**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПМ.03 Участие в интеграции программных модулей**

**Группа: 4ПКС-115**

**УТВЕРЖДАЮ**

**Председатель цикловой комиссии**

**программирования и баз данных**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Пестов А.И./**

**\_\_\_\_.\_\_\_\_. 2018**

**ПРОЕКТ КУРСОВОЙ**

**На тему: Программа для просмотра изображений с графическим интерфейсом на языке C#**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Руководитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Морозова М.В./**

**Исполнитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Деменчук Г.М./**

**Оценка за проект:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_.\_\_\_\_.2018**

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc523600377)

[1 ОБЩИЙ РАЗДЕЛ 4](#_Toc523600378)

[1.1 Системные требования 4](#_Toc523600379)

[1.2 Характеристика системы программирования 4](#_Toc523600380)

[2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 5](#_Toc523600381)

[2.1 Предпроектное исследование предметной области 5](#_Toc523600382)

[2.2 Анализ требований и определение спецификация программного обеспечения 5](#_Toc523600383)

[2.3 Проектирование программного обеспечения 5](#_Toc523600384)

[2.4 Разработка пользовательских интерфейсов программного обеспечения 5](#_Toc523600385)

[2.5 Тестирование и отладка программного обеспечения 5](#_Toc523600386)

[3 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ 6](#_Toc523600387)

[3.1 Руководство программиста 6](#_Toc523600388)

[3.2 Руководство пользователя 6](#_Toc523600389)

[ЗАКЛЮЧЕНИЕ 7](#_Toc523600390)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 8](#_Toc523600391)

[ПРИЛОЖЕНИЕ А 9](#_Toc523600392)

[ПРИЛОЖЕНИЕ Б 10](#_Toc523600393)

# ВВЕДЕНИЕ

На сегодняшний день оповещения являются неотъемлимой частью нашей жини. Все оповещения по достаточно просто разделить по приоритету:

* Телефонные вызовы;
* SMS-оповещения;
* Оповещения от различных мессенджеров;

Как мы видим, телефонные вызовы являются наиболее приоритетными, но они требуют немедленного реагирования от пользователя и заставляют прерываться от его деятельности. SMS-оповещения не требуют немедленной реакции, при этом его получат болеее широкое количество пользователей по сравнению с оповещениями мессенджеров. По состоянию на 2018 год по данным Pew Research Center 59% опрошенных взрослых в мире являются владельцами смартфонов, а 31% пользуются простыми мобильными телефонами, единственный недостаток заключается в том, что данный вид оповещений является платным. При рассмотрений оповещений от мессенджеров мы игнорируем людей, у которых простые (кнопочные) мобильные телефоны, но при этом данный вид оповещений является бесплатным.

Взвесив все преимущества и недостатки, выбор был сделан в пользу SMS как наиболее сбалансированный тип оповещений т.к. нет смысла игнорировать аудиторию простых кнопочных телефонов

# 1 ОБЩИЙ РАЗДЕЛ

## 1.1 Системные требования

Системные требования к электронной вычислительной машине:

* Операционная система: Windows 7 и более новые;
* Процессор с частотой 1.6 ГГц и выше;
* 1 Гб оперативной памяти;
* 500 Мб свободного пространства на жестком диске;
* Видеоадаптер с поддержкой DirectX 9, минимально допустимое разрешение экрана – 1024 х 768.

## 1.2 Характеристика системы программирования

# 2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 2.1 Предпроектное исследование предметной области

При разработке программного решения наиболее остро встает вопрос об источнике получения расписания учебного заведения и его последующего анализа. Далее будут рассмотрены источники расписания именно для Колледжа Информатики и Программирования при ФУ РФ. На данный момент существует три наиболее рациональных источника данных:

* Электронный журнал колледжа по адресу kip.eljur.ru
* Расписание в виде PDF на сайте kip.fa.ru
* Физическая распечатка PDF в колледже на стойке информатики

Сразу же мы можем исключить парсинг физической распечатки PDF по фотографии с телефона т.к. для подобного анализа изображений, требуется достаточно много условий:

* Большое количество операций по препроцессингу изображений: от банального размытия по гауссу, заканчивая вычислением пропорций между контурами ячеек с названием пары от ячеек с названием группы и выправления горизонта с помощью детектора границ Кэнни, определения прямых линий по преобразованию Хафа и вычисления результирующего угла между перпендикулярами, проходящими через центр изображения.
* Более тонкая настройка файла rus.traineddata для анализа кириллического шрифта с помощью open-source технологии Google Tesseract OCR;
* Возможная визуальная ограниченность для камеры телефона (лист бумаги находится за стеклом и отсвечивает, вспышка);
* При архитектуре standalone-приложения будет огромнейшая нагрузка на аппаратные средства мобильного устройства и как следствие, его быстрый разряд, при клиент-серверном решении требуется активное интернет-соединение;
* Большой процент риска человеческого фактора (лист бумаги может кто-то подменить в шутку, украсть и т д)

Электронный журнал колледжа, именно АИС «Электронный журнал» мы также не можем рассматривать в качестве достоверного источника информации о расписании т.к для оформления процедуры официального досупа к API системы требуется достаточно большое количество времени, а в противном случает мы нарушаем лицензию использования АИС и коэффициент рисков и времени разработки достаточно высок. В данном способе возможно три направления, расположенных в порядке «легальности»:

* Направление официального запроса к АИС с указанием целей, для которых мы планируем использовать доступ к API, а также информацией о нас, далее простой парсинг json по devkey на стороне клиента C#;
* Эмуляция браузера с поддержкой javascript и парсинг тегов HTML-документа с последующую их конвертацию в JSON, который будет отдаваться WEB API нашего сервера;
* Эмуляция клиента мобильного приложения IOS/Android и парсинг JSON/XML, который мы получили от backend-сервера АИС. Можно подменить headers непосредственно на клиенте C#, можно через прослойку в виде нашего backend-сервера, который может отвечать за прямое взаимодействие с сервером АИС.

Дать официальный запрос мы не можем т.к. аудитория возможных пользователей нашего сервиса слишком мала и время достаточно сильно ограничено.

Второй способ является сбалансированным, но самым нестабильным т.к. при изменении разметки страницы весь алгоритм парсинга перестает выполнять свою работу. Существенным плюсом является возможность узнавать расписание любых групп (нет ассоциации пользователь -> группа, в отличии от мобильного приложения), за анализ страницы может отвечать библиотека BeautifulSoup.

Третий способ представляет из себя подмену https-сертификата приложения для осуществления атаки по типу MITM (человек посередине) т.к. у большинства приложений отсутствует проверка на подмену сертификата, кроме банковских. Возможно получить только расписание своей группы т.к. есть четкая ассоциация токен -> группа. Токен работает ограниченный промежуток времени. За получение неправомерного доступа к компьютерной информации, а также создание компьютерных программ, заведомо предназначенных для осуществления несанкционированного доступа к информации, установлена ответственность в соответствии со статьями 272 и 273 Уголовного кодекса Российской Федерации. Способ не подходит.

Последним источником информации о расписании остается расписание занятий на сайте Колледжа Информатики и Программирования при ФУ РФ kip.fa.ru. Также есть свои преимущества и недостатки.

Преимущества:

* Достаточно прост процесс получения исходного файла для обработки: обыкновенный get-запрос по статической ссылке;
* Возможно применить гибридный парсинг исходного файла как растрового изображения в формате .png или .jpg с использованием технологии OpenCV и Tesseract OCR, так и векторного файла .PDF с символами, что в итоге многократно уменьшит риск каких-либо синтаксических ошибок в тексте расписания;
* Не требует глубокого препроцессинга исходного изображения и не нарушает какие-либо права, в отличии от вышеописанных способов.

Недостатки:

* В любой момент Финансовый Университет может отказаться от него, закрыть доступ и полностью перейти на АИС «Электронный журнал», что и произошло в процессе разработки данного решения;
* Иногда обновленное расписание может выкладываться с задержкой;
* Извлечение данных даже с векторного PDF является достаточно трудоемкой задачей;
* Возможный риск человеческого фактора.

В данном способе также существует несколько векторов развития событий:

* Можно парсить кароч как нормальное изображение, можно как PDF, можно гибридно

Наиболее рациональным способом получения расписания является совокупность всех данных из всех трех источников (сайт колледжа, эл. Дневник, и парсинг физической распечатки)

был выбран способ парсинга в виде изображения т.к. хотелось попробовать использовать flesk api, библиотеку компьютерного зраения opencv и open-source технологию распознавания контуров текста на изображении goole tessarect orc.

## 2.2 Анализ требований и определение спецификация программного обеспечения

Диаграмма вариантов использования представлена на рисунке 2.1, и состоит из действий, совершаемых пользователем в программе.

[РИСУНОК ДИАГРАММЫ]

В клиенте С# для рассылки расписания по SMS у пользователя есть возможность:

* При знании мастер-пароля – добавление новых пользователей

## 2.3 Проектирование программного обеспечения

## 2.4 Разработка пользовательских интерфейсов программного обеспечения

## 2.5 Тестирование и отладка программного обеспечения

# 3 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ

## 3.1 Руководство программиста

## 3.2 Руководство пользователя

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ А

# ПРИЛОЖЕНИЕ Б