**Имена:**  Гергана Василева Атанасова **фн:**  *61488***Начална година:**  **2012 Програма:** бакалавър, (СИ) **Курс: 2**   
**Тема: Метеорологичен център  
Дата: 05.07.2013 Предмет: WWW технологии имейл: gvasilevaa@gmail.com**

**преподавател:** доц. д-р Милен Петров, **асистент:** Кулев/Стоицев

**Файл: wwwTech2013\_fn61488\_FinalProject.zip**

**ТЕМА: Разработване на Метеорологичен център – регистрация, въвеждане на данни и визуализация**

1. **Условие**

**Метеорологичен център**

Проектът служи за въвеждане и преглеждане на метеорологични данни и прогнози.

Има три категории потребители: админ, служител, клиент (потребителско име, парола,име, имейл).

***Администраторите:***

● въвеждат/редактират/трият служители

● въвеждат/редактират/трият локации

● могат да командироват служители по локации (командировките имат начална дата,крайна дата, служител и локация). Могат да редактират и анулират командировки.

● могат да въвеждат/редактират/трият данни за всички локациии (вж. подолу)

***Служителите:***

● Могат да въвеждат данни във формат: локация, температура, влажност, време с

точност до час. Могат и да редактират и трият тези данни.

● Служителите могат да въвеждат данни само за локации, в които са командировани и то само за до +7 дни от днешна дата, ако днешната дата е периода на командировката.

***Потребителите*** (няма нужда от логин)

● могат да разглеждат списъка с локациите, да сортират и да търсят по име локация

● да преглеждат прогнозата за дадена локация

Служителите трябва да могат да въвеждат данни. Данните са във формат: място,

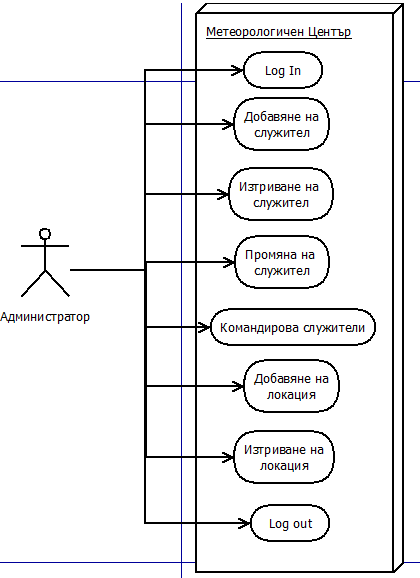
темепература, влажност, време с точност до час. Служителите могат и да редактират и трият тези данни.

**2. Въведение**

Системата ще представлява метеорологичен център, където оторизирани администратори ще въвеждат/редактират/трият служители/локации; служителите ще могат да въвеждат данни за времето за определени локации, където са командировани локации. Системата е предназначена и за обикновени потребители, които ще я достъпват чрез браузър и ще имат възможност да разглеждат прогнозата за деня/24 часа/ 5 дни .

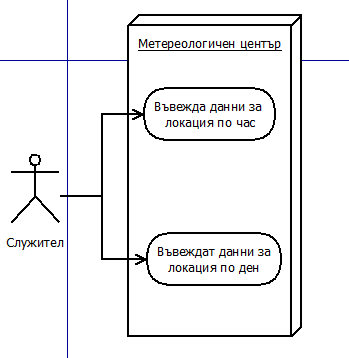
**3. Теория и проектиране**

**1. Admin\_Use\_Case figure**

****

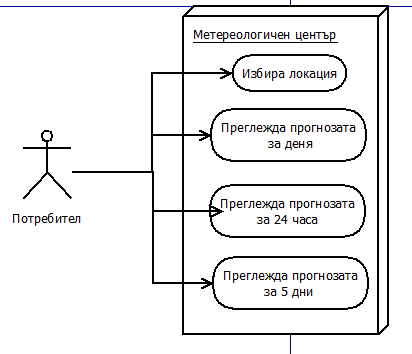
На фигурата са показани функционалностите, който системата поддържа за администратора. За да може администратора да добавя/променя/изтрива служители/локация трябва да е логнат преварително. Аминистраторър може и да командирова служители на дадени локации.

**2. Employee\_Use\_Case figure**



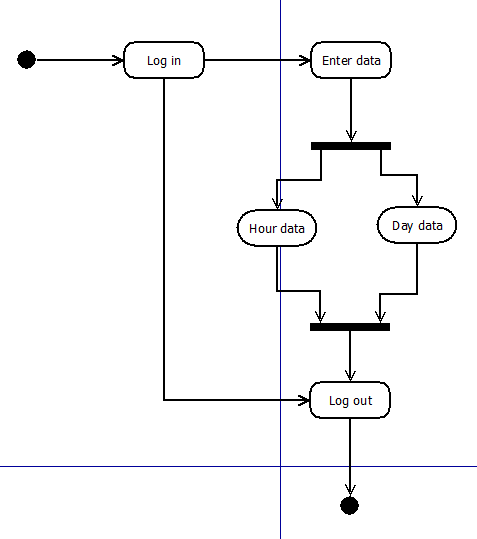
На фигурата са показани функционалностите, който системата поддържа за служителите. За да може служителя да добавя данни за локация, където е командирован трябва да е логнат преварително.

**3. User\_Use\_Case figure**

****

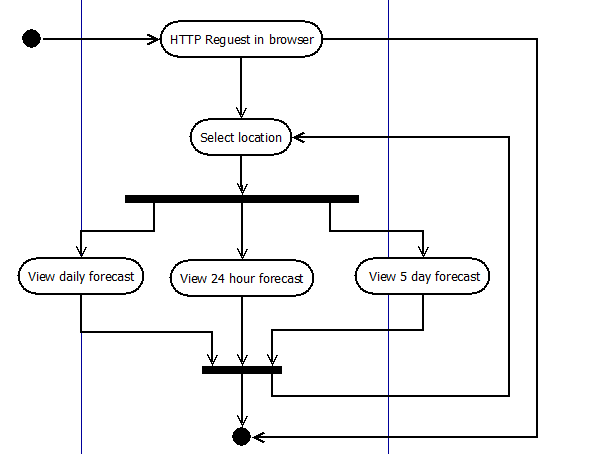
На фигурата са показани функционалностите, който системата поддържа за потребителите. На потребителя не му е нужен профил, за да може да разглежда прогнозите.

4. Activity diagram for employee



Диаграмата описва последователността от стъпки, който служител може да предприеме при употреба на системата.

5. Activity diagram for user



Диаграмата описва последователността от стъпки, който потребител може да предприеме при употреба на системата.

**4. Използвани технологии**

HTML5, CSS3, jQuery, AJAX, SQL, PHP;

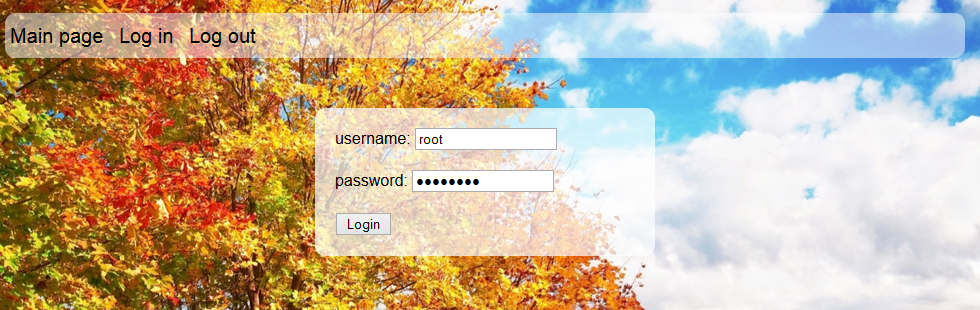
**5. Инсталация и настройки**

Не е нужна инсталация. Обикновеният потребител, може да разгледа системата чрез уеб-браузър дори и без да е нужно да се регистрира в системата.

**6. Кратко ръководство на потребителя**

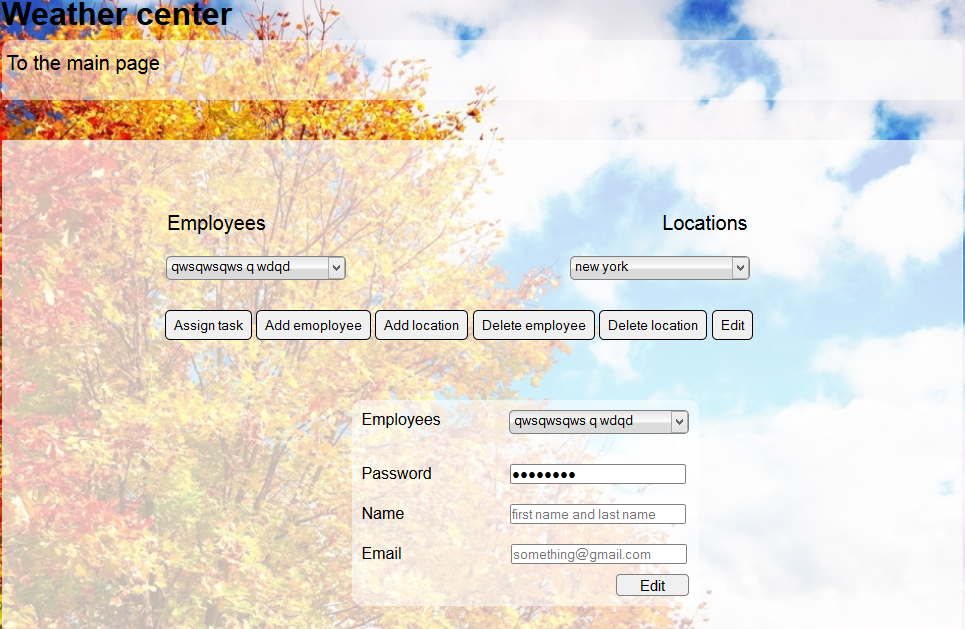
Системата няма нужда от логин за обикновения потребител. Единствено администраторите и служителите могат да се регистрират, да въвеждат, да изтриват данни и други.

Логин формата:



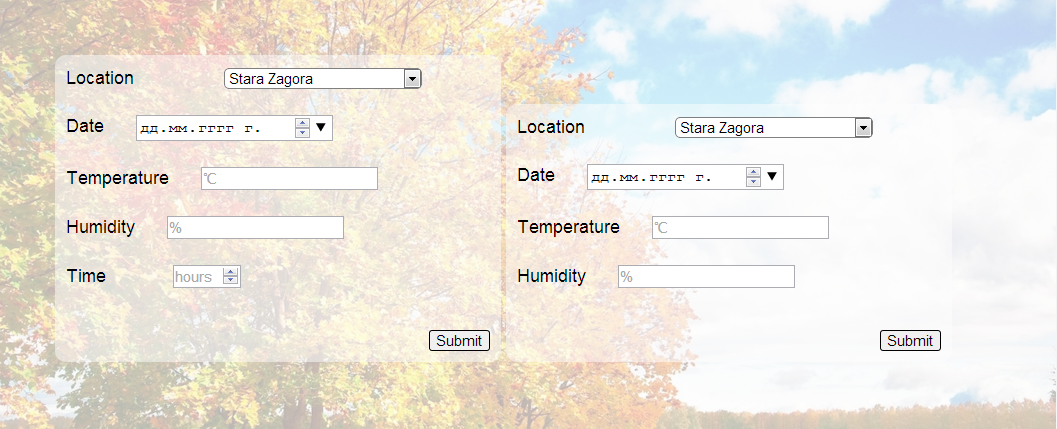
В тази форма мгат да се логват както администратора, така и въведените от него служители. След това те биват пренасочени съответно към страниците от проложението, които са предназначени за тях.

Страница на администратора:



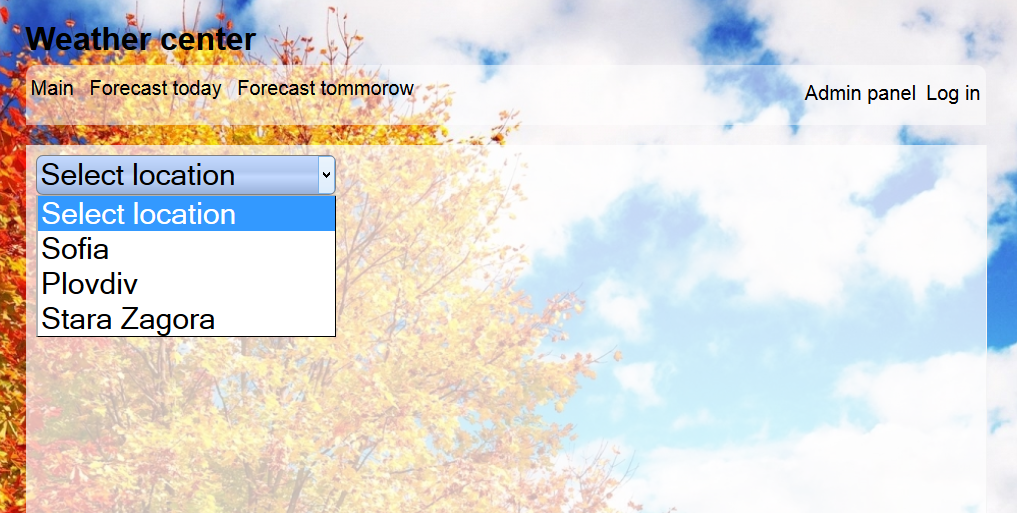
Администраторската форма – може да се въвеждат, изтриват и поправят локации и служители, както и да се назначават задачи на служителите. За целта админитратора попълва полетатана показалите се форми – потребителско име, име, парола, имейл.

Страница на служителите:

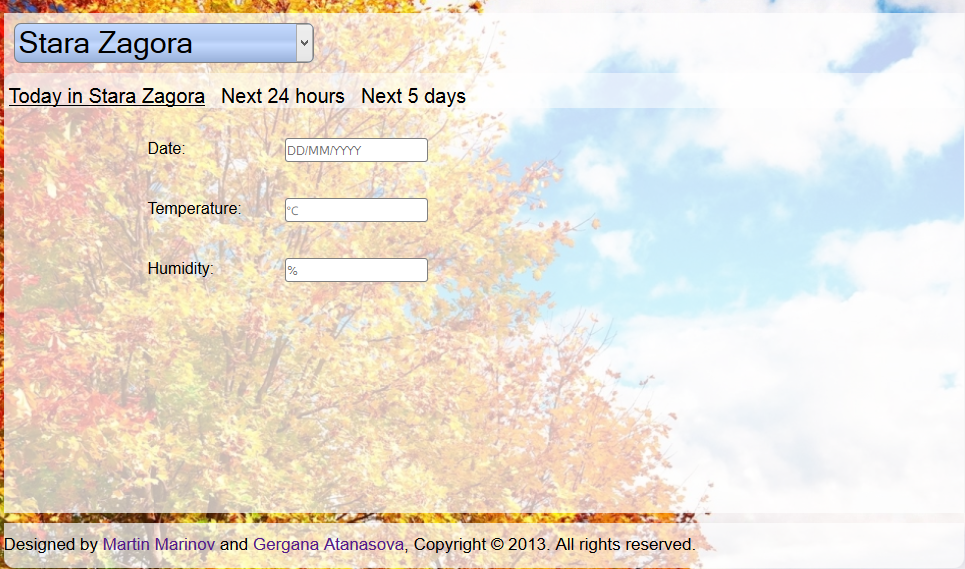


Служителската форма – служителите могат да избират локация измежди тези, за който са командировани от админисратора и да въвеждат съответните дани за нея - температура, влажност и т.н., попълвайки съответните полета.

Началната страница:



Началната страница е мястото където могат да се виждат данни за съответните дестинации от потребителите (като предварително са избрали такава). Системата поддържа дневна прогноза, 24-часова и 5 дневна.След като се избере дестинация, има 3 менюта, които в зависимост кое изберем, ще ни се визуализира форма за визуализиране на данни.



**7. Примерни данни**

--

-- Структура на таблица `employee`

--

CREATE TABLE IF NOT EXISTS `employee` (

`id\_emp` int(11) NOT NULL AUTO\_INCREMENT,

`user\_emp\_name` varchar(25) COLLATE utf8\_unicode\_ci NOT NULL,

`employee\_name` varchar(45) COLLATE utf8\_unicode\_ci NOT NULL,

`employee\_password` varchar(32) COLLATE utf8\_unicode\_ci NOT NULL,

`employee\_email` varchar(25) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`id\_emp`),

KEY `id\_emp` (`id\_emp`),

KEY `id\_emp\_2` (`id\_emp`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=18 ;

--

-- Ссхема на данните от таблица `employee`

--

INSERT INTO `employee` (`id\_emp`, `user\_emp\_name`, `employee\_name`, `employee\_password`, `employee\_email`) VALUES

(6, 'Siqna', 'sisi', '47bce5c74f589f4867dbd57e9ca9f808', 's@gmail.com'),

(15, 'user1', 'geri', '24c9e15e52afc47c225b757e7bee1f9d', 'geri\_v\_a@abv.bg'),

(16, 'asd', 'asdasd', '0aa1ea9a5a04b78d4581dd6d17742627', 'asdsad@abv.bg'),

(17, 'gvasilevaa', 'gergana', 'f03aa693b10635812712f80730ccdaf3', 'geri@gmail.com');

--

-- Структура на таблица `weather\_data`

--

CREATE TABLE IF NOT EXISTS `weather\_data` (

`id\_weather\_data` int(11) NOT NULL AUTO\_INCREMENT,

`weather\_date` date NOT NULL,

`weather\_temp` int(11) NOT NULL,

`weather\_humidity` int(11) NOT NULL,

`weather\_hour` int(11) DEFAULT NULL,

`location\_id\_location` int(11) NOT NULL,

PRIMARY KEY (`id\_weather\_data`,`location\_id\_location`),

KEY `fk\_weather\_data\_location1\_idx` (`location\_id\_location`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci AUTO\_INCREMENT=33 ;

--

-- Ссхема на данните от таблица `weather\_data`

--

INSERT INTO `weather\_data` (`id\_weather\_data`, `weather\_date`, `weather\_temp`, `weather\_humidity`, `weather\_hour`, `location\_id\_location`) VALUES

(1, '2013-06-26', 23, 50, 4, 6),

(2, '2013-06-26', 23, 50, 8, 6),

(3, '2012-03-16', 12, 55, 11, 2),

(4, '2012-03-16', 25, 25, 4, 2),

(11, '2013-06-22', 12, 20, 4, 2),

(13, '2013-03-05', 25, 23, 12, 2),

(14, '2013-07-01', 12, 65, 2, 2),

(15, '2013-07-01', 14, 55, 4, 2),

(16, '2013-07-01', 16, 50, 6, 2),

(17, '2013-07-01', 17, 50, 8, 2),

(18, '2013-07-01', 19, 45, 10, 2),

(19, '2013-07-01', 20, 45, 12, 2),

(20, '2013-07-01', 22, 40, 14, 2),

(21, '2013-07-05', 16, 81, 2, 2),

(22, '2013-07-05', 18, 82, 4, 2),

(23, '2013-07-05', 19, 82, 6, 2),

(24, '2013-07-05', 17, 82, 8, 2),

(25, '2013-07-05', 19, 75, 10, 2),

(26, '2013-07-05', 21, 75, 12, 2),

(27, '2013-07-05', 25, 75, 14, 2),

(28, '2013-07-05', 26, 65, 16, 2),

(29, '2013-07-05', 27, 60, 18, 2),

(30, '2013-07-05', 26, 62, 20, 2),

(31, '2013-07-05', 22, 62, 22, 2),

(32, '2013-07-05', 18, 68, 24, 2);

**8. Описание на програмния код**

Основата на front-end-а е реализирана с HTML5. За стилизиране е използвано CSS3, а за ефектите - jQuery и AJAX. Тъй като системата има опция за регистрация и логин, тя поддържа база от данни, реализирана с помощта на MySQL Workbench и phpmyadmin.

Пример за Ajax заявка за управление на сесии:

$(document).ready(function(){

$.ajax({

type: "POST",

url: "session.php",

success: function(data){

console.log("session invoked");

console.log(data);

if(data==true){

$('#warning').hide('fast');

$('#content').show('fast');

$('#login').hide('fast');

$('#logout').show('fast');

$.ajax({

type: "POST",

url: "load\_locations.php",

dataType: 'json',

success: function(data){

//console.log(data.name[1]);

var i=0;

$.each(data.name, function() {

$(".location").append(new Option(data.name[i], data.id[i]+"|"+data.name[i]));

++i;

});

}

});

}else{

$('#warning').text("You first have to log in!");

$('#content').hide('fast');

$('#login').show('fast');

$('#logout').hide('fast');

}

}

});

});

**9. Приноси на студента, ограничения и възможности за бъдещо разширение**

Моята част от проекта беше предимно да реализирам back-end-a на системата. Използвах jQuery, Ajax, JSON, PHP и MySQL. Изградих базата данни и осигурих коректната й употреба и попълване и употреба.

Системата има потенциал за развитие като се добавят и други данни, които да се въвеждат за прогнозата от служителите – скорост на вятъра, лунни фази и т.н. Могат да се добавят и картинки за визуализиране на прогнозите.

**10. Какво научих**

Да използвам jQuery, Ajax, JSON, MySQLi и PHP .

**11. Използвани източници**

[http://www.codeproject.com/Articles/31271/Ajax-Tutorial-for-Beginners-with-XML-JSON-Part-2#\_rating](http://www.codeproject.com/Articles/31271/Ajax-Tutorial-for-Beginners-with-XML-JSON-Part-2%23_rating%20)

<http://www.codeproject.com/Articles/31155/Ajax-Tutorial-for-Beginners-Part-1%20>

[http://www.colorzilla.com/gradient-editor/](http://www.colorzilla.com/gradient-editor/%20)

<http://php.net/manual/en/index.php>

<http://php.net/manual/en/book.mysqli.php>

и други...