

**Разработка серверного модуля информационной системы для информирования студентов университета**

*Якимов Антон Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема  
студент*

*Малиевский Ян Генрихович*

*Приамурский государственный университет имени Шолом-Алейхема  
студент*

*Баженов Руслан Иванович*

*Приамурский государственный университет имени Шолом-Алейхема  
к.п.н., доцент, зав. кафедрой информационных систем, математики и  
методик обучения*

**Аннотация**

На сегодняшний день с помощью мобильных технологий человек может мгновенно получать любую информацию, прилагая при этом минимум усилий. В данной статье рассматривается разработка серверного модуля информационной системы, которая позволит оповещать студентов о различных информациях с помощью мобильного приложения и всегда быть в курсе дела.

**Ключевые слова:** расписание студентов, парсер, база данных, сервер, мобильное приложение

**Development of backend information system to inform students of the university**

*Yakimov Anton Sergeevich*

*Sholom-Aleichem Priamursky State University  
Student*

*Malievskij Jan Genrihovich*

*Sholom-Aleichem Priamursky State University  
Student*

*Bazhenov Ruslan Ivanovich*

*Sholom-Aleichem Priamursky State University  
candidate of pedagogical sciences, associate professor, Head of the Department of  
Information systems, Mathematics and teaching methods*

**Abstract**

Today, with the help of mobile technologies people can instantly obtain any information, while making the least effort. This article discusses the development of backend information system, which will inform the students about the different information through mobile applications and always be aware of the case.

**Keywords:** students schedule, parser, database, backend, mobile app

В наше время использование мобильных технологий позволяет быть в курсе всех событий в мире, прилагая для этого минимум усилий. Кроме того, мобильные технологии позволяют снизить стоимость продукции для конечных потребителей за счет оптимизации процессов, сокращения производственных издержек и непроизводственных затрат.

В данном случае рассмотрим разработку информационной системы, которая позволяет информировать и оповещать студентов университета о различных объявлениях и обновлении расписании.

Актуальностью данной проблемы заключается в том, что существующая сейчас система не эффективна, так как подобные информации находятся в корпусе университета, и при этом также приходится на месте связываться с деканатом, для получения каких-либо новостей о мероприятиях и других изменениях.

Подобные проблемы были описаны в таких работах, как проектирование и разработка мобильного приложения «Расписание МГТУ» на платформе операционной системе Android, который описал М.А.Сухов [4]. А.В.Митлошук и Т.В.Князькова продемонстрировали мобильное приложение для просмотра расписания студентов [5]. А.С.Бородин и В.В.Бурлуцкий отобрали работу о мобильном расписании для высших учебных заведений [6]. И.А.Труфанов описал разработку мобильного приложения «Расписание ФинУниверситета» [7] и т.д.

Разрабатываемый серверный модуль должен отвечать следующим требованиям:

- распознавать Excel-шаблона расписания студентов;
- хранить полученную информацию в базу данных;
- получать данные с существующей базы данных балльно-рейтинговой системы данного университета;
- принимать запросы со стороны мобильного приложения;
- осуществлять поиск в базе данных;
- сформировать нужную информацию на мобильное приложение для оповещения студентов.

Для реализации серверного модуля данной информационной системы, нужно в первую очередь разработать структуры базы данных. Для построения базы данных (рис. 1) необходимо определить сущности и атрибуты, определить связь между ними.

Для организации и ведения базы данных была выбрана система управления базы данных MySQL [1].

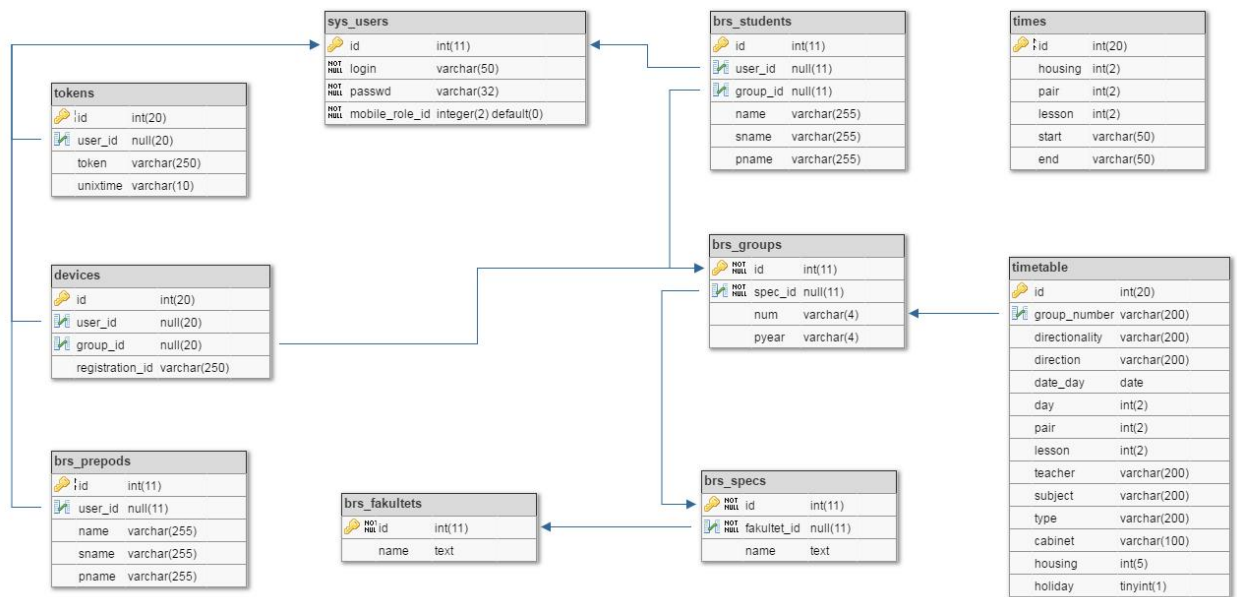


Рисунок 1 – Физическая схема базы данных

Далее нужно произвести анализ существующего Excel-шаблона расписания студентов (рис. 2), которое публикуется в бумажном виде на стенде данного университета [2].

Федеральное государственное образовательное учреждение высшего профессионального образования "Федеральное государственное университет имени Шереметьевых" Всероссийский государственный университет имени Шереметьевых Факультет математики, информатики, экономики и истории на Московском шоссе 2016-2018 гг. (10)						
Направление подготовки	44.03.05 ПО	09.03.03 ПИ	09.03.02 ИСиТ	44.03.05 ПО	09.03.02 ИСиТ	
Направленность	ИнМ	ПИЭ	ИСИТыП	ИнМ	ИСИТыП	
Номер группы	1251	1551	1751	1241	1741	
Сем. группа	1 курс			2 курс		
П о н е д е л ь н и к	21 марта	1	Математика (математический анализ) (лк), Эйрих Н.В., ауд. 304			
		2	Математика (математический анализ) (пр), Эйрих Н.В., ауд. 304			
		3	Физическая культура и спорт (группа 4 М), Изосимов А.Н., спортзал			Численные методы (лк), Эйрих Н.В., ауд. 303,
		4	Физическая культура и спорт (группа 7 Ж), Кирьянцева Л.П., спортзал		Архитектура компьютера (лк), Плахотная Л.А., ауд. 306	Иностранный язык (пр), Хозяйкина Г.С., ауд. 303
		5	Операционные системы, сети и интернет - технологии (лк), Лучанинов Д.В., ауд. 408		Архитектура компьютера (пр), Плахотная Л.А., ауд. 306	Численные методы (лк), Эйрих Н.В., ауд. 303
		6	Программирование (пр), Лучанинов Д.В., ауд. 408			
В т о р п и к	22 марта	1				
		2	Экономическая теория (лк), Голык А.С., ауд. 303	Иностранный язык (пр), Кашпаур В.С., ауд. 416	Философия (лк), Цыцарев А.А., ауд. 216	
		3	Физическая культура и спорт (группа 4 М), Изосимов А.Н., спортзал		Философия (пр), Цыцарев А.А., ауд. 31	
		4	Физическая культура и спорт (группа 7 Ж), Кирьянцева Л.П., спортзал		Математическая логика (лк), Сырнова А.С., ауд. 206	
		5	Математический анализ (пр), Эйрих Н.В., ауд. 303	Исследование операций (лк), Кириллова Д.А., ауд. 306	Математическая логика (пр), Сырнова А.С., ауд. 206	

Рисунок 2 – Excel-шаблон расписания студентов

После данного анализа мы выявили закономерность расположения ячеек, в том числе и структуру самого содержимого, то есть текста. Далее разработали полноценный и гибкий парсер расписания с помощью языка PHP [3] на основе регулярных выражений. Парсер позволяет циклично проходить всю структуру данного шаблона, попутно распознавая

содержимое ячейки, и разбивает на такие элементы как «Преподаватель», «Дисциплина», «Тип» (лекция, практика и т.д.), «Аудитория», «Корпус» и другой информации.

```
private function parserLessonCabinet()
{
    foreach($this->cabinets as $cabinet)
    {
        $return = null;
        if($cabinet[1]){
            if(preg_match("/". $this->parserLessonTrim . $cabinet[0]. "[\.\s]*(\d+)". $th
            {
                $this->currentLesson = str_replace($finded[0], ' ', $this->currentLess
                $return = $finded[1]; break;
            }
        } else {
            if(preg_match("/". $this->parserLessonTrim . $cabinet[0] . $this->parserLes
            $this->currentLesson = str_replace($finded[0], ' ', $this->currentLess
            $return = $cabinet[0]; break;
        }
    }
}
return $return;
}
```

Рисунок 3 – Образец кода распознавания аудитория

Во время распознавания расписания парсер автоматически обрабатывает полученную информацию и в надлежащем виде сохраняет в таблицу базы данных под названием timetable.

id	group_number	directionality	direction	date_day	day	pair	lesson	teacher	subject	type	cabinet	housing	holiday
1	1251	ИиМ	44.03.05 ПО	2016-03-21	1	1	1	NULL			NULL	NULL	0
2	1251	ИиМ	44.03.05 ПО	2016-03-21	1	1	2	NULL			NULL	NULL	0
3	1251	ИиМ	44.03.05 ПО	2016-03-21	1	2	1	NULL			NULL	NULL	0
4	1251	ИиМ	44.03.05 ПО	2016-03-21	1	2	2	NULL			NULL	NULL	0
5	1251	ИиМ	44.03.05 ПО	2016-03-21	1	3	1	Изосимов А.Н.	Физическая культура и спорт (группа 4 М)	Лекция	спортзал	1	0
6	1251	ИиМ	44.03.05 ПО	2016-03-21	1	3	2	Изосимов А.Н.	Физическая культура и спорт (группа 4 М)	Лекция	спортзал	1	0
7	1251	ИиМ	44.03.05 ПО	2016-03-21	1	4	1	Кириянцева Л.П.	Физическая культура и спорт (группа 7 Ж)	Лекция	спортзал	1	0
8	1251	ИиМ	44.03.05 ПО	2016-03-21	1	4	2	Кириянцева Л.П.	Физическая культура и спорт (группа 7 Ж)	Лекция	спортзал	1	0
9	1251	ИиМ	44.03.05 ПО	2016-03-21	1	5	1	Лучанинов Д.В.	Операционные системы, сети и интернет -технологии	Лекция	408	1	0
10	1251	ИиМ	44.03.05 ПО	2016-03-21	1	5	2	Лучанинов Д.В.	Операционные системы, сети и интернет -технологии	Лекция	408	1	0
11	1251	ИиМ	44.03.05 ПО	2016-03-21	1	6	1	Лучанинов Д.В.	Программирование	Лабораторная работа	408	1	0
12	1251	ИиМ	44.03.05 ПО	2016-03-21	1	6	2	Лучанинов Д.В.	Программирование	Лабораторная работа	408	1	0
13	1251	ИиМ	44.03.05 ПО	2016-03-22	2	1	1	NULL		NULL	NULL	NULL	0
14	1251	ИиМ	44.03.05 ПО	2016-03-22	2	1	2	NULL		NULL	NULL	NULL	0
15	1251	ИиМ	44.03.05 ПО	2016-03-22	2	2	1	NULL			NULL	NULL	0
16	1251	ИиМ	44.03.05 ПО	2016-03-22	2	2	2	NULL			NULL	NULL	0
17	1251	ИиМ	44.03.05 ПО	2016-03-22	2	3	1	Изосимов А.Н.	Физическая культура и спорт (группа 4 М)	Лекция	спортзал	1	0
18	1251	ИиМ	44.03.05 ПО	2016-03-22	2	3	2	Изосимов А.Н.	Физическая культура и спорт (группа 4 М)	Лекция	спортзал	1	0

Рисунок 4 – Таблица расписаний

Данная таблица расписаний нужна для того, чтобы в дальнейшем сформировать для мобильного приложения обработанное и структурированное расписание нужной группы по определенным дням.

Для того чтобы передать какую-либо информацию мобильному приложению, нужно разработать API-класс, который является «мостом» между клиентским и серверным модулем в передаче и получении данных.

API содержит в себе девять базовых функций:

- авторизация;
- расписание;
- факультеты;
- курсы;
- группы;
- студенты;
- сообщения;
- поиск;
- балльно-рейтинговая система.

Данные функции срабатываются в зависимости от того, в какой форме был передан запрос по API с мобильного приложения. Все данные запросы «общаются» между клиентским и серверным модулем в виде JSON структуры.

В целях безопасности и надежности передачи данных, все запросы с мобильного приложения защищены от нежелательных посторонних лиц, пытающихся подделать данный запрос в адрес сервера. Также все данные запросы зашифрованы с помощью AES стандарта шифрования, как и со стороны клиентского модуля, так и серверного.

Ознакомимся со всеми функциями подробнее:

**Авторизация.**

Если со стороны мобильного приложения была запрошена авторизация пользователя, передав при этом введенный логин, пароль (данный пароль при этом передается в зашифрованном хеш виде) и идентификатор устройства, то функция сверяет полученные данные с таблицей пользователей в базе данных. Если пользователь был найден и данные верны, то возвращает статус об успешной авторизации и информацию самого пользователя, либо в противном случае, вернет статус ошибки.

В случае успешной авторизации, помимо информации пользователя, также вернет и сгенерированный идентификационный токен, который позволяет в дальнейшем держать связь с сервером без необходимости повторной авторизации при последующих других запросов.

Переданный ранее идентификатор устройства предназначен для того, чтобы посредством технологии Google Cloud Messaging могли оповещать данного пользователя какой-либо информацией или уведомлением об изменении расписания. Соответственно, данный идентификатор устройства сохраняет в таблицу базы данных под названием devices.

Идентификационный токен, который генерируется во время авторизации, также заносится в таблицу базы данных под названием tokens. Данный токен хранится в базе данных в течении 3 суток, после истечения данного срока токен самоуничтожается. Если авторизованный пользователь, попытается выполнить какие-либо другие запросы серверному модулю, то перед тем как выполнить запрос, сервер сначала проверяет на наличие идентификационного токена в базе данных, который пользователь передал во время запроса. Если токен не найден, то сервер возвращает клиентскому модулю статус о том, что токен недействителен.

```
public function user()
{
    $this->beforeCheck(array('login', 'hash', 'device'), false);

    if($auth = $this->db->getRow('SELECT id, login, passwd, mobile_role_id FF
    {
        $student = $this->db->getRow('
            SELECT s.id, s.name, s.sname, s.pname, g.num, s.group_id, s.user_
            FROM brs_students as s
            INNER JOIN brs_groups as g
            ON s.user_id = ?i AND g.id = s.group_id',
            $auth['id']
        );

        $prepod = $this->db->getRow('
            SELECT p.id, p.name, p.sname, p.pname, p.user_id
            FROM brs_prepods p
            WHERE p.user_id = ?i',
            $auth['id']
        );

        if(!$student && !$prepod)
            $this->view(500);

        if($prepod){
```

Рисунок 5 – Образец кода авторизации пользователей

### *Расписание*

Если со стороны мобильного приложения был передан запрос на получение расписания, передав при этом токен и идентификатор группы, то функция производит в таблице расписании поиск нужной информации относительно выбранной группы. После успешного поиска клиентскому модулю возвращает актуальное расписание заданной группы на текущую и следующую неделю.

```

public function timetable()
{
    $this->beforeCheck(array('group', 'token'));
    $startDay = date('Y-m-d', strtotime("monday this week"));
    $endDay    = date('Y-m-d', strtotime("sunday next week"));

    if($data = $this->db->getAll('
        SELECT id, group_number, date_day, day, pair, lesson, teacher, subject
        FROM timetable_active
        WHERE group_number = ?s AND date_day >= ?s AND date_day <= ?s',
        $this->request['group'],
        $startDay,
        $endDay
    )){
        $countDay = 0;
        $lastDate = $startDay;

        $timesList = $this->db->getAll('SELECT housing, pair, lesson, start,
        foreach($timesList as $timesL){
            $times[$timesL['housing']][$timesL['pair']][$timesL['lesson']][] =
            $times[$timesL['housing']][$timesL['pair']][$timesL['lesson']][] =
        }

        foreach($data as $row){
            if((int)$row['holiday'] == false)
            {
                if($row['date_day'] != $lastDate){
                    $lastDate = $row['date_day'];
                    $countDay++;
                }
            }
        }
    }
}

```

Рисунок 6 – Образец кода поиска расписания

### *Факультеты*

Если со стороны мобильного приложения был передан запрос на получение списка факультетов, передав при этом идентификационный токен, то функция производит в таблице факультетов поиск всех имеющихся факультетов. После успешного поиска клиентскому модулю возвращает список найденных факультетов.

```

public function faculties()
{
    $this->beforeCheck('token');

    if(!$faculties = $this->db->getAll('SELECT id, name FROM brs_fakultets'))
        $this->view(500);

    $this->response = array('faculties' => $faculties);
    $this->view(200);
}

```

Рисунок 7 – Образец кода поиска факультетов

### *Курсы*

Если со стороны мобильного приложения был передан запрос на получение списка курсов, передав при этом токен и идентификатор нужного факультета, то функция отбирает те группы, которые принадлежат



выбранному факультету и на основе найденных групп вычисляет принадлежность их курсов. После обработки данных клиентскому модулю возвращает список уникальных курсов.

```
public function courses()
{
    $this->beforeCheck(array('faculty', 'token'));

    $lastYear = time() > mktime(0,0,0,8,15) ? date("Y") : date("Y") - 1;
    if(!$groups = $this->db->getAll('
        SELECT g.pyear
        FROM brs_groups as g, brs_specs as s
        WHERE g.spec_id = s.id AND s.fakultet_id = ?i',
        $this->request['faculty']))
    {
        $this->view(500);
    }

    foreach($groups as $v){
        $groupYears[] = $v['pyear'];
    }
}
```

Рисунок 8 – Образец кода поиска курсов

### *Группы*

Если со стороны мобильного приложения был передан запрос на получение списка групп, передав при этом токен, идентификатор факультета и номер курса, то функция ищет те группы, которые принадлежат выбранному факультету и курсу. После успешного поиска клиентскому модулю возвращает список найденных групп.

```
public function groups()
{
    $this->beforeCheck(array('faculty', 'course', 'token'));
    $lastYear = time() > mktime(0,0,0,8,15) ? date("Y") : date("Y") - 1;
    $i = 0;

    if(!$groups = $this->db->getAll('
        SELECT g.id, g.num, s.name, g.num, g.pyear
        FROM brs_groups as g
        INNER JOIN brs_specs as s
        ON s.fakultet_id = ?i AND s.id = g.spec_id',
        $this->request['faculty']))
    {
        $this->view(500);
    }

    foreach($groups as $g){
        if($this->request['course'] == ($lastYear - $g['pyear'] + 1)){
            $groups[] = $g;
        }
    }
}
```

Рисунок 9 – Образец кода поиска групп

### *Студенты*

Если со стороны клиентского модуля был передан запрос на получение списка студентов, передав при этом токен и идентификатор группы, то



функция производит поиск студентов по заданной группе. После успешного поиска клиентскому модулю возвращает список найденных студентов.

```
public function students()
{
    $this->beforeCheck(array('group', 'token'));

    if(!$students = $this->db->getAll('SELECT name, sname, pname, user_id FROM ...'))
        $this->view(500);

    $this->response = array('students' => $students);
    $this->view(200);
}
```

Рисунок 10 – Образец кода поиска студентов

### *Сообщения*

Если со стороны клиентского модуля был передан запрос на отправку сообщения, передав при этом токен, список групп, список пользователей, тему сообщения и текст сообщения, то функция с помощью технологии Google Cloud Messaging рассылает данное сообщение указанным групп и пользователям. Точнее, функция перебирает таблицу устройств и сверяет с перечисленных выше групп и пользователей. Если найденный идентификатор устройства принадлежит указанной группе или пользователю, то отправляет им сообщения.

```
public function message()
{
    $this->beforeCheck(array('groups', 'users', 'subject', 'message'));

    $this->GCM = new \GCM\Main;
    $this->GCM->send(
        'message', null,
        $this->request['subject'],
        $this->request['message'],
        $this->userID,
        $this->request['groups'],
        $this->request['users']
    );

    if($this->GCM->status)

```

Рисунок 11 – Образец кода отправки сообщения

### *Поиск*

Если со стороны клиентского модуля был передан запрос на поиск данных, передав при этом токен и фразу поиска, то функция производит поиск всех упоминаний о студентах и групп по заданной фразе. Студентов производит поиск по фамилии, имени и отчеству, а групп ищет по их номерам или названию специальности. После успешного поиска клиентскому модулю возвращает список найденных запросов в надлежащем виде.

```

public function search()
{
    $this->beforeCheck(array('find', 'token'));
    $find = trim(preg_replace('/[^А-Яа-яЁё0-9]/u', '', $this->request['find']));

    if((mb_strlen($find, 'UTF-8') < 2) || (mb_strlen($find, 'UTF-8') > 32))
        $this->view(406);

    $lastYear = time() > mktime(0,0,0,8,15) ? date("Y") : date("Y") - 1;
    $finded = array();
    $groups = array();
    $students = array();

    if ctype_digit(strval($find))
    {
        $findedGroups = $this->db->getAll('
            SELECT g.id, g.pyear, g.num, s.name
            FROM brs_groups as g
            INNER JOIN brs_specs as s
            ON g.spec_id = s.id
        ');
    }
}

```

Рисунок 12 – Образец кода поиска данных

### *Балльно-рейтинговая система*

Если со стороны клиентского модуля был передан запрос на поиск баллов, передав при этом идентификационный токен, то функция производит поиск всех баллов найденных дисциплин текущего учебного курса для текущего пользователя. Пользователя вычисляют с помощью токена, т.к. у каждого авторизованного пользователя может быть только один уникальный токен, которого можно идентифицировать «личность» данного пользователя. После успешного поиска клиентскому модулю возвращает список найденных дисциплин и учебных планов, соответственно и баллы тоже, к каждому пройденному учебному плану.

```

public function brs()
{
    $this->beforeCheck('token');
    $year = time() > mktime(0,0,0,8,15) ? date("Y") : date("Y") - 1;

    if(!$student = $this->db->getRow('SELECT id FROM brs_students WHERE user_id = ?i', $this->userID))
        $this->view(500);

    $result = $this->db->getAll('
        SELECT
            brs_students.id as student,
            brs_uplan_labs.name as lname,
            brs_uplan_labs.id as lid,
            brs_uplan_subjects.id as usid,
            min_mark,
            max_mark,
    ');
}

```

Рисунок 13 – Образец кода поиска баллов студента

На этом заканчивается знакомство с функциями API-класса. Далее рассмотрим принцип составления расписания в виде Excel-шаблона. Само

расписание студентов составляется в программном обеспечении Microsoft Excel, где в шаблоне указывают все необходимые данные в надлежащем виде согласно своей структуре и расположения ячеек.

Имя учебного заведения		44.03.05 ПО	09.03.03 ПИ	09.03.02 ИСиТ	44.03.05 ПО	09.03.02 ИСиТ	
Направленность		ИиМ	ПИиЭ	ИСиТиП	ИиМ	ИСиТиП	
Номер группы		<b>1251</b>	<b>1551</b>	<b>1751</b>	<b>1241</b>	<b>1741</b>	
Дисциплина	Дата	1 курс			2 курс		
п о н е д е л ь н и к	28 марта	1	Математика (математический анализ) (лк), Эйрих Н.В., ауд. 306				
		2	Математика (математический анализ) (пр), Эйрих Н.В., ауд. 304				
		3	Физическая культура и спорт (группа 4 М), Изосимов А.Н., спортзал				Численные методы (лк), Эйрих Н.В., ауд. 303,
		4	Физическая культура и спорт (группа 7 Ж), Кирьянцева Л.П., спортзал		Архитектура компьютера (лк), Платовая Л.А., ауд. 306	Иностранный язык (лк), Хозяйкина Г.С., ауд. 303	
		5	Операционные системы, сети и интернет - технологии (пр), Лучанинов Д.В., ауд. 408			Архитектура компьютера (пр), Платовая Л.А., ауд. 306	Численные методы (лб), Эйрих Н.В., ауд. 303
		6	Программирование (лк), Лучанинов Д.В., ауд. 408				
в		1					
		2	Экономическая теория (лк), Голик А.С., ауд. 303	Иностранный язык (пр), Каллаур В.С., ауд. 416	Философия (лк), Цыцарев А.А., ауд. 216		

Рисунок 14 – Образец расписания Excel-шаблона

После того, как составитель расписания создал шаблон и заполнил все необходимые данные, далее нужно зайти в личный кабинет официального сайта университета, используя свой логин и пароль.

## Профиль пользователя

Вход
[Забыли пароль?](#)

**Имя пользователя \***

**Пароль \***

Рисунок 15 – Авторизация для входа в личный кабинет

Затем заходит во вкладку «Расписание» в своем личном кабинете, и заливает данный шаблон расписания на сайт данного университета.

## Расписание занятий



Добавить   Редактировать   Удалить

Выбрать файл...   38.-16.05-21-05--2016.xls

Загрузить

Рисунок 16 – Расписание в личном кабинете

После успешной загрузки расписания, серверный модуль на этом этапе автоматически «цепляет» загруженное расписание, и производит по ней распознавание ячеек с помощью парсера, который в свою очередь обрабатывает их и заносит в базу данных.

Тем самым, данная разработанная информационная система позволяет существенно снизить время и затраты сотрудников университета, позволяя информировать студентов с оперативной отдачей и минимальными затратами по времени. Результатом этой системы заключается в том, чтобы держать студентов в курсе всех происходящих событий в режиме реального времени в виде уведомлений, приходящих на мобильное приложение.

### Библиографический список

1. Справочное руководство по MySQL [Электронный ресурс]. URL: <http://www.mysql.ru/docs/man> (дата обращения: 28.08.2016).
2. Приамурский государственный университет имени Шолом-Алейхема. Расписание занятий [Электронный ресурс]. URL: <http://pgusa.ru/ru/studentu/raspisanie-zanyatiy> (дата обращения: 28.08.2016).
3. РНР. Справочник функций [Электронный ресурс]. URL: <http://php.net/manual/ru/funcref.php> (дата обращения: 28.08.2016).
4. Сухов М.А. Проектирование и разработка мобильного приложения «Расписание МГТУ» на платформе операционной системе Android // ФГБОУ ВПО «Магниторский государственный технический университет им. Г.И. Носова». 2015. №1. С. 26-31.
5. Митлошук А.В., Князькова Т.В. Мобильное приложение для просмотра расписания студентов // Молодежный научный форум: технические и математические науки. 2016. № 2 (31). С. 28-32.
6. Бородин А.С., Бурлуцкий В.В. Мобильное расписание для высших учебных заведений // Молодежь и современные информационные технологии. 2013. С. 141-143.
7. Труфанов И.А. Разработка мобильного приложения «Расписание ФинУниверситета» // Развитие Российской экономики: Проблемы и перспективы. 2014. С. 809-820.