

Правительство Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Санкт-Петербургский государственный университет»

Кафедра информатики

Котенко Егор Даниилович

Разработка мобильного облачного сервиса "Система Домашних Заданий"  
для Microsoft Azure и взаимодействующего с ним клиентского мобильного  
приложения (ОС: Windows Phone 8)

Дипломная работа

Допущена к защите.

Зав. Кафедрой:

д.ф.-м.н., профессор Косовский Н.К.

Научный руководитель:

д.т.н., профессор Сафонов В.О.

Рецензент:

аспирант кафедры информатики

Крень М.В.

Санкт-Петербург

2015

SAINT-PETERSBURG STATE UNIVERSITY

Chair of Computer Science

Egor Kotenko

Development of Homework System mobile cloud service for Microsoft Azure and  
interacting Windows Phone 8 client mobile application

Graduation Thesis

Admitted for defence.

Head of the chair:

Dr.sc.math., professor Nikolai Kossovski

Scientific supervisor:

Dr.sc.tech., professor Vladimir Safonov

Reviewer:

Ph.D student of Computer Science Chair

Mariia Kren

Saint-Petersburg

2015

## Оглавление

1. Введение .....	4
2. Постановка задачи .....	6
2.1. Предметная область и терминология .....	6
2.2. Цели работы .....	8
3. Обзор существующих решений.....	10
3.1. Android и Google Play.....	10
3.2. iOS и App Store.....	12
3.3. Windows Phone и Windows Phone Store .....	14
3.4. Вывод.....	15
4. Облачные вычисления .....	16
4.1. Модели обслуживания в облачных вычислениях.....	17
4.2. Виды (модели) облаков .....	18
4.3. Уровни компонент облачных вычислений .....	19
4.4. Архитектура облачных вычислений .....	19
4.5. Обзор платформ облачных вычислений .....	20
5. Microsoft Azure .....	23
5.1. Обзор платформы .....	23
5.2. Компоненты Microsoft Azure.....	26
5.2.1. Компонента Вычисления (Compute).....	26
5.2.2. Компонента Сервисы данных(Data Services).....	27
5.2.3. Компонента Сетевые средства (Networking) .....	28
5.2.3. Компонента Управления приложениями (App Services) .....	29
5.2.4. Компонента Коммерция (Commerce) .....	31
5.3. Краткие итоги .....	32
6. Windows Phone .....	33
7. Языки программирования .....	36
7.1. C#.....	36
7.2. Javascript .....	37
8. Реализация .....	39
8.1. Облачная часть.....	40
8.2. Клиентская часть.....	41
9. Результаты и дальнейшая работа .....	45
9.1. Результаты.....	45
9.2. Дальнейшая работа .....	45
Список литературы .....	47

## 1. Введение

На сегодняшний день невозможно описать образ современного человека, не упоминая смартфон или электронный планшет. В последние годы данные мобильные устройства стали уже не просто средствами связи – они способны выполнять основные функции персонального компьютера, электронной книги, фотоаппарата, видеокамеры, навигатора и множества других гаджетов. В связи с таким широким спектром возможностей, компактностью и удобством использования смартфоны завоевывают всё большую популярность по всему миру и, в частности, в России. Согласно оценке экспертов [1], в 2014 году объем российского рынка смартфонов достиг 27,4 млн в штуках и \$6,3 млрд в деньгах, а годом ранее эти показатели составляли 18,8 млн единиц и \$5,4 млрд соответственно. На данный момент смартфоны распространены среди представителей практически всех социальных классов независимо от возраста и рода деятельности.

Столь высокая функциональность смартфона обусловлена не только стандартными возможностями устройства, но и огромным количеством мобильных приложений, значительно расширяющих набор функций, предоставляемых аппаратом. Рынок мобильных приложений развивается не менее стремительно рынка смартфонов: по данным статистического анализа [2], прогнозируется, что рынок мобильных приложений в России достигнет \$1.3 млрд к 2016 году, увеличившись в восемь раз по сравнению с 2012 годом. Каждое мобильное приложение, реализующее востребованную в какой-либо сфере функциональность, находит свою аудиторию и захватывает свой сегмент рынка.

Факультативное образование и частное репетиторство в России являются достаточно развитыми сферами деятельности. По данным

исследования Ассоциации репетиторства [3], более 30 тысяч официально зарегистрированных частных преподавателей предоставляют свои услуги на территории Москвы и Санкт-Петербурга, количество репетиторов на территории РФ, официально не оформляющих своё частное предпринимательство, измеряется сотнями тысяч.

Таким образом, сфера науки и образования не является исключением – приложения, связанные с процессами обучения, также пользуются спросом, пусть и уступая в этом показателе развлекательному контенту. Но в том числе благодаря и чрезмерному количеству увеселительных и бессодержательных предложений, растёт спрос на услуги и приложения в каком-то роде противоположного, учебного характера, например, в сфере преподавания и репетиторства.

Тема данной работы была выбрана с учётом положительной динамики популярности мобильных приложений, активного развития облачных технологий и растущей актуальности приложений образовательного характера.

## 2. Постановка задачи

### 2.1. Предметная область и терминология

Предметной областью данной работы является разработка мобильного облачного сервиса и клиентского мобильного приложения. В работе использованы следующие определения и термины:

- Облако — метафора для изображения сервисов, предоставляемых через Интернет или другую сеть;
- Сервис (служба) — программная система со стандартизированными интерфейсами. Могут взаимодействовать друг с другом или со сторонними приложениями посредством сообщений, основанных на определенных протоколах. В основном взаимодействие происходит через сеть Интернет;
- Виртуальная машина (ВМ, от англ. virtual machine) — программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы;
- .NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.
- IEEE (Institute of Electrical and Electronics Engineers)- международная некоммерческая ассоциация специалистов в области техники;
- WCF (Windows Communication Foundation) — программный фреймворк, используемый для обмена данными между приложениями, входящий в состав .NET Framework;
- WPF (Windows Presentation Foundation) система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая

- (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0);
- ASP.NET(Active Server Pages) — технология создания веб-приложений и веб-сервисов от компании Майкрософт. Она является составной частью платформы Microsoft .NET;
  - Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода.
  - Push-уведомления - один из способов распространения информации (контента) в Интернете, когда данные поступают от сервера к клиенту на основе ряда параметров, установленных клиентом, где запрос на передачу информации происходит по инициативе центрального сервера.
  - SQL (structured query language) — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД);
  - LINQ (Language Integrated Query) — проект Microsoft по добавлению синтаксиса языка запросов, напоминающего SQL, в языки программирования платформы .NET Framework;

## 2.2. Цели работы

Целью данной работы является разработка и публикация в облаке мобильного сервиса "Система Домашних Заданий" для Microsoft Azure и клиентского мобильного приложения для платформы Windows Phone 8 для взаимодействия с разработанным мобильным сервисом. Приложение должно предоставлять возможность подписаться на один или несколько курсов по различным предметам, получать домашние задания по выбранным курсам, вводить и проверять правильность ответов на задачи из домашних заданий.

Основные критерии, которым должно удовлетворять разработанное клиентское приложение:

- Отображение набора доступных обучающих курсов;
- Возможность подписаться на необходимые курсы, или же отписаться от них;
- Отображение списка актуальных домашних заданий по выбранным курсам;
- Отображение списка условий задач из актуальных домашних заданий;
- Оповещение о наличии новых актуальных домашних заданий посредством push-уведомлений;
- Возможность проверки правильности ответов, полученных в результате решения задач.

Облачный сервис должен, соответственно, обеспечивать:

- Хранение структурированных данных о всех курсах, домашних заданиях и задачах;



- Поддержку работы планировщика, обеспечивающего рассылку информации о новых актуальных домашних заданиях;
- Корректность обработки и исполнения запросов к данным;
- Доступность, использование сервиса должно быть возможно при наличии соединения устройства с сетью;
- Кроссплатформенность, для возможного подключения с устройств на иных платформах.

### 3. Обзор существующих решений

По данным International Data Corporation [1], самыми распространёнными в мире и, в частности, в России мобильными операционными системами являются, соответственно, Android, iOS и Windows Phone. Рассмотрим официальные магазины мобильных приложений для данных операционных систем и проанализируем данные платформы на предмет наличия и качества образовательных мобильных приложений и, конкретнее, приложения-помощника для репетитора и его учеников.

#### 3.1. Android и Google Play

Android [4]—операционная система для смартфонов, планшетных компьютеров, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, телевизоров и других устройств. Основана на ядре Linux и собственной реализации машины Java от Google. В 86 % смартфонов, проданных во втором квартале 2014 года, была установлена операционная система Android, при этом за весь 2014 год было продано более 1 миллиарда Android-устройств.

Google Play – крупнейший магазин приложений, игр, книг, музыки и фильмов для Android-устройств.

В Google Play присутствует множество приложений, связанных с репетиторством (Рисунок 1).

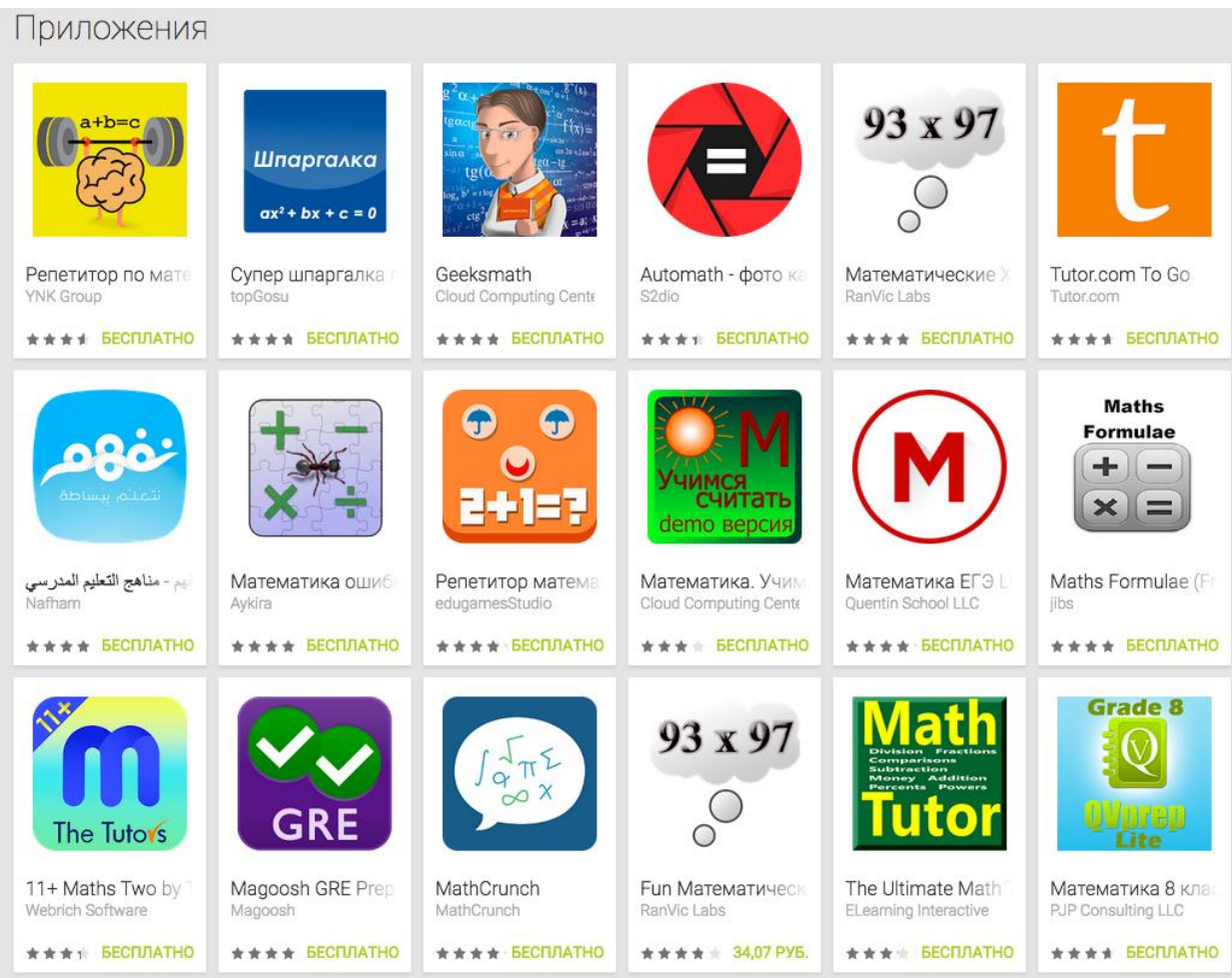


Рисунок 1. Приложения Google Play, связанные с репетиторством

В процессе анализа большинства существующих мобильных Android-приложений в сфере частного образования, было выявлено, что практически все приложения можно разделить на следующие категории:

- Дубликаты существующих учебных и дидактических материалов для общеобразовательных учреждений. Такие приложения состоят из электронного варианта того или иного учебника и разбора решений типовых задач;
- Площадки для поиска репетиторов и частных преподавателей;
- Шпаргалки с основными теоретическими материалами по различным предметам;
- Сборники готовых решений и ответов для заданий из самых распространённых образовательных материалов;

- Сервисы онлайн и офлайн помощи ученикам и студентам. Другими словами, решения заданий из разных отраслей образовательной программы для средних и высших учебных заведений на заказ;
- Электронные журналы для преподавателей.

### 3.2. iOS и App Store

iOS[5] – операционная система для смартфонов, электронных планшетов и носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple. В отличие от Windows Phone (Microsoft) и Android (Google), выпускается только для устройств, производимых фирмой Apple.

App Store – магазин мобильных приложений, раздел онлайн-супермаркета iTunes Store, содержащий различные приложения для мобильных телефонов iPhone, плееров iPod Touch и планшетов iPad, а также для персональных компьютеров Mac. App Store является единственным официальным магазином приложений для iOS-устройств.

В App Store как и в Google Play присутствует огромное количество образовательных приложений, в том числе связанных со сферой репетиторства (Рисунок 2).

#### Приложения для iPhone

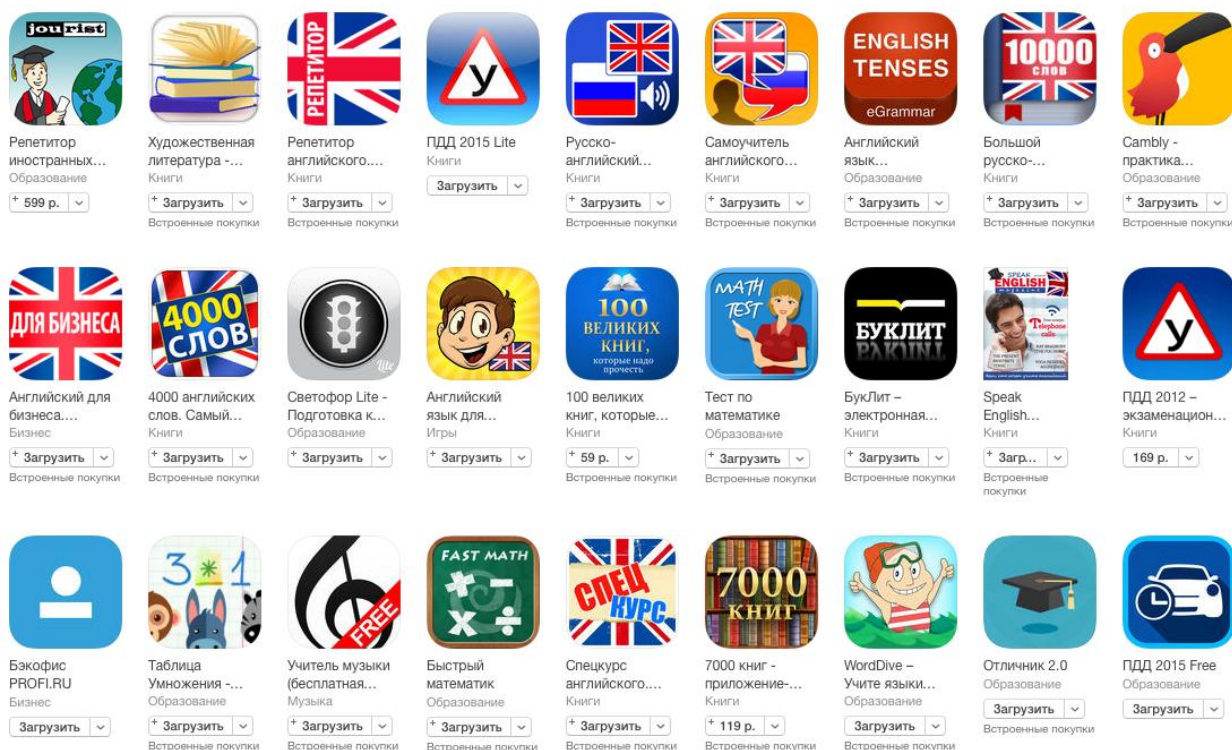


Рисунок 2. Образовательные мобильные приложения в App Store

После детального рассмотрения большинства приложений в сфере образования и репетиторства, представленных в App Store, можно разбить их на следующие виды:

- Словари и самоучители иностранных языков;
- Электронные версии общеобразовательных учебников и готовые домашние задания к ним;
- Сервисы связи между преподавателем и учениками, позволяющие пересылать документы и текстовые сообщения;
- Калькуляторы и вычислители разных уровней сложности;
- Развивающие математические игры для детей;
- Сервисы поиска частных преподавателей.

### 3.3. Windows Phone и Windows Phone Store

Windows Phone [6]– мобильная операционная система, разработанная Microsoft, вышла 11 октября 2010 года. В России телефоны с Windows Phone начали продаваться 16 сентября 2011 года. Детальное описание данной операционной системы представлено в параграфе 6 настоящей работы.

Windows Phone Store (ранее Windows Phone Marketplace) - магазин приложений Microsoft для собственной мобильной платформы Windows Phone.

Windows Phone Store значительно уступает Google Play и App Store по количеству доступных для загрузки приложений. Исключением не является и раздел образовательных программных продуктов, например, по запросу "репетиторство" в Windows Phone Store отображается всего 3 приложения (Рисунок 3), лишь одно из которых реально связано с учебным процессом.



Рисунок 3. Поиск приложений по запросу "репетиторство" в Windows Phone Store

### 3.4. Вывод

Итогом работы над данным обзором является набор выводов, связанных практически со всем рынком мобильных приложений в сфере образования и репетиторства.

Подавляющее большинство мобильных образовательных приложений ориентированы на обучающихся. В остальных приложениях, обеспечивающих связь между учащимся и преподавателем, не реализованы следующие возможности:

- загрузка полноценного предметного курса, состоящего из тематических разделов и соответствующих домашних заданий;
- автоматическая рассылка домашних заданий;
- автоматическая проверка правильности введенных ответов.

Выбор Windows Phone в качестве платформы для разрабатываемого приложения является наиболее оправданным вариантом, если подходить к рассмотрению данного вопроса с точки зрения количества конкурирующих образовательных приложений в рамках одной платформы.

В Google Play, App Store и Windows Phone Store не содержится программного продукта, аналогичного по функциональным возможностям приложению, разработанному в результате данной дипломной работы.



## 4. Облачные вычисления

Облачные вычисления (cloud computing) являются одним из наиболее популярных направлений развития ИТ. Понятие облака (cloud) уже давно ассоциируется с метафорическим изображением Интернета, с помощью которого доступны некоторые сервисы. Облачные вычисления – это практическая реализация данной идеи. Облачные вычисления основаны на масштабированных и виртуализованных ресурсах (данных и программах), которые доступны пользователям через Интернет и реализуются на базе мощных центров обработки данных. Рисунок 4 иллюстрирует общую структуру облака (cloud) и демонстрирует его возможных клиентов.



Рисунок 4. Архитектура облачных вычислений



#### 4.1. Модели обслуживания в облачных вычислениях

Рассмотрим классификацию моделей обслуживания в облаке:

- Инфраструктура как сервис (Infrastructure as a Service - IaaS) – модель обслуживания клиентов облака, при которой провайдер облака предлагает реальные или виртуальные машины и их ресурсы: образы дисков, виртуальные локальные сети и др. Виртуальная инфраструктура (за небольшую арендную плату или вовсе бесплатно, причем без необходимости делать какие-либо инсталляции на своих компьютерах) – это и есть самое ценное в облачных вычислениях, одна из основных причин, почему столь большое число клиентов начало использовать облако.
- Платформа как сервис (Platform as a Service - PaaS) – модель обслуживания клиентов облака, при которой провайдер облака предлагает клиентам целую компьютерную платформу: операционную систему, окружение для выполнения программ на языках программирования, базу данных и Web-сервер. К этому классу моделей относится Windows Azure.
- Программное обеспечение как сервис (Software as a Service - SaaS) – модель обслуживания клиентов облака, при которой провайдер облака устанавливает в облаке прикладные программы, которые используются клиентами облака. Яркий пример – Google Cloud Apps, облачные решения фирмы Google, полезные приложения, которые легко интегрировать в браузеры клиентов.
- Сеть как сервис (Network as a Service – NaaS) – относительно новый вид облачных услуг, при котором провайдер облака предлагает клиентам сетевые услуги: транспорт по сети, виртуальные частные сети (VPN) и др. Пример – облачный вариант электронной почты, который бесплатно предлагается в настоящее время многими фирмами (например, Microsoft hotmail.com).

## 4.2. Виды (модели) облаков

По своим возможностям и масштабам различные виды облаков существенно различаются. Причина в том, что, как уже отмечалось, чем меньше потенциальное число пользователей облака, тем меньше ресурсов требуется, чтобы его реализовать. Выделяют следующие виды облаков:

- **Общедоступное облако (public cloud)** – модель облака, при которой облачные приложения, память и другие ресурсы доступны любому пользователю. Данная модель наиболее дорогостоящая и ресурсоемкая. Позволить себе разработать и поддерживать подобную модель может только очень крупная компания. Работа общедоступного облака базируется на нескольких крупных центрах обработки данных, каждый из которых занимает огромное здание и потребляет огромное количество электроэнергии. Примером общедоступного облака является Microsoft Windows Azure. Другой пример – облако Amazon Web Services.
- **Облако сообщества (community cloud)** – менее масштабная модель облака, при которой облачная инфраструктура предоставляется какому-либо сообществу, т.е. более узкому кругу лиц. Пример: IEEE Community Cloud. Чтобы воспользоваться этим облаком, необходимо, как минимум, стать членом IEEE.
- **Частное облако (private cloud)** – модель облака, при которой облачные услуги предоставляются только сотрудникам какой-либо организации. Создание и поддержание в рабочем состоянии такого облака вполне реально для любой фирмы, даже для небольшой. Рекомендуем начинать Ваши облачные разработки именно с создания частного облака.
- **Гибридное облако (hybrid cloud)** – модель облака, реализующая сочетание нескольких взаимосвязанных облаков (общедоступных,

облаков сообщества или частных) с целью удобства их совместного использования для решения каких-либо конкретных задач.

#### 4.3. Уровни компонент облачных вычислений

Различаются следующие уровни архитектуры облачных вычислений:

- Уровень клиента – это клиентское ПО, используемое для доступа к облачным сервисам, например, Web-браузер.
- Уровень сервисов – это сами сервисы, используемые через облачную модель.
- Уровень приложений – это программы, доступные через облако и не требующие инсталляции на компьютере пользователя (как уже отмечалось, в этом – одно из главных преимуществ облачной модели).
- Уровень платформы – это программная платформа, объединяющая полный набор инструментов для развертывания и использования облачных вычислений на пользовательском компьютере (без дополнительных инсталляций, покупки оборудования и др.). Пример такой платформы: Microsoft.NET Azure Services Platform.
- Уровень памяти – поддержка хранения данных пользователя и доступа к ним через облако. Фактически для хранения данных пользователя используется набор Web-сайтов, реализованных на компьютерах центра обработки данных.
- Уровень инфраструктуры – предоставление полной виртуализованной платформы через облако, например, Amazon Web Services.

#### 4.4. Архитектура облачных вычислений

Рассмотрим схему архитектуры облачных вычислений:

- Сервисы, доступные через облако
- Инфраструктура для их развертывания и использования

- Платформа – набор инструментов для использования облака
- Память – поддержка хранения пользовательских данных в ЦОД, реализующем облако
- Архитектор облака – главный разработчик его архитектуры.
- Интегратор облака – его системный администратор, отвечающий за добавление компонент в облако и их изменение.

Компоненты облака, как правило, являются Web-сервисами. На Рисунке 5 приведен пример типичной архитектуры облачных вычислений.

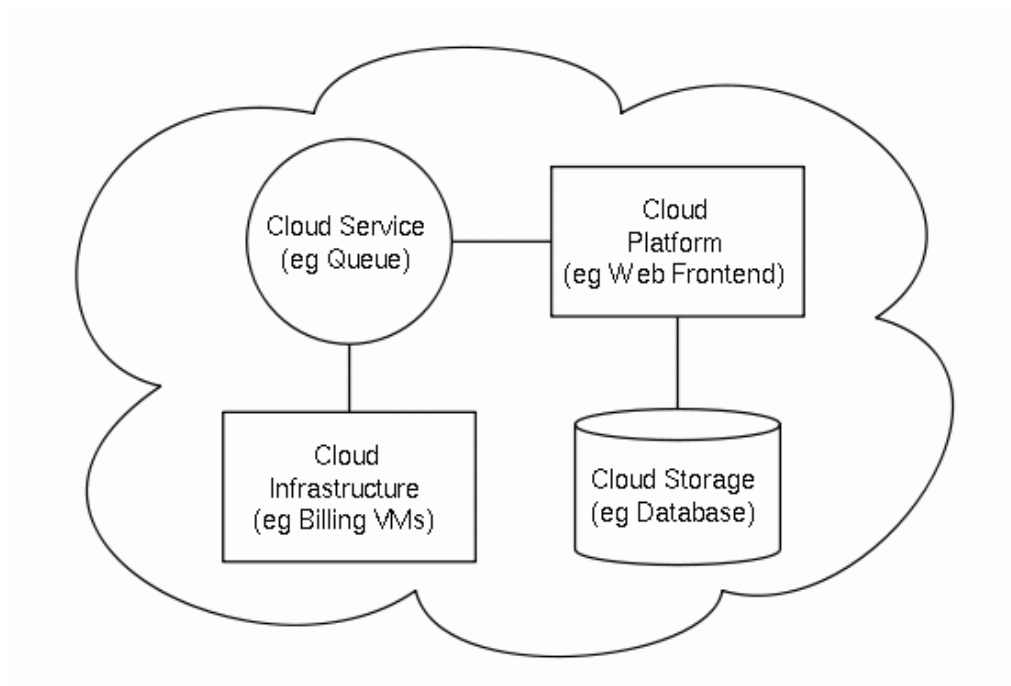


Рисунок 5. Пример организации облачных вычислений с использованием различных уровней

#### 4.5. Обзор платформ облачных вычислений

На сегодняшний день существует множество платформ облачных вычислений, рассмотрим наиболее известные из них:

- Amazon Web Services (AWS) является наиболее ранней, наиболее общей и наиболее известной из облачных сервисных платформ. Более

подробную информацию можно найти на официальном сайте облака Amazon [7].

- IBM Cloud – облачная платформа, ориентированная на уровень предприятия. Ее облачные сервисы могут предоставляться и как элементы общедоступного облака, и как компоненты частного облака. Детальнее с облаком IBM можно ознакомиться на официальном сайте [8].
- Microsoft Windows Azure - также поддерживает как публичные, так и частные облачные сервисы, основана на архитектуре .NET. Данная платформа подробно рассматривается в параграфе 5 данной работы.
- Oracle Cloud – аналогично IBM cloud, доступно как в публичной, так и в частной форме. Подробнее об облачных вычислениях фирмы Oracle можно прочитать на официальном сайте компании [9].
- Salesforce.com: Force.com cloud – легко интегрируется с программными инструментами, опубликованными на известном сайте Salesforce.com. Информация об облаке Force.com приведена на официальном сайте [10].
- Google Cloud– набор облачных сервисов, ориентированный на веб-разработчиков и приложений для веб-хостинга. Типичный пример – надстройка Google для браузеров, обеспечивающая поиск с помощью поисковой машины Google. На официальном сайте [11] приведена основная информация об облаке Google.

В качестве инфраструктурного решения для достижения целей данной работы (параграф 2.2.) была выбрана платформа облачных вычислений Microsoft Azure. Ряд преимуществ Microsoft Azure перед иными облачными средами:

- Надежное и отказоустойчивое хранилище стоит относительно недорого, данные в "облаке" могут храниться сколь угодно долго;

- Обработка данных не требует большого числа ресурсов;
- Входящий трафик бесплатен; минимальные затраты на исходящий трафик при выгрузке в локальное хранилище или СУБД;
- Наиболее развитая инфраструктура для бэкенда мобильных приложений и сервисов;
- Мобильный доступ к данным и веб-ресурсам.

Более того, согласно исследованию аналитической компании Nasuni [12], проведённому в январе 2015 года, платформа Microsoft Azure является лидером в тестах производительности при записи и чтении данных из облака, доступности данных и минимальному числу ошибок (0 %).

## 5. Microsoft Azure

Microsoft Azure (до 25.03.14 – Windows Azure) [13] – облачная платформа от Microsoft. Microsoft Azure обеспечивает хранение, использование и модификацию данных и запуск программ на компьютерах центров обработки данных Microsoft.

### 5.1. Обзор платформы

Microsoft Azure полностью реализует две облачные модели — платформы как сервиса (Platform as a Service, PaaS) и инфраструктуры как сервиса (Infrastructure as a Service, IaaS) [13].

Microsoft Azure как PaaS обеспечит не только все базовые функции операционной системы, но и дополнительные: выделение ресурсов по требованию для неограниченного масштабирования, автоматическую синхронную репликацию данных для повышения отказоустойчивости, обработку отказов инфраструктуры для обеспечения постоянной доступности.

Microsoft Azure как IaaS реализует возможность аренды таких ресурсов, как серверы, устройства хранения данных и сетевое оборудование. Управление всей инфраструктурой осуществляется поставщиком, потребитель управляет только операционной системой и установленными приложениями. Такие сервисы оплачиваются по фактическому использованию и позволяют увеличивать или уменьшать объем инфраструктуры через специальный портал, предоставляемый поставщиками. В данной сервисной модели могут быть запущены практически любые приложения, установленные на стандартные образы ОС.

Облачная платформа состоит из следующих компонент [14] (Рисунок 6):

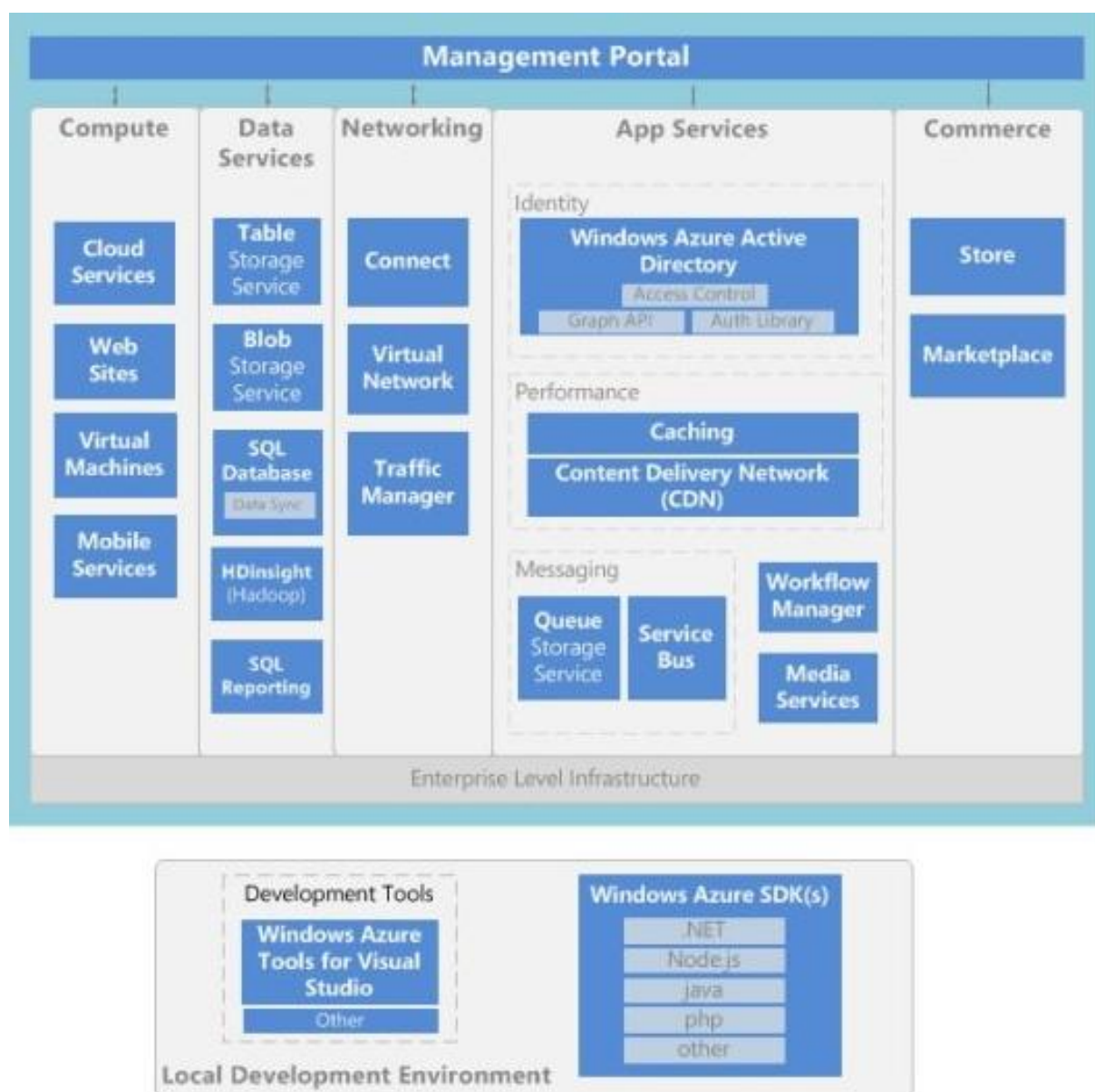


Рисунок 6. Архитектура Microsoft Azure

- **Вычисления (Compute)** – управление вычислениями в облаке: облачными сервисами, Web-сайтами, виртуальными машинами и мобильными сервисами.
- **Сервисы данных (Data Services)** – управление облачными данными и базами данных.
- **Сетевые средства (Networking)** – управление сетевыми соединениями (Connect) между локальной сетью клиента и облачными



приложениями; виртуальными частными сетями (Virtual Network); балансировкой загрузки сети (Traffic Manager).

- Управление приложениями (App Services).
- Коммерция (Commerce) – продажа (Store) и распространение (Marketplace) облачных приложений.

Детальное описание всех компонент Azure приводится в параграфе 5.2. данной работы. Взаимодействие с компонентами организуется с помощью портала управления, новый интерфейс которого представлен на Рисунке 7.

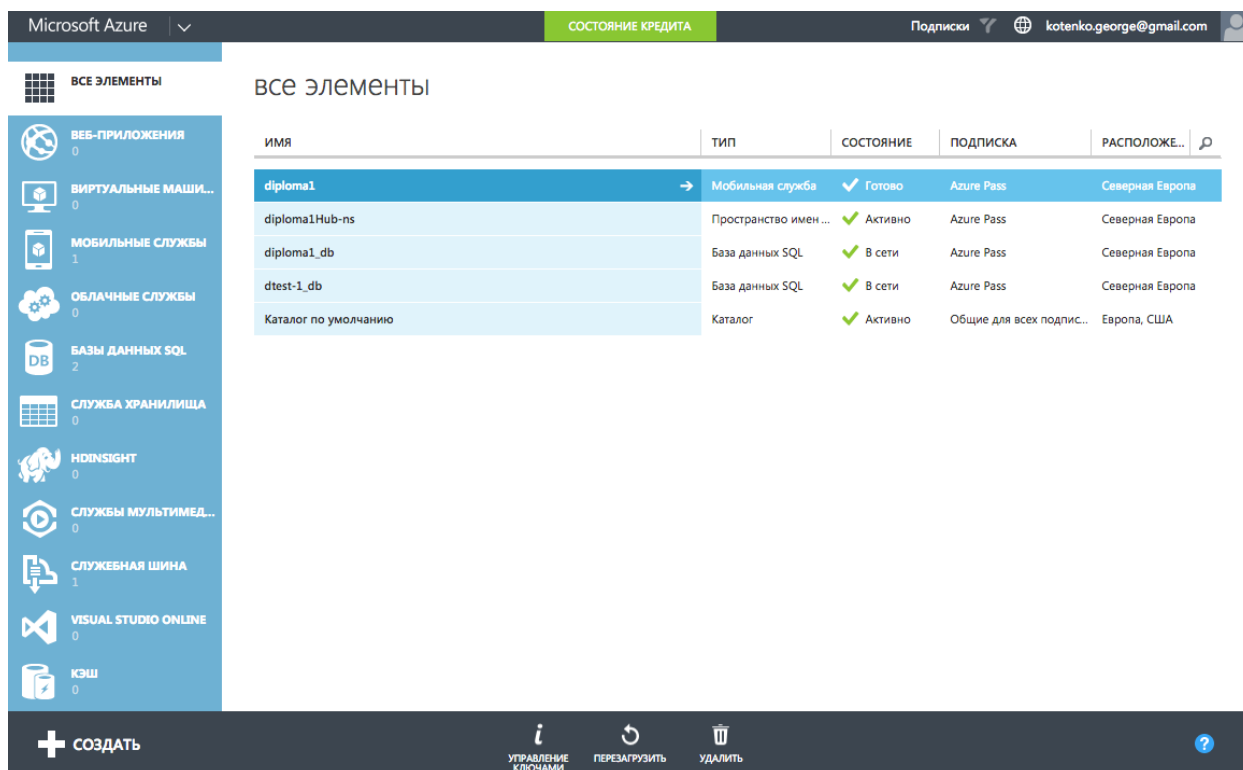


Рисунок 7. Портал управления Microsoft Azure

Реализация Microsoft Azure основана на надежной и безопасной платформе .NET, исполнение программ в которой обеспечивается в особом безопасном режиме (managed execution – управляемое выполнение). Часть .NET, называемая Windows Communication Foundation (WCF) и предоставляемый ею механизм сервисов и является основой реализации платформы Microsoft Azure.

## 5.2. Компоненты Microsoft Azure

### 5.2.1. Компонента Вычисления (Compute)

Компонента Вычисления позволяет выполнять приложения в Microsoft Azure с использованием следующих возможностей [13, 14]:

- Web-сайты (WebSites): поддерживаются ASP.NET, Java, Node.js или PHP и могут быть развернуты за секунды с использованием FTP, Git, TFS, Mercurial и Dropbox.
- Облачные сервисы (CloudServices) — быстрое развертывание многоярусных приложений и управление ими. Предлагаются Web-роль (веб-роль) и Worker-роль (рабочая роль).

Веб-роли в Microsoft Azure имеют особое назначение: предоставление выделенного веб-сервера служб IIS для размещения интерфейсных веб-приложений. Веб-роли позволяют развертывать веб-приложения с последующим масштабированием вычислительных ресурсов.

Приложения, размещенные в рабочих ролях, могут выполнять асинхронные, продолжительные или непрерывные задачи независимо от действий пользователей. Изоляция фоновых процессов приложения в рабочей роли и размещение интерфейсной части в веб-роли позволяет распределить логику приложения и управлять масштабированием приложения.

- Виртуальные машины (VirtualMachines). Получение полного контроля над виртуальным облачным сервером, в соответствии с потребностями задачи. Позволяют переносить приложения и инфраструктуру без изменения существующего исходного кода. Возможность выбора из

галереи ОС: WindowsServer или Linux с различными наборами предварительно установленных приложений.

- Мобильные сервисы (Mobile Services). Сервис предлагает облачную инфраструктуру для всех популярных мобильных платформ: Windows 8, Windows Phone, iOS и Android. На основе сервиса можно построить облачный бэкенд, на который перенести задачи по хранению данных, аутентификации и Push-уведомлений. Поддерживается Xamarin.

### 5.2.2. Компонента Сервисы данных(Data Services)

Компонента Сервисы данных (Data Services) обеспечивает возможность хранения, модификации и генерации отчетов для данных (SQL Reporting) в Microsoft Azure. Имеются следующие сервисы [13, 14]:

- Таблицы (Tables) — хранилище таблиц используется приложениями, которые хранят большие объёмы данных с дополнительными требованиями к структурированию. В таблице хранятся структурированные данные, между которыми не устанавливаются отношения.
- Блобы — хранилище BLOB-объектов — это простейший способ хранения больших объёмов неструктурированных текстовых или двоичных данных, таких как видео, музыкальные файлы и изображения.
- SQL Database — реляционная база данных — это высокодоступная масштабируемая облачная служба базы данных, построенная на основе технологий SQL Server.
- SQL DataSync — облачная служба синхронизации данных, обеспечивающая как однонаправленную, так и двунаправленную синхронизацию. Служба DataSync позволяет легко обмениваться

данными между SQL в Azure и локальными базами данных SQL Server, а также между несколькими базами данных SQL Databases (SQL Azure).

- SQL Reporting — служба Microsoft SQL Reporting позволяет легко встроить в приложение Windows Azure возможности работы с отчетами. Данная служба более не поддерживается и не разрабатывается.
- HDInsight – основан на инструменте Apache Hadoop для параллельной обработки больших объемов данных. Упрощает работу с большими данными, благодаря интеграции с инструментами типа Microsoft Office and System Center.

### 5.2.3. Компонента Сетевые средства(Networking)

Компонента Сетевые средства (Networking) предоставляет услуги сетевых соединений и маршрутизации на уровнях TCP/IP (транспортных протоколов Интернета) и DNS (преобразования доменных имен в IP-адреса). Предлагаются следующие сервисы [13, 14]:

- Сервис Соединение (Microsoft Azure Connect) обеспечивает конфигурирование безопасных IP-соединений между компьютерами или виртуальными машинами сети организации клиента и экземплярами приложений, исполняемых в Microsoft Azure.
- Виртуальная сеть Azure (Microsoft Azure Virtual Network) предоставляет возможность организации виртуальных частных сетей (VPN) в Windows Azure для их безопасной связи с локальной ИТ-инфраструктурой клиента. Сервис позволяет использовать Microsoft Azure как филиал или как защищенную частную сеть в облаке.

- Менеджер трафика (Traffic Manager) выполняет балансировку входящего трафика между различными сервисами, доступными через Интернет, с использованием политик, основанных на DNS.
- Сервис Удаленные приложения (Azure RemoteApp) позволяет размещать в облаке Azure существующие клиентские Windows-приложения и получать к ним доступ с любых компьютеров, планшетов, ноутбуков или телефонов через RDP-клиент (Windows, Mac OS X, iOS и Android).
- Экспресс маршрут (ExpressRoute) предлагает пользователям возможность подключить собственную инфраструктуру напрямую к датацентрам Microsoft Azure минуя сеть Интернет как промежуточное звено.

### 5.2.3. Компонента Управления приложениями (App Services)

Компонента Управления приложениями включает следующие сервисы [14]:

- Управления доступом и идентификацией пользователей
- Управления производительностью приложений
- Управления сообщениями между приложениями Azure
- Управления планированием работы приложений (workflowmanager)
- Управления обработкой мультимедийной информации (media services).

Управление доступом и аутентификации пользователей происходит через компоненту Azure Active Directory, которая предоставляет следующие сервисы для управления идентификацией пользователей в облачных приложениях:

- Сервис управления доступом (Access Control Service) – облачный сервис для простой аутентификации и авторизации пользователей для

предоставления им доступа к облачным приложениям и сервисам, позволяющий вынести проверки аутентификации и авторизации из кода приложения.

- Библиотека управления графом (Graph API) - обеспечивает доступ программным путем к Windows Azure Active Directory (AD) с помощью вызовов методов по стандарту REST.
- Библиотека аутентификации (Authentication Library) - предоставляет возможность разработчиком клиентских приложений аутентифицировать пользователей через Windows Azure Active Directory или других провайдеров аутентификации, а затем получать маркеры доступа для безопасных вызовов методов. Библиотека аутентификации (AAL) также дает возможность разработчикам сервисов защитить их ресурсы путем проверки входных маркеров.

Следующая группа сервисов относится к разделу управления производительностью приложений:

- Сервисы Windows Azure Caching - позволяют организовать кэширование в облаке для использования приложениями и сервисами с целью повышения их эффективности. В частности, поддерживается распространенный сценарий кэширования состояния сеанса и вывода в ASP.NET. Кэширование позволяет улучшить производительность приложений, благодаря временному хранению информации из других серверных источников.
- Сервисы Сети выдачи контента (Content Delivery Network) - кэшируют большие бинарные объекты Windows Azure и статический выходной контент вычислительных объектов в стратегически размещаемых точках для обеспечения максимальной скорости выдачи контента пользователям.

Сервисы управления сообщениями между приложениями (сообщения хранятся в различного рода очередях):

- Очереди в памяти (Storage Queues) - обеспечивают надежный обмен сохраняемыми сообщениями между ярусами приложений, выполняемых в Windows Azure. Очереди сообщений являются частью компоненты Windows Azure Storage, которая также обеспечивает сервисы бинарных объектов (Blob) и таблиц (Table).
- Шина сервисов (Service Bus) - обеспечивает безопасную и широко доступную инфраструктуру для коммуникации сервисов, широкомасштабного распределения событий между ними, их именования и публикации. Шина сервисов предоставляет возможность соединений средствами Windows Communication Foundation (WCF) и другими способами коммуникации, включая вызовы методов по стандарту REST. Имеются примеры интеграции шины сервисов в приложения, в которых реализованы мосты (bridges), преобразования (transforms) и B2B-сообщения на основе Azure Service Bus.

#### 5.2.4. Компонента Коммерция (Commerce)

Компонента Коммерция включает в себя следующие сервисы [13]:

- Store — Windows Azure Store предлагает унифицированный доступ к сервисам (не Microsoft) для проектов Microsoft Azure с единым биллингом и панелью управления.
- Marketplace — магазин облачных сервисов и данных для организаций. В настоящий момент в Marketplace доступно более 600 облачных решений и 170 источников данных.

### 5.3. Краткие итоги

Платформа облачных вычислений Microsoft Azure предоставляет набор сервисов, покрывающий очень широкий спектр сценариев. Практически все сервисы Microsoft Azure имеют API, построенное на REST, что позволяет разработчикам использовать облачные сервисы с любой операционной системы, устройства и платформы. Перспективы облачных вычислений на платформе Azure очень велики, так как они предоставляют мощные масштабируемые виртуализованные вычислительные сервисы, не требуя дополнительных ресурсов от компьютеров пользователей. Кроме того, надежность и безопасность обеспечивается базовой платформой .NET. Так же отличительными особенностями и, одновременно, преимуществами данной платформы являются оплата только потребляемых ресурсов; общая, многопоточная структура вычислений; абстракция от инфраструктуры.



## 6. Windows Phone

Согласно отчету аналитической компании International Data Corporation [1], в 2014 году Windows Phone вошел в тройку лидеров по выпуску операционных систем для смартфонов и был назван платформой с наиболее быстрыми показателями роста доли рынка (на момент публикации прирост составлял 91% в год). На рынке мобильных приложений для Windows Phone устройств практически не развит сегмент, связанный с образованием и частным репетиторством (параграф 3.3. данной работы). Поэтому именно Windows Phone (а именно последняя версия - Windows Phone 8) была выбрана как целевая клиентская платформа.

Windows Phone[6] — мобильная операционная система, разработанная Microsoft, вышла 11 октября 2010 года, 21 октября начались поставки первых устройств на базе новой платформы. Операционная система является преемником Windows Mobile, хотя и несовместима с ней, с полностью новым интерфейсом и — впервые — с интеграцией сервисов Microsoft: игрового Xbox Live и медиаплеера Zune. В отличие от предшествующей системы, Windows Phone в большей степени ориентирован на рынок потребителей, чем на корпоративную сферу.

20 июня 2012 года на организованной Microsoft конференции под названием Windows Phone Summit была анонсирована Windows Phone 8 — последняя на данный момент версия операционной системы Windows Phone. Главное преимущество Windows Phone 8 — возможность объединить планшеты, смартфоны и персональные компьютеры в единую «экосистему», то есть возможность создания условий для разработчиков, облегчающих портирование программного обеспечения между этими устройствами.

Windows Phone 8 [15] — второе поколение операционной системы Windows Phone от Microsoft. Официальный запуск состоялся 29 октября 2012 года.[1]. Система, также, как и её предшественница, использует интерфейс Modern UI.

В Windows Phone 8 используется новая архитектура Windows NT(Рисунок 8), которая используется в настольных операционных системах Microsoft. Из-за смены ядра устройства под управлением Windows Phone 7.x, построенной на ядре Windows CE, не могут обновиться до Windows Phone 8. Новые приложения, созданные для Windows Phone 8, не могут запускаться на Windows Phone 7.x, тогда как приложения, написанные для Windows Phone 7.x, работают на Windows Phone 8, будучи автоматически перекомпилированными «в облаке».

Архитектура Windows NT [16] имеет модульную структуру и состоит из двух основных уровней — компоненты, работающие в режиме пользователя, и компоненты режима ядра. Программы и подсистемы, работающие в режиме пользователя, имеют ограничения на доступ к системным ресурсам. Режим ядра имеет неограниченный доступ к системной памяти и внешним устройствам. Ядро системы NT называют гибридным ядром или макроядром. Архитектура включает в себя само ядро, уровень аппаратных абстракций (HAL), драйверы и ряд служб (Executives), которые работают в режиме ядра (Kernel-mode drivers) или в пользовательском режиме (User-mode drivers).

Пользовательский режим Windows NT состоит из подсистем, передающих запросы ввода-вывода соответствующему драйверу режима ядра посредством менеджера ввода-вывода. Есть две подсистемы на уровне пользователя: подсистема окружения (запускает приложения, написанные для разных операционных систем) и интегрированная подсистема (управляет

особыми системными функциями от имени подсистемы окружения). Режим ядра имеет полный доступ к аппаратной части и системным ресурсам компьютера. И также предотвращает доступ к критическим зонам системы со стороны пользовательских служб и приложений.

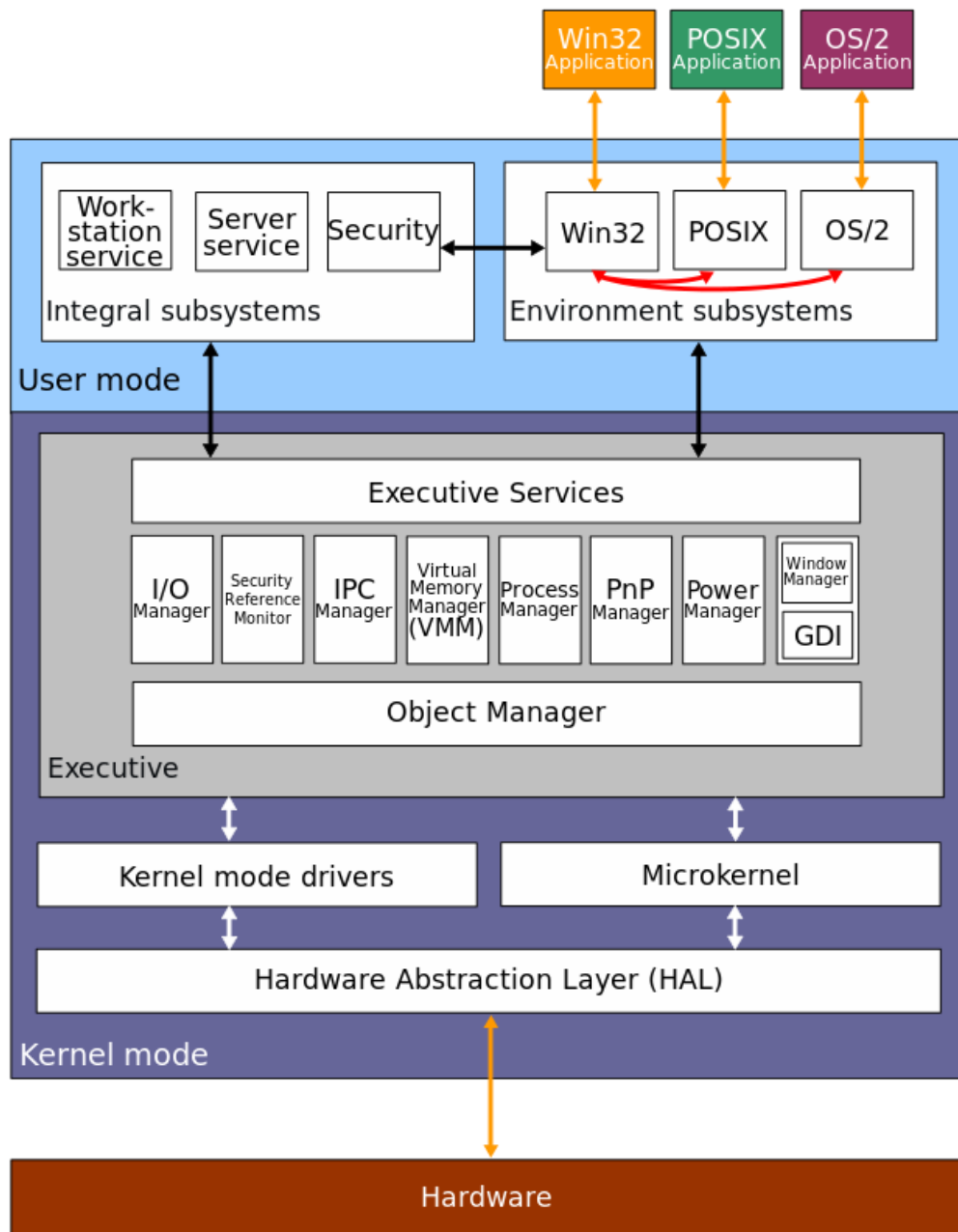


Рисунок 8. Архитектура Windows NT

## 7. Языки программирования

Языком разработки клиентского мобильного приложения является язык программирования C#, охватывающий несколько парадигм программирования, включая статическую типизацию, императивный, декларативный, функциональный, обобщенный, объектно-ориентированный и компонентно-ориентированный подходы.

При реализации API для доступа к данным мобильного облачного сервиса используется прототипно-ориентированный сценарный язык программирования JavaScript.

### 7.1. C#

Язык C# [17] создан компанией Microsoft в рамках инициативы .NET в 1998 – 2001 годах группой инженеров под руководством Андерса Хейлсберга и позже стандартизирован как ECMA-334 и ISO/IEC 23270:2006. C# - один из языков, разработанных для Common Language Infrastructure.

Язык C# поддерживает полиморфизм, перегрузку операторов (явного и неявного приведения типов), делегаты, атрибуты, события, свойства, обобщенные типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Согласно стандарту ECMA, C# разработан для следующих целей:

- Язык C# задуман, как простой, современный, универсальный, объектно-ориентированный язык программирования.
- Язык и его реализации должны обеспечивать поддержку таких принципов разработки программного обеспечения, как: сильная типизация, проверка выхода за границы массива, обнаружение использования неинициализированных переменных и автоматическая

сборка мусора. Надежность, долговечность ПО и продуктивность программистов имеет важное значение.

- Язык задуман для использования в разработке компонент ПО, развертываемых в распределенных средах
- Важное значение имеет портативность исходного кода, так же как и портативность программиста, особенно программиста, знакомого с языками C и C++
- Интернационализация имеет важное значение
- Язык C# задуман, как пригодный для написания приложений встроенных и общих систем, от самых огромных, использующих сложные ОС, до очень малых, предназначенных для определенных функций.
- И хотя приложения C# задуманы быть экономичными относительно требуемой памяти и производительности, язык напрямую не соперничает в скорости и объеме с языком C или ассемблером.

## 7.2. Javascript

Язык JavaScript [18] обычно используется как встраиваемый язык для программного доступа к объектам приложений. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. Языком JavaScript не владеет никакая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке. Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг,

анонимные функции, замыкания — что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

- объекты, с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

В языке отсутствуют такие полезные вещи, как:

- модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
- стандартные интерфейсы к веб-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

## 8. Реализация

Общая архитектура проекта (Рисунок 9) состоит из клиентского мобильного приложения для платформы Windows Phone 8, мобильного облачного сервиса и службы, отвечающей за соединение между ними. Так как служба Windows Communication Foundation, реализующая соединение между облачным сервером и клиентским приложением является частью фреймворка .NET, который изначально интегрирован с платформой облачных вычислений Microsoft Azure, то на самом деле необходимо было реализовать только две компоненты архитектуры: клиентскую и облачную.

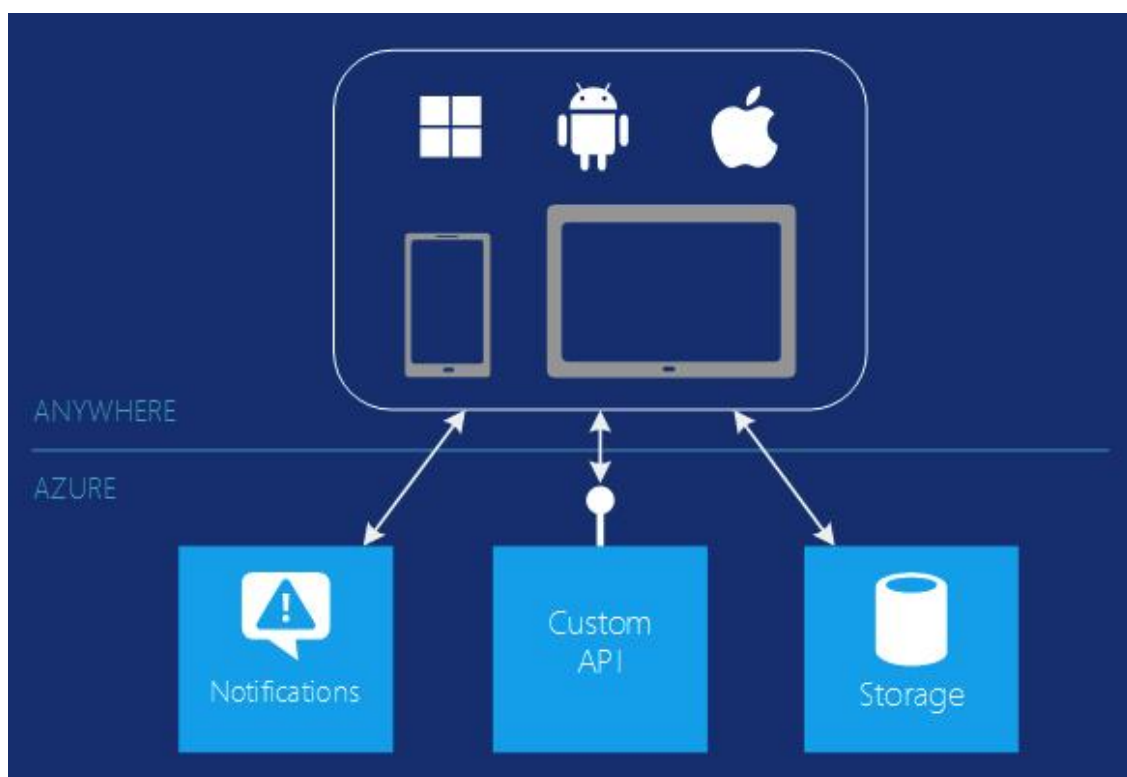


Рисунок 9. Общая архитектура проекта

## 8.1. Облачная часть

Облачная часть представляет собой облачную службу Microsoft Azure, включающую:

- Базу данных SQL – содержит таблицы с информацией о доступных курсах, домашних заданиях по каждому курсу, задачах для каждого домашнего задания;
- API для работы с базой данных – набор обработчиков запросов от клиентских приложений. Описаны все встречающиеся запросы, каждому запросу соответствует один скрипт написанный на языке JavaScript. На Рисунке 10 показан обработчик запроса на получение в поле response очереди из всех задач, соответствующих домашнему заданию, id которого был получен в поле request;
- Планировщик – внутренний таймер сервиса;
- Push-уведомления – отправляются на клиентские устройства. Уведомление отправляется, когда планировщик определяет, что необходимо опубликовать новое домашнее задание в списке актуальных домашних заданий (каждое дз имеет свою заранее обозначенную дату публикации). Непосредственно в момент отправки уведомления о новом актуальном домашнем задании, происходит его отображение в соответствующей странице клиентского приложения.





Рисунок 10. Пример обработки запроса

## 8.2. Клиентская часть

Клиентская часть представляет собой страничное WPF-приложение, написанное с помощью интегрированной среды разработки программного обеспечения Microsoft Visual Studio Professional и Windows Phone SDK - комплекта средств разработки для Windows Phone.

Приложение состоит из трёх основных страниц, файла Models.cs, описывающего все модели данных (классы), используемые в данном приложении, и файла Helpers.cs, реализующего класс, помогающий загрузить изображения в облачную базу данных.

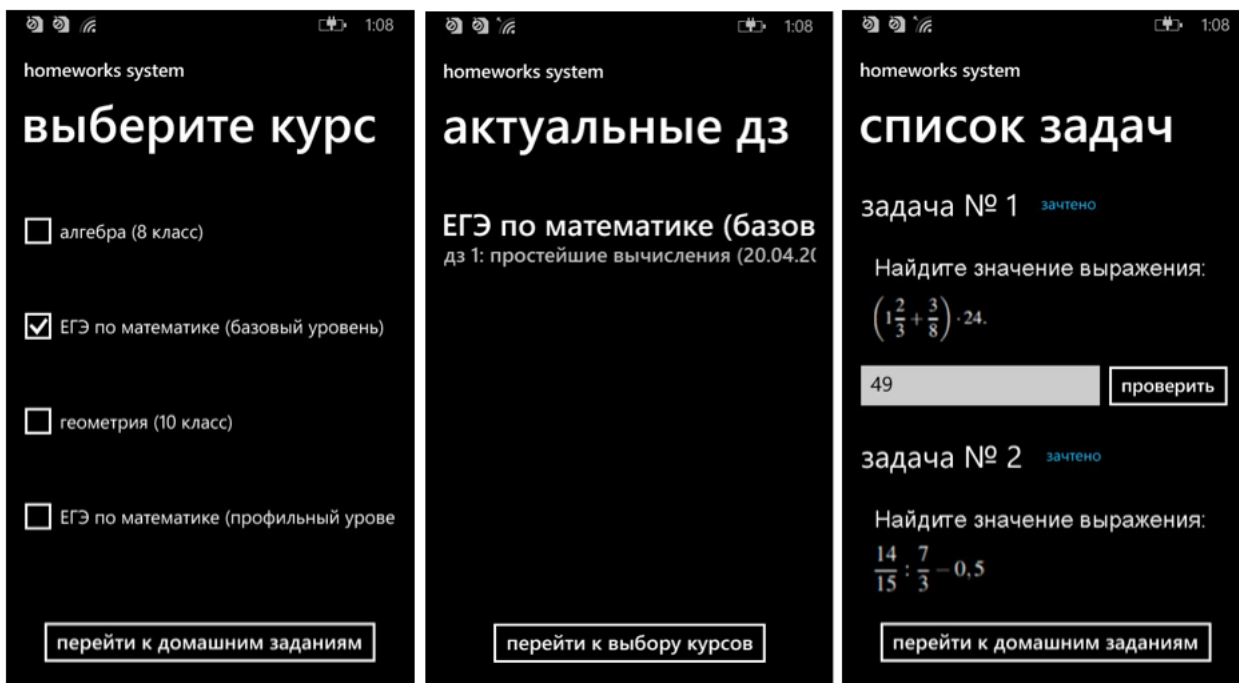
Каждый элемент страничного интерфейса состоит из файла разметки страницы на языке XAML и code-behind файла на языке C#: CoursesPage.xaml (страница с доступными курсами, Рисунок 11) и CoursesPage.xaml.cs, HomeworksPage.xaml (страница с актуальными домашними заданиями по выбранным курсам, рисунок 12) и HomeworksPage.xaml.cs,

ProblemsListPage.xaml (страница задач выбранного домашнего задания, Рисунок 13) и ProblemsListPage.xaml.cs.

Файлы разметки страниц содержат стандартный для технологии Windows Presentation Foundation набор классов (и их свойства):

- Grid – определяет гибкую область сетки, состоящую из столбцов и строк;
- StackPanel - располагает дочерние элементы в одну строку, которую можно ориентировать по горизонтали или по вертикали;
- TextBlock - Обеспечивает упрощенное управление отображением небольшого количества потока содержимого;
- ListView - Предоставляет элемент управления списка Windows, отображающий коллекцию элементов, которые могут быть выведены на экран при использовании одного из четырех различных представлений;
- CheckBox - представляет стандартный флажок выбора;
- Button - представляет элемент управления "кнопка".

Переходы между страницами реализованы с помощью службы навигации WPF – методов Navigate(), GoBack() и прочих.



*Рисунок 11.  
Страница доступных курсов*

*Рисунок 12.  
Страница актуальных ДЗ*

*Рисунок 13.  
Список задач из ДЗ*

При написании приложения использовался стандартный стиль программирования на основе событийной модели.

Первый вариант события – действие пользователя, влекущее переход на новую страницу. При отображении новой страницы необходимо обратиться к облачному сервису для получения данных для отображения. Например, при отображении страницы ProblemsListPage, показывающей список задач выбранного домашнего задания, необходимо получить данные об этих задачах. Это реализуется с помощью обращения к соответствующему элементу API облачного сервиса – "getproblems", который по полученному id домашнего задания возвращает очередь из задач, соответствующих данному дз. Ниже приведен фрагмент кода, реализующего данный запрос к API облачной службы со стороны клиентского приложения:

```
ProblemsList.ItemsSource =
(await App.MobileService.InvokeApiAsync<Problems[]>
("getproblems", HttpMethod.Get, new Dictionary<string, string>()
{ {"id", hwID} }
)
).Select((p) => new ProblemsViewModel(p));
```

Аналогично реализуется отображение других страниц, а так же варианты событий, влекущих проверку правильности введённого

пользователем ответа на конкретную задачу или обновление соответствующего поля в таблице доступных курсов, в связи с тем, что пользователь подписался на какой-либо курс либо отписался от него.

Для детального ознакомления с реализацией клиентского мобильного приложения и его использования дистрибутивы можно скачать [19] с GitHub.

## 9. Результаты и дальнейшая работа

### 9.1. Результаты

В результате данной работы был разработан Мобильный облачный сервис для Microsoft Azure и взаимодействующее с ним Windows Phone приложение, удовлетворяющее сформулированным требованиям.

Благодаря различным XML-схемам расположения элементов на экране, и, используя контроль версии ОС Windows Phone, была достигнута поддержка и корректное отображение на различных устройствах.

Использование облачной среды Microsoft Azure гарантирует доступность сервиса через интернет в любой момент времени и позволяет в дальнейшем обеспечить поддержку различных клиентских платформ.

В ходе работы были изучены различные подходы к разработке мобильных приложений и их взаимодействие с платформами облачных вычислений, освоены новые и активно развивающиеся технологии.

### 9.2. Дальнейшая работа

В качестве дальнейшей работы возможны следующие направления:

- Разработка клиентского приложения для других мобильных и иных платформ;
- Модернизация пользовательского интерфейса;
- Добавление административного интерфейса:

- Возможность использования приложения не только учениками, но и репетиторами – возможность добавления собственных курсов для своих учеников;
- Ведение статистики количества людей, подписанных, на курсы;
- Регистрация пользователей (интеграция с аккаунтами социальных сетей);
- Введение коммуникации с учениками через приложение;
- Реализация функционала изменения структуры курса или домашнего задания под каждого конкретного ученика.

## Список литературы

1. International Data Corporation (IDC). Worldwide Quarterly Mobile Phone Tracker. // [https://www.idc.com/tracker/showproductinfo.jsp?prod\\_id=37](https://www.idc.com/tracker/showproductinfo.jsp?prod_id=37)
2. Json&Partners. Аналитика рынка мобильных приложений. // [http://www.json.ru/poleznye\\_materialy/free\\_market\\_watches/analytics/rynok\\_mobilnyh\\_prilozhenij\\_v\\_rossii\\_i\\_mire/](http://www.json.ru/poleznye_materialy/free_market_watches/analytics/rynok_mobilnyh_prilozhenij_v_rossii_i_mire/)
3. Ассоциация репетиторов. Индекс репетиторства. // <http://blog.repetit.ru/2014/09/29/repetitorstvo-2013-2014-god/>
4. Android OS // <https://ru.wikipedia.org/wiki/Android>
5. iOS // <https://ru.wikipedia.org/wiki/IOS>
6. Windows Phone OS // [https://ru.wikipedia.org/wiki/Windows\\_Phone](https://ru.wikipedia.org/wiki/Windows_Phone)
7. Amazon Cloud Drive // [https://www.amazon.com/clouddrive/home/190-7797696-3967456?encoding=UTF8&ref=sa\\_menu\\_acd\\_urc&tag=95060-20](https://www.amazon.com/clouddrive/home/190-7797696-3967456?encoding=UTF8&ref=sa_menu_acd_urc&tag=95060-20)
8. IBM Cloud // <http://www.ibm.com/cloud-computing/us/en/?lnk=buc1>
9. Oracle Cloud // <https://cloud.oracle.com/home>
10. Salesforce.com : Force.com // <http://www.salesforce.com/platform/solutions/employee-apps/>
11. Google Cloud Platform // <https://cloud.google.com/appengine/docs>
12. Nasuni – The State Of Cloud Storage 2015 // <http://www.nasuni.com/resource/the-state-of-cloud-storage-in-2015>
13. Microsoft Azure // [https://ru.wikipedia.org/wiki/Microsoft\\_Azure](https://ru.wikipedia.org/wiki/Microsoft_Azure)
14. Сафонов В.О., 2013. «Развитие платформы облачных вычислений MicrosoftWindowsAzure». // <http://www.intuit.ru/studies/courses/11007/1117/info>
15. Windows Phone 8 // [https://ru.wikipedia.org/wiki/Windows\\_Phone\\_8](https://ru.wikipedia.org/wiki/Windows_Phone_8)
16. Архитектура Windows NT // [https://ru.wikipedia.org/wiki/Windows\\_NT](https://ru.wikipedia.org/wiki/Windows_NT)
17. Язык C# // [https://ru.wikipedia.org/wiki/C\\_Sharp](https://ru.wikipedia.org/wiki/C_Sharp)
18. Язык JavaScript // <https://ru.wikipedia.org/wiki/JavaScript>

19. "Система домашних заданий" – исходные коды. //

<https://github.com/pedoegor/HomeworksSystem.git>