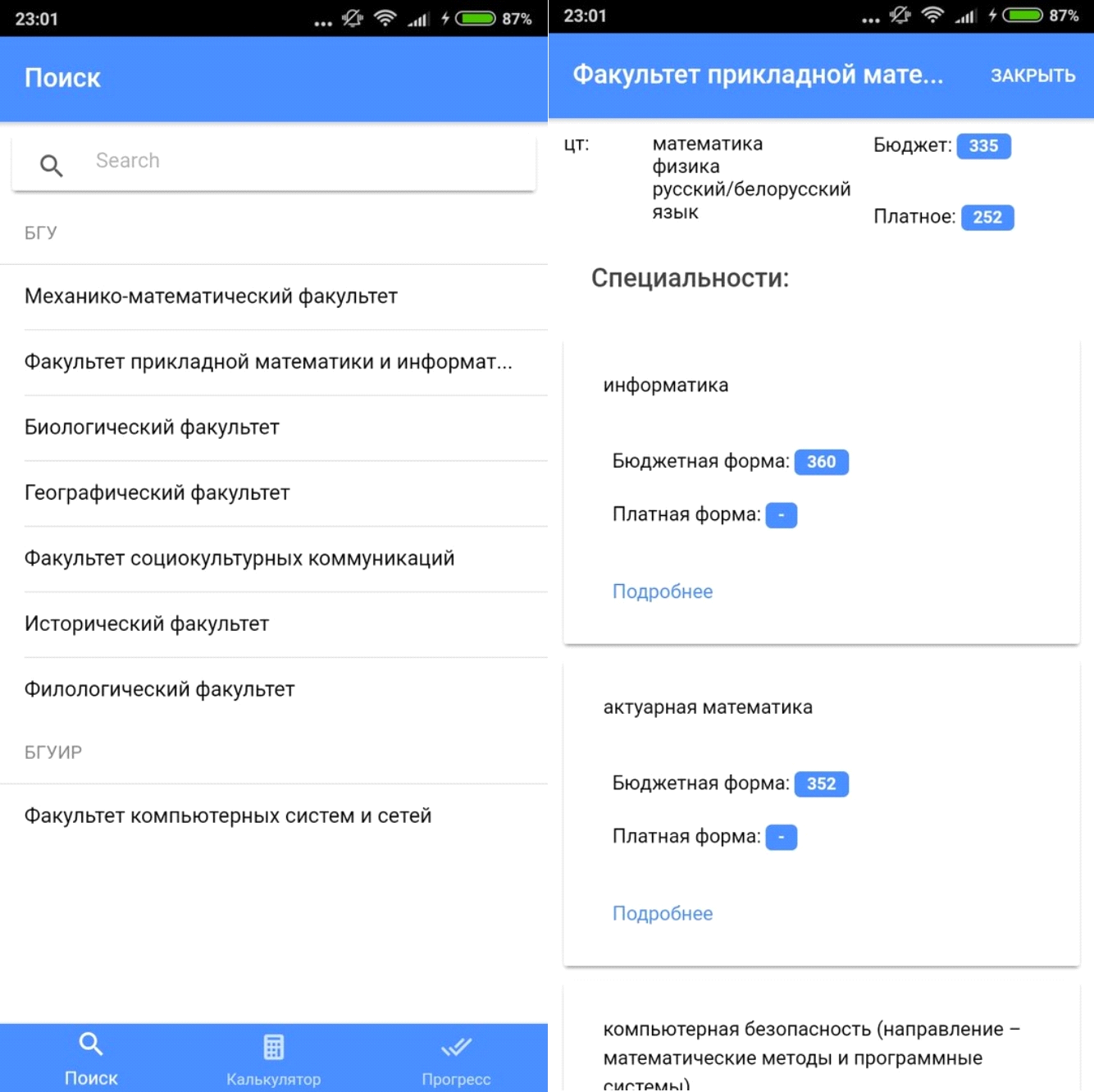
# Список использованной литературы

1. habrahabr.ru
2. stackoverflow.com
3. ionicframework.com
4. [www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/](http://www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/)
5. Курс лекций о структуре Android (преподаватель Дерюшев A.A.)

Приложения

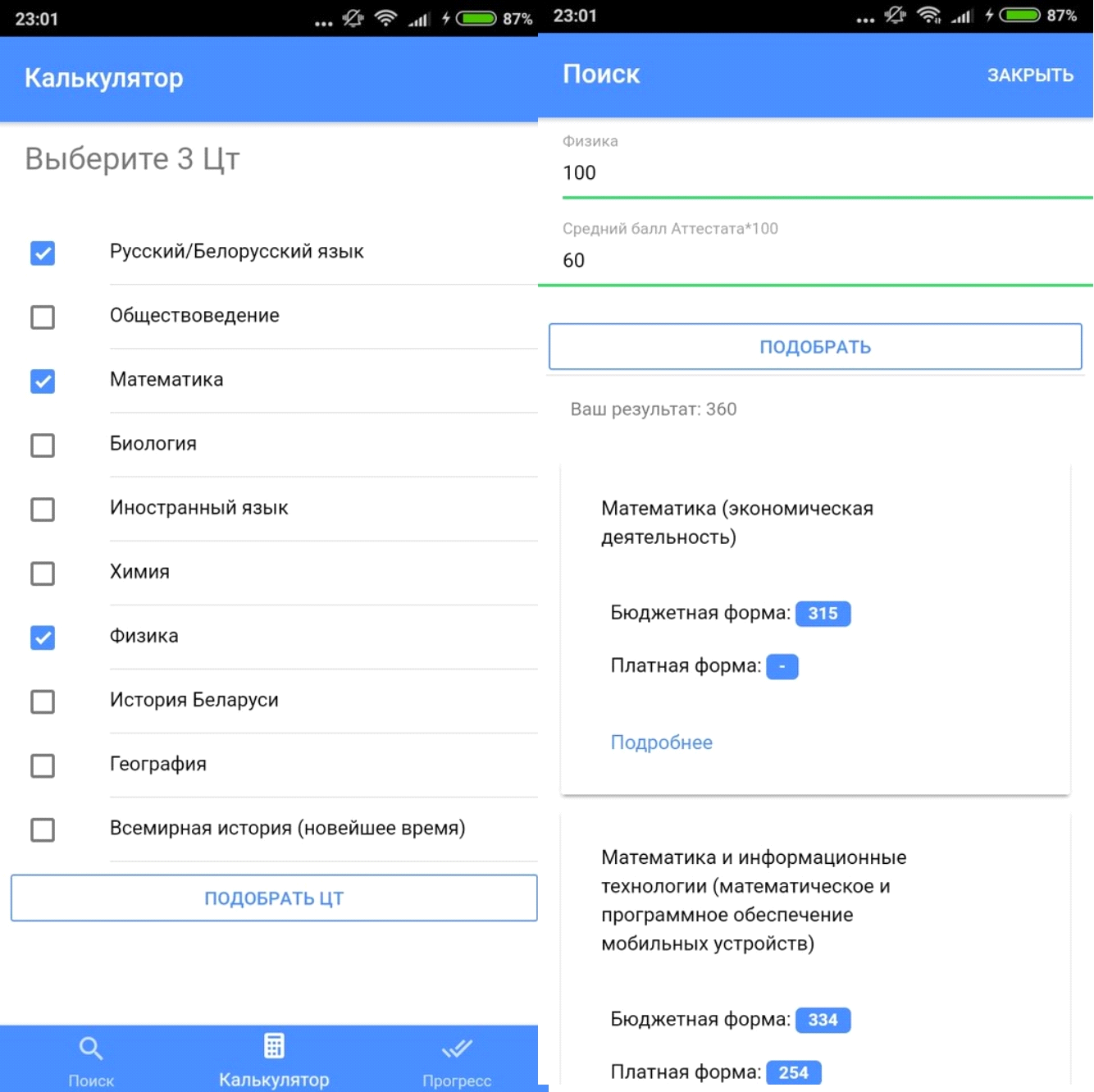
## Скриншоты

### 5.1.1 Главное окно приложения и работа с ним.

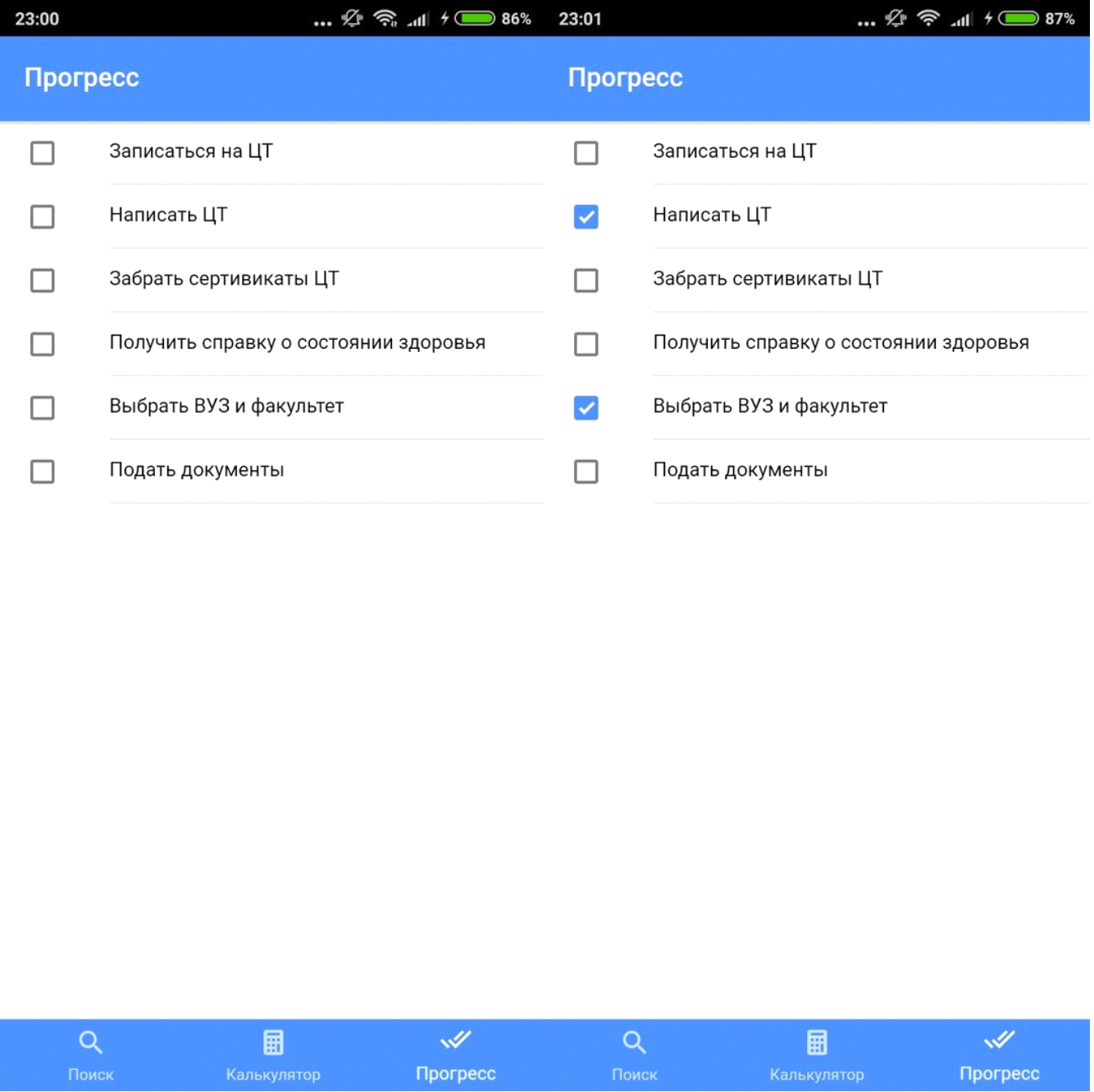
****

.

### Окно Calculator приложения и работа с ним.

****

### 5.1.3 Окно Progress и работа с ним



## Вставки кода

### 5.2.1 ionViewWillLoad()

ionViewWillLoad(){

this.items.Ex = this.navParams.get(“data”);

this.newHttp.get(“assets/data/univs.json”).subscribe(data=>{

this.items = data;

});

}

### 5.2.2 toggleCh()

toggleCh(ch:**string**){  
 **if**(**this**.**active**.indexOf(ch) >= 0){  
 **this**.**active**.splice(**this**.**active**.indexOf(ch), 1);  
 }  
 **else**{  
 **this**.**active**.push(ch);  
 }  
}