Софийски университет „Св.Климент Охридски”  
  
Факултет по математика и информатика

## Индивидуален проект по АПИС

05.06.2017г. Изготвил: Aлександър Танков  
 Специалност: ИС  
 Ф.Н.: 71492

Съдържание

Интервюта стр.3

Подобни системи стр.3

CD модел стр.5

Пълно описание на потребителските случаи стр.6

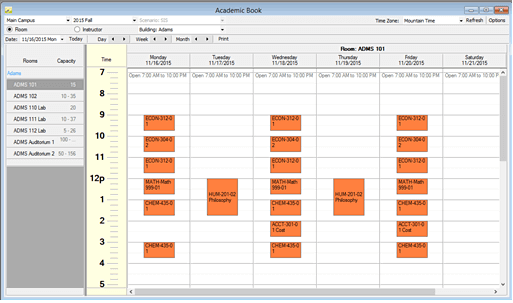
Домейн модел стр.9

# Интервю

На 2016-04-18 участвах в проведеното интервю с жената, занимаваща се с разпределянето на стаите във ФМИ, за да си изясним изискванията към системата. Това много ни помогна, защото получихме информация как точно се случва разпределянето на стаите в момента и какви външни системи се използват.

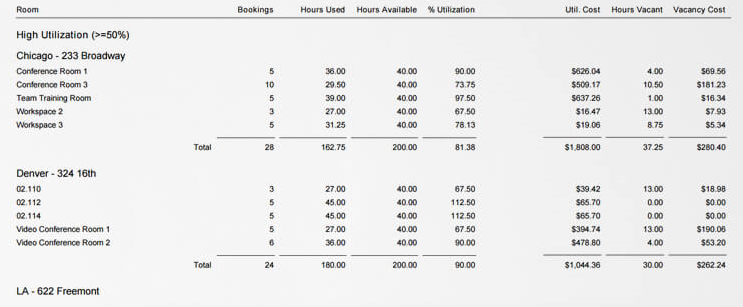
# Подобни системи

Разглежданата от мен иформационна система е под името EMS ( https://www.emssoftware.com ). Това е платформа за организиране на зали, мероприятия и срещи. Тя е създадена да помага на хората по-лесно и по-бързо да разпределят занятия по различни стаи, да откриват подходящи конферентни зали за специални мероприятия и бързо да откриват свободни стаи за малки срещи.



Една от основните функционалности на този продукт е подреждането на стаите за занятията на един клас/поток/група. То може да се прави спрямо големината, оборудването, разположението и заетоста на стаята. Това спестява много време на хората, които се занимават с това в университетите и училищата.

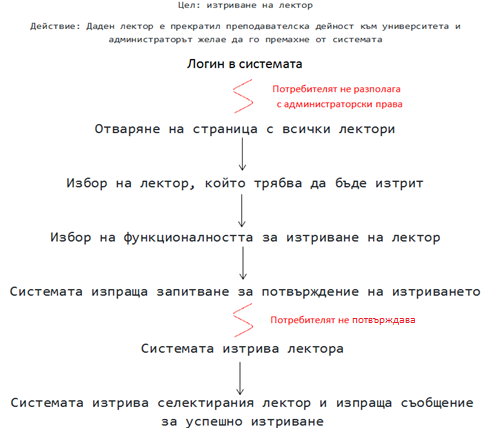
Всички занятия са изобразени в календар, до който всеки един потребител има достъп. Така по всяко време всеки може да провери кога и къде трябва да бъде за даденото занятие. Нещо, което го считам за пропуск (или просто в пробната версия не съществува) е търсенето на занятие в календара. Също така смятам, че би било доста полезно, когато се създава някакво занятие хората участващи в него да бъдат информирани по email.

Информационната система предоставя още функционалности като една от тях е предоставянето на статистики за използваемостта на стаите.

Като изисквания към нашия проект могат да се вземат предвид предлаганите функционалности от тази информационна система както да се добавят и други по желание на клиента (в зависимост с това за какво точно ще се използва системата и какви са нейните цели).

# CD модел

След събирането на информацията за системата, участвах в определянето на основните функционалности и различните потребителски роли, които ще се поддържат от системата.

На базата на тази инфорамция създадох модел на последователностите при изтриване на лектор:

Този модел на последователностите представя действията на администратора по време на изтриване на лектор. Той включва логване в системата, избиране на лектора за премахване и потвърждение на администратора, че иска да го премахне. След това системата премахва лектора и изпраща съобщение за успешно изтриване.

# Пълно описание на потребителските случаи

Участвах в разработването на шаблона, по който са писани всичките потребителски случаи и в приоритизирането им.

От идентифицираните потребителски случаи, описах следните три в пълен (fully dressed) формат:

## use case 1: Поискване на резервация на зала от лектор

Всеки лектор в системата има право да поиска да запази зала. Когато той избере час, периодичност и зала, трябва да попълни заглавие на заниманието, което ще се провежда, курс и специалност. В резултат системата изпраща до администратора поискването на лектора и той преценява дали да запази тази зала за него или не.

ниво: Потребителска цел.

актьор: Лектор

заинтересовани страни:

Лектор: Иска да запази специфична зала с удобен час и дата

действие:

На потребителят му се е наложило да направи допълнително упражнение и иска да запази зала

предпоставки:

Лекторът трябва да е влязъл в системата като лектор

последствия:

Инициаторът е поискал да запазил успешно залата и се чака отговор от администратора.

главен успешен сценарий:

01. Потребителят отива на календар страницата

02. Системата предоставя седмичен календар с попълнени резервираните часове

03. Потребителят избира свободен час.

04. Системата предоставя форма за резервация на зала с попълнени избраните дата, час и лектор.

05. Потребителят избира периодичност, специалност, курс и предмет от предоставените списъци.

06. Потребителят потвърждава резервацията.

07. Системата изпраща поискването на лектора за резервация към администратора.

разширения:

03a. Потребителя избира зает час

01. Системата информира потребителя, че няма как да запази вече зает час.

02. Продължаваме със стъпка 03 от главния сценарий.

06а. Потребителя избира специалност, която е заета в указания час.

01. Системата информира потребителя, че специалноста е заета.

02. Продължаваме от стъпка 03 от главния сценарий.

07а. В продължение на дълъг период от време потребителят не потвърждава резервацията.

01. Системата премахва резервацията на стаята за този час

02. Продължаваме със стъпка 02 от главния сценарий.

специални изисквания:

- системата запазва залата за по-малко от 0.8 секунди.

честота на използване:

- много

други: N/A

## use case 2: Редактиране на резервация от лектор

Лекторът, който е запазил зала, има възможност да променя детайлите по тази резервация. Това се прави с цел ако трябва да се променят часа или залата за занятието. В резултат на промяната има проверка дали залата е свободна за указаната дата. Ако е свободна се променя резервацията, ако не е се съобщава на потребителя.

ниво: Потребителска цел

актьор: лектор

заинтересовани страни:

- Лектор: Иска да промени данните на резервация, която сам е направил

действие:

Лекторът е осъзнал, че дадена резервация е грешна и иска да я редактира

предпоставки:

Лекторът трябва да е влязъл в системата и да има поне една резервация

последствия:

Лекторът е променил успешно данните на резервацията.

главен успешен сценарий:

01. Потребителят отива на календар страницата

02. Системата предоставя седмичен календар на лектора с попълнени резервираните часове

03. Потребителят избира собствена резервация

04. Системата показва детайлите на резервацията

05. Потребителят избира функционалноста за промяна на резервацията

06. Системата предоставя възможност на потребителя да промени всяка една от данните на резервацията с изключение на датата и часа.

07. Потребителят променя данните, които желае

08. Системата проверява дали указаните данни са правилни

09. Системата показва съобщение за успешно променена резервация

разширения:

08а. Валидацията на въведените данни не е успешна

01. Системата предоставя възможност на потребителя да промени всяка една от данните на резервацията.

02. Системата сигнализира кои полета са попълнени грешно

03. Продължаваме със стъпка 07 от главният сценарий

специални изисквания:

- системата променя резервацията за по-малко от 1 секунда.

честота на използване:

- по-малко

други:

N/A

## use case 3: Добавяне на лектор

Администраторът има права и възможност да добавя нови лектори към факултета. Ако нов колега бъде назначен на работа, да може администраторът лесно и удобно да му създаде акаунт с който лектора да има достъп до системата.

ниво: Потребителска цел

актьор: администратор

заинтересовани страни:

- Администратор: Иска да добави нов лектор

действие:

Назначили са нов лектор и администратора иска да го добави в системата, за да може да я ползва

предпоставки:

Потребителят е влязъл в системата като администратор

последствия:

Добавен е нов лектор успешно и е показана страницата с всички лектори

главен успешен сценарий:

01. Потребителят отива на добавяне на нов лектор

02. Системата предоставя форма за добавяне на лектор.

03. Потребителят попълва данните за новия лектор (име, имейл, телефон, предмети, титли)

04. Системата валидира попълнените данни

05. Системата изпраща съобщение за успешно добавяне на лектор

06. Системата показва страница с всички лектори.

разширения:

04а Валидацията на данните е неуспешна

01. Системата сигнализира кои полета са попълнени грешно

02. Системата предоставя възможност на потребителя да промени всяка една от данните на лектора

03. Продължаваме със стъпка 02 от главният сценарий

специални изисквания:

Системата добавя нов лектор за около 1 секунда

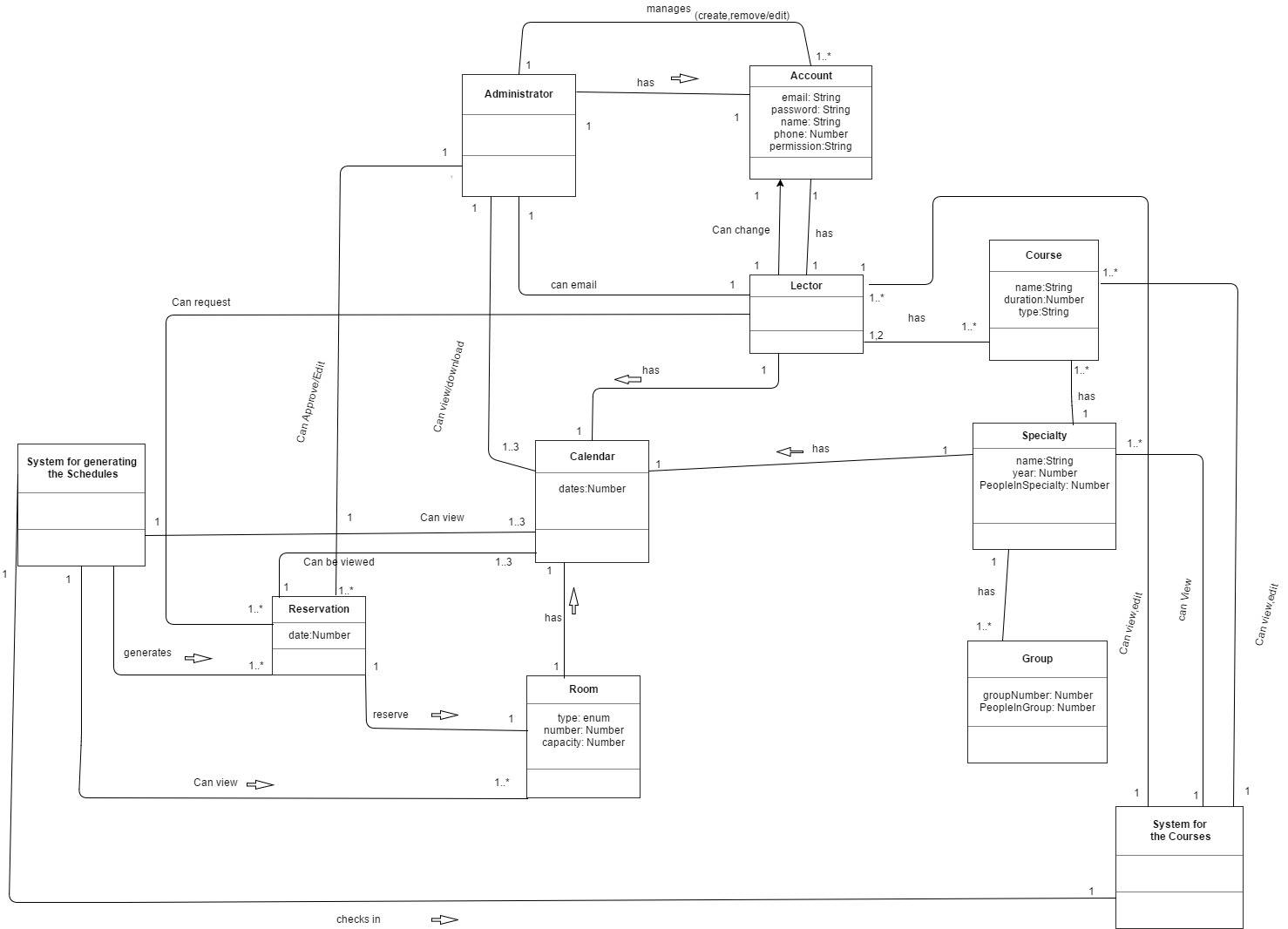
честота на използване:

- по-малко

други:

N/A

# Домейн модел



На базата на участието ми в определянето на домейн модела, създадох диаграма за добавянето на лектор в системата.

За да добави лектор, администраторът трябва да мине през няколко етапа. Той трябва да избере функционалността за добавяне на лектор. След това попълва неговите данни и го добавя.

