

Имена: Ния Митева фн: 62043
Начална година: 2016 Програма: бакалавър, (СИ) Курс: 4
Тема: Уеб инструмент за управление на scrum процеси
Дата: 2019-06-10 Предмет: wwwTech2018_19_12ed_SI_summer
имейл: mitevania@gmail.com

преподавател: доц. д-р Милен Петров

ТЕМА: Уеб инструмент за управление на scrum процеси

1. Условие
2. Въведение
3. Теория
4. Използвани технологии
5. Инсталация и настройки
6. Кратко ръководство на потребителя
7. Описание на програмния код
8. Приноси на студента, ограничения и възможности за бъдещо разширение
9. Какво научих
10. Използвани източници

Предал (подпис):

/фн, имена, спец., група/

Приел (подпис):

/доц. Милен Петров/

Scrum.it

1. Условие

Избраната задача за изпълнение бе поставена по следния начин:

Да се създаде уеб приложение, което да предоставя на потребителя възможност да управлява scrum процеси.

В системата се създава проект, по който ще се разботи и ще може да се следи работата по него. За всеки проект автоматично се създава backlog „табло“ с всички задачи, които трябва да се свърша по проекта. Потребител добавя задачи към backlog списъка. За съответния проект се добавят и sprint цикли. Потребител избира задачи от backlog списъка, които иска да се свършат през съответния sprint и ги добавя в „таблото“ на sprint-а.

Всяка задача има име, описание, приоритет, тип (bug, feature, story card), потребител (който работи по нея), стойност/сложност, състояние и % завършеност.

Всеки sprint е разделен на няколко „табла“: „To do“, „In development“, „Testing“, „Done“. Това, в кое табло ще се намира дадена задача се определя от нейния статус, който се задава от потребител.

За всеки проект ще има екип от участници в реализацията на проекта

За всяка задача ще може да се определя кой ще работи по нея.

Реализирано, приложението ще дава възможност на потребителите да управляват своите проекти и scrum процеси.

2. Въведение

В една софтуерна компания, например, използването на подобно приложение за управление на scrum процеси, е особено необходимо. С него значително се улеснява работния процес. Самото управление на тези процеси се извършва от Project Manager. Той следи за работа по проекта на всеки от екипите. Задава задачи, определени от изискванията на клиент, определя и кои от тях да се свършат през съответния sprint.

За самия scrum процес са характерни ежедневните екипни срещи. Реализацията на такова прилежание може да намали тези срещи. По време на една такава среща се обсъжда кой върху каква задача работи, тя в какво състояние е, за колко време може да я свърши и т.н. Project manager-а може да проследи тези неща през приложението.

Разработчици като част от екип, се „записват“ за дадена задача от списъка на съответния sprint. В описанието на задачата вижда спецификацията на задачата (какво се очаква от него да бъде свършено).

Когато една задача е със статус “Testing”, се дава знак на QA-а определен за нея, че работата на разработчика е свършена и задачата е готова да бъде тествана преди да бъде включена в проекта и предадена на клиента.

Нещо което също е много специфично за scrum процеса са „таблата с лепкащи се листчета“. Всяко sprint табло се разделя на няколко части: „To do“, “In development”, “Testing”, “Done”. Всяка част съдържа списък от задачи (лепкащите се листчета), които се местят в различните части в зависимост от статусът им. Реализацията на тази функционалност в системата ще олесни още повече работата на екипа, разработващ проект.

3. Теория

За момента в системата няма регистрация и вписване. Реализираните функционалности могат да се ползват от всеки потребител. Към момента потребител ползващ системата може да добавя/редактира/изтрива проекти, да добавя/изтрива/редактира спринт цикли, задачи към проект и участници в реализацията на проект. За бъдещо развитие е предвиден администраторски панел и администраторски функционалности.

4. Използвани технологии

В разработката на системата е целено наличието на трислойна архитектура. Тя е изградена на база на **MVC (Model-View-Controller)** модела.

Презентационният слой е разработен чрез употребата на предимно използваните днес за създаване на интерфейси **HTML, CSS** и **JavaScript**.

Информацията, свързана с регистрационните данни на потребителите, както и данните за всеки проект, задачите към него, екип и т.н. се съхраняват в базата от данни, достъпна чрез **MySQL Workbench 6.3 SE**. Приложението е тествано на локално инсталиран **Apache** сървър.

Бизнес логиката е имплементирана чрез употребата на езика **PHP**. **Model** частта от **MVC** модела осъществява изграждането на основните класове за системата и комуникацията с базата данни. **Controller** частта от **MVC** модела осъществява извличането и изпращането на данни до презентационния слой на системата. **View** частта **MVC** модела съдържа **html** темплейтите за визуализацията на отделните страници и елементи от приложението.

За написване то на самия код е използван **Visual Studio Code**.

5. Инсталация и настройки

Начин за достъп до системата е чрез инсталиране на актуалната версия на **XAMPP**. Чрез него могат да се пуснат сървър и **MySQL** база от данни, без необходимост от инсталация на други програми. Чрез **phpMyAdmin** могат да се създадат и редактират необходимите таблици за базата от данни. Файловете на проекта трябва да са поставени в конкретната папка на **XAMP (htdocs)**. За стартиране на приложението в браузъра се изписва localhost и номера на порта, на който да се извика, ако той е различен от 80.

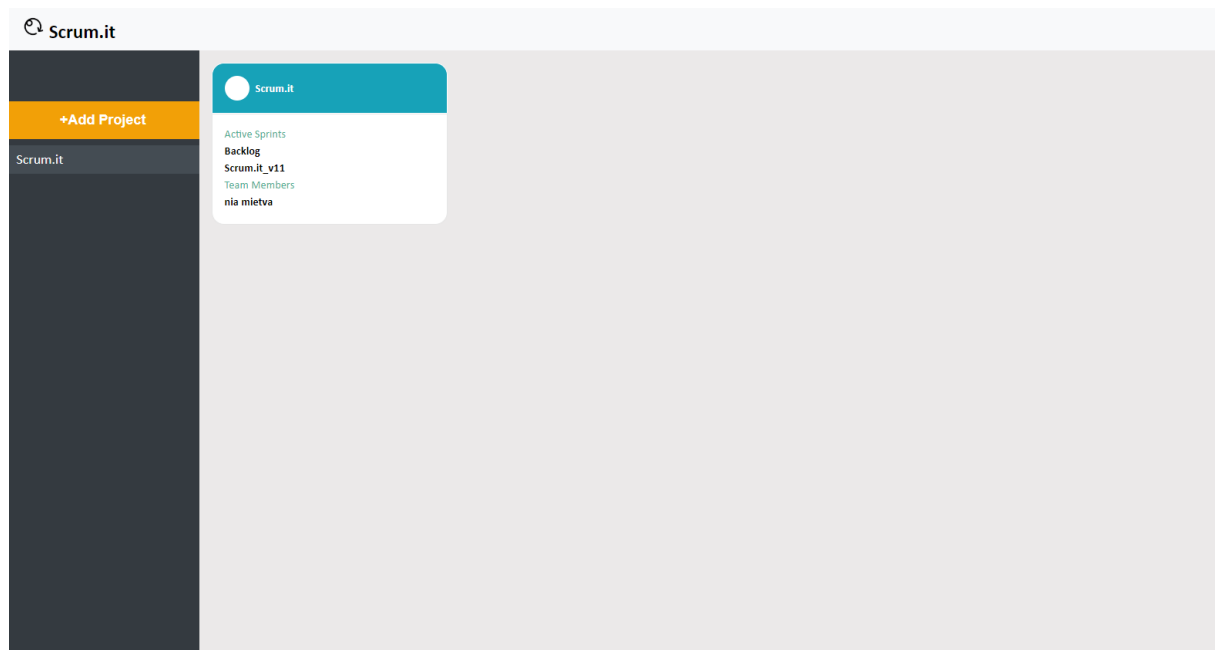
Във папката **src/database** се съхранява **SQL** файла **scrumit.sql**, в които се намират заявките за създаване на необходимите таблици за базата от данни web_project на приложението, както и примерни данни за запълване.

Логиката по осъществяването на връзка с базата от данни е изнесена като код във файла **framework/database/connection.php**. Така при необходимост от корекция, промени ще се правят само в този файл.

Пътищата към отделните файлове и директории във файловата система на проекта са конфигурирани във файла **framework/core/framework.php**, където при необходимост лесно могат да се променят.

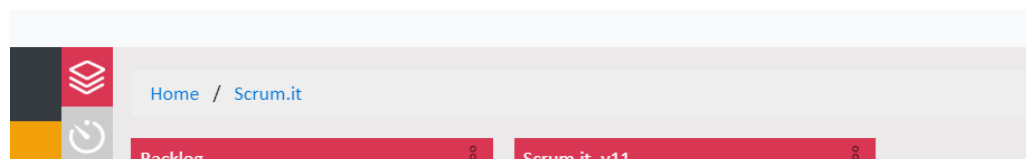
6. Кратко ръководство за потребителя

- Всяка страница съдържа странична навигация с бутон за добавяне на проекти и линкове към отделните проекти
- **Начална страница**
 - съдържа информационна карти за всеки проект, с информация за неговите спринтове и екип



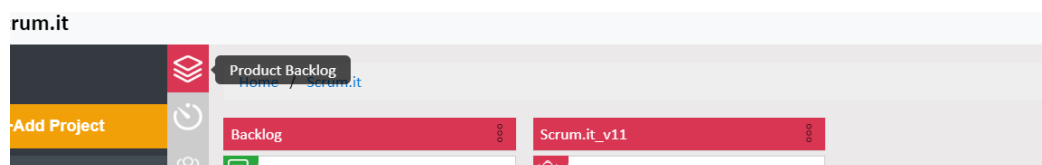
- **Страница на проект**

- път (bread crumbs) , за да знае потребителят къде се намира



- съдържа меню с връзки към различните табове с информация за проекта

- за всеки таб бутон, при преминаване върху него с мишката се появява етикет с името му



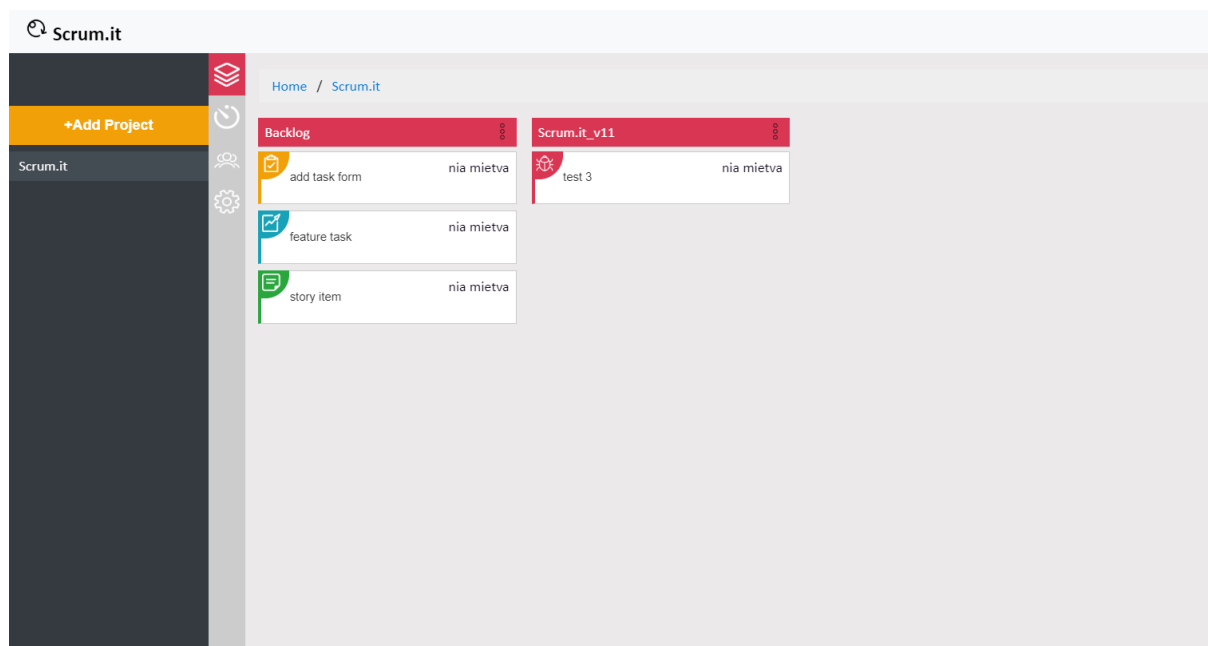
- **таб 1 – backlog**

- = списък от всички задачи – backlog list

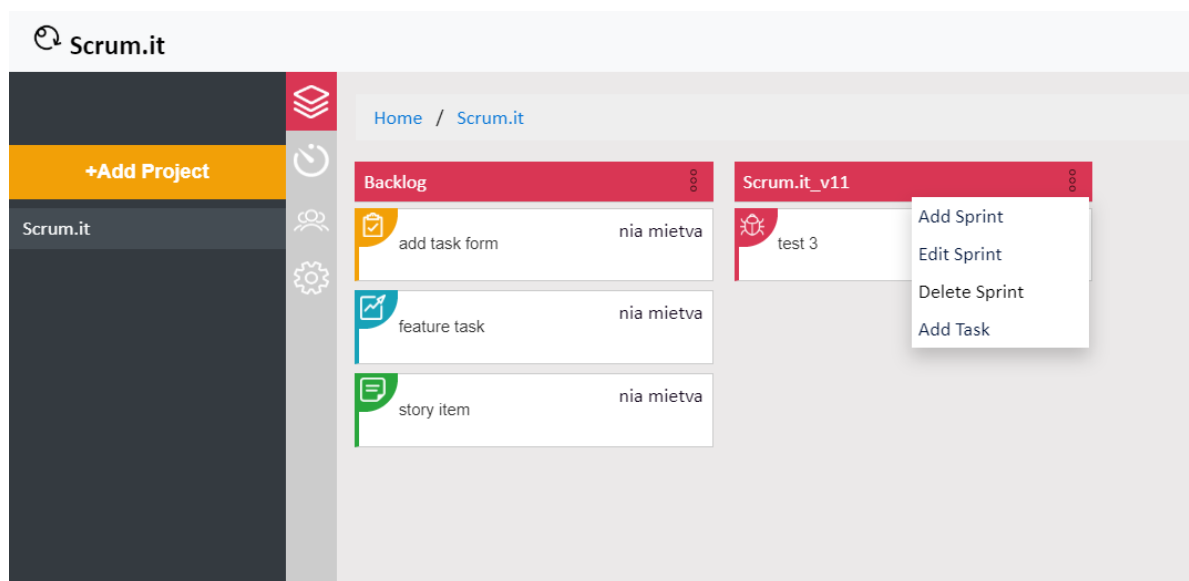
- = списъци със задачи за съответе спринт към проекта

- = горния ляв ъгъл на всяка картичка с информация за задача, показва какъв тип е тя

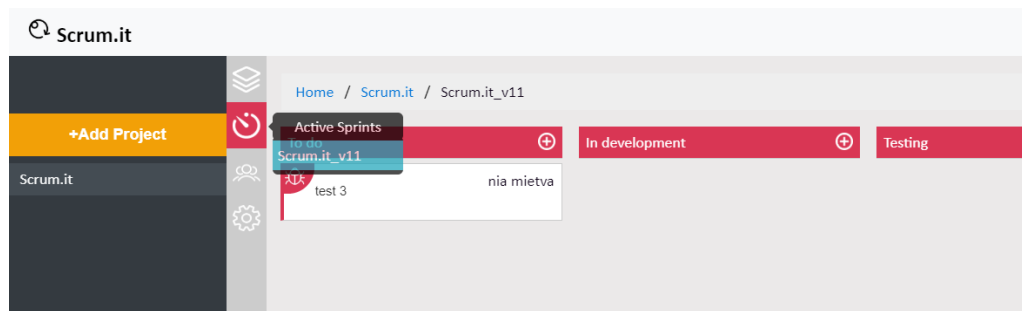
- = всяка картичка на задача показва съответните й име, тип и член на екип



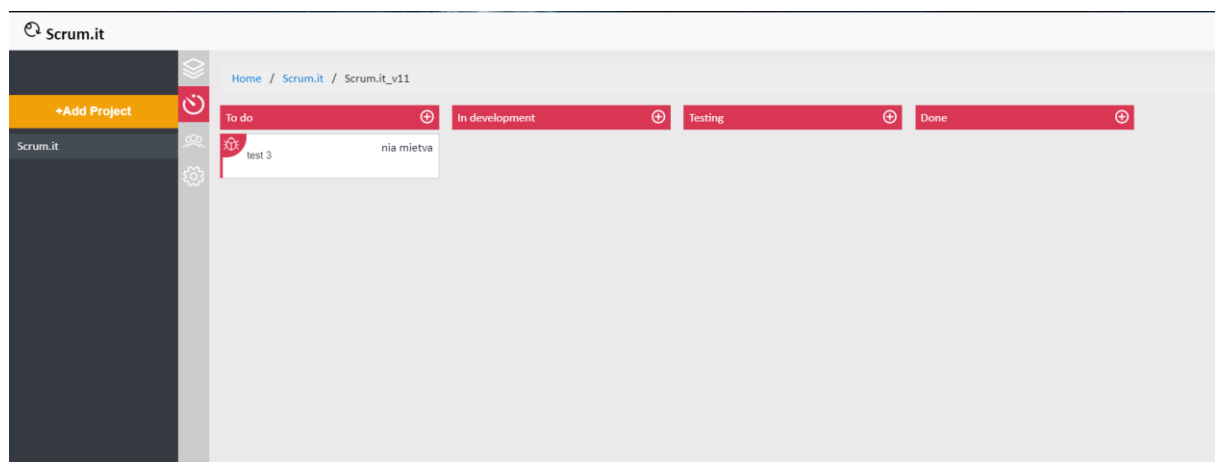
= в дясната страна на заглавието на спринт списъка – падащо меню с конфигурации за спринт



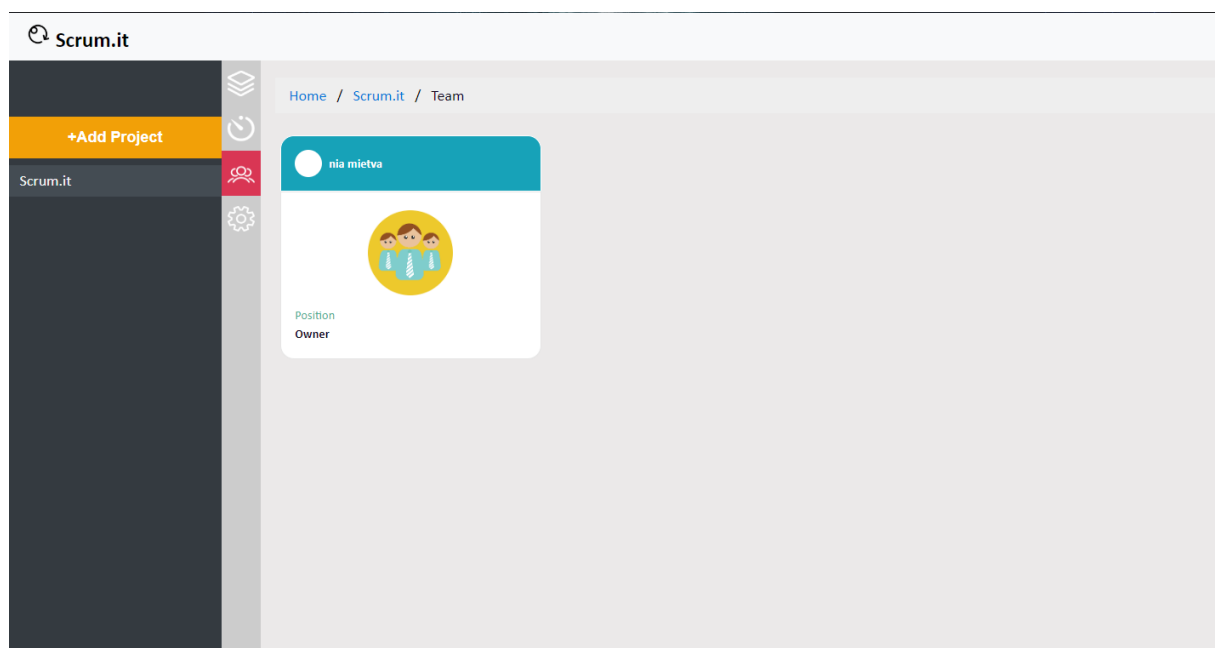
- таб 2 – активен спринт
= tooltip с dropdown



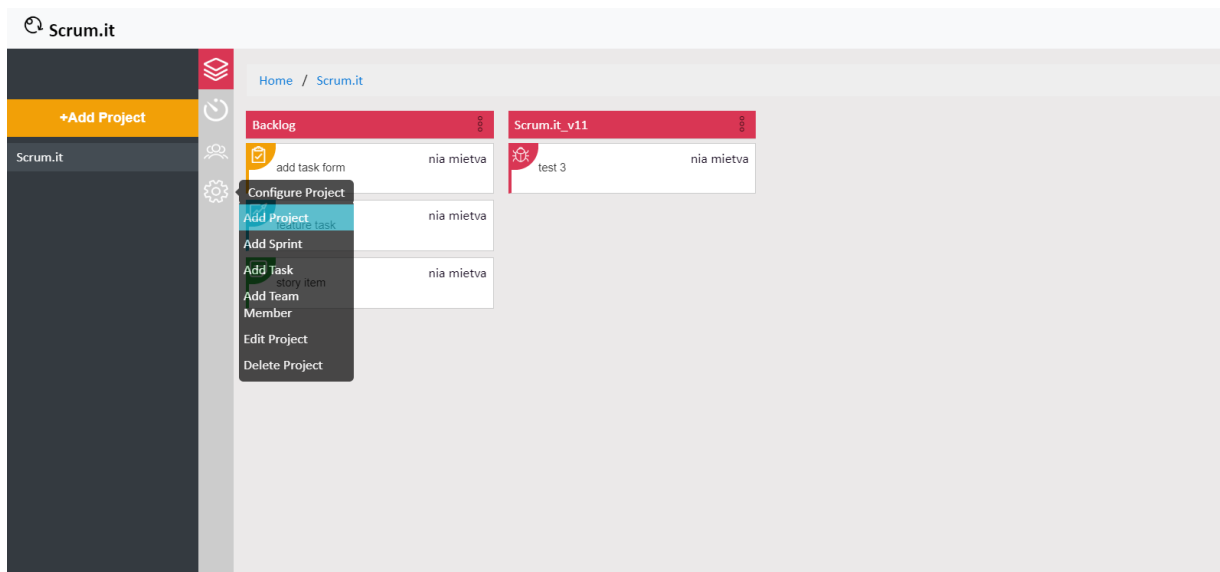
= спикъци според статус на задача
 = бутон за добавяне на задача в десния край на заглавната част на списъка



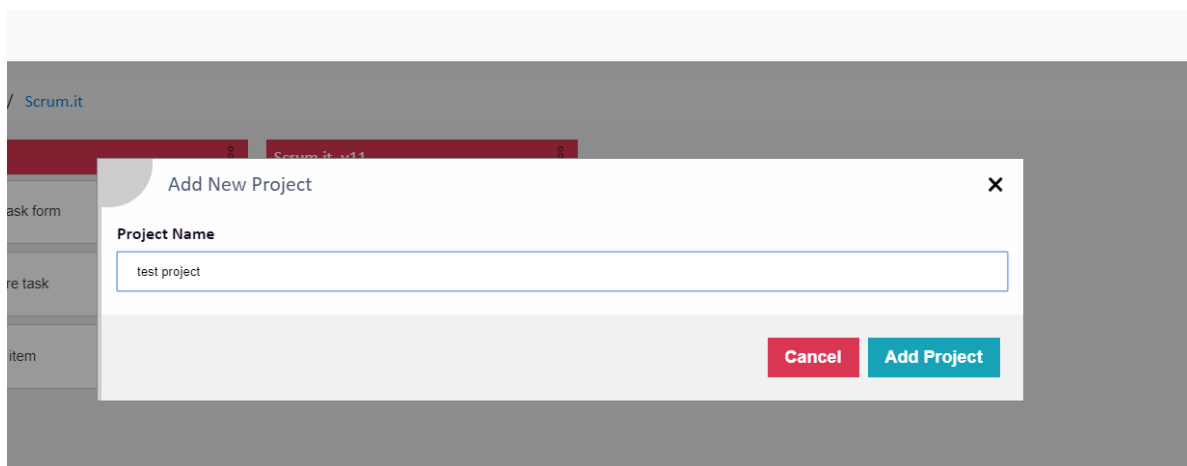
- таб 3 – екип, който работи по проекта
 = информационни картичка за всеки член от екипа

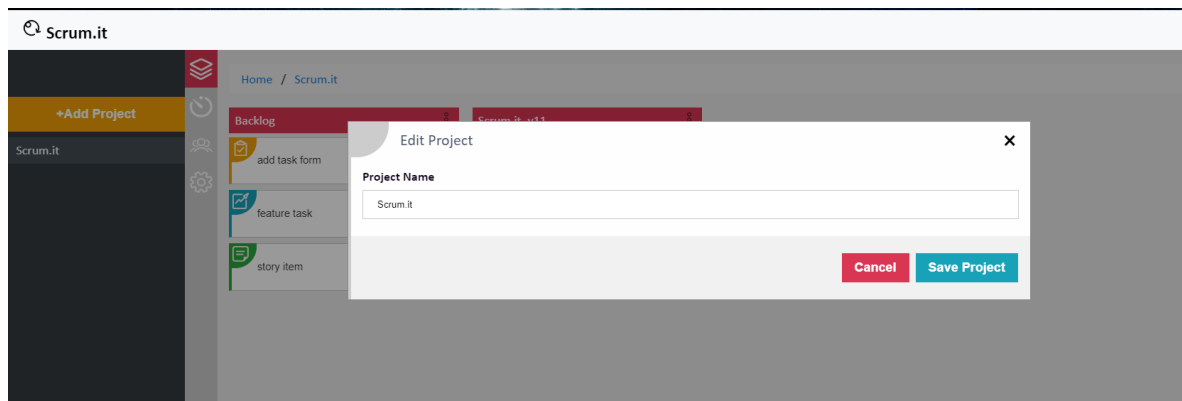


- таб 4 – конфигурационно меню за конкретния проект
= tooltip + dropdown menu

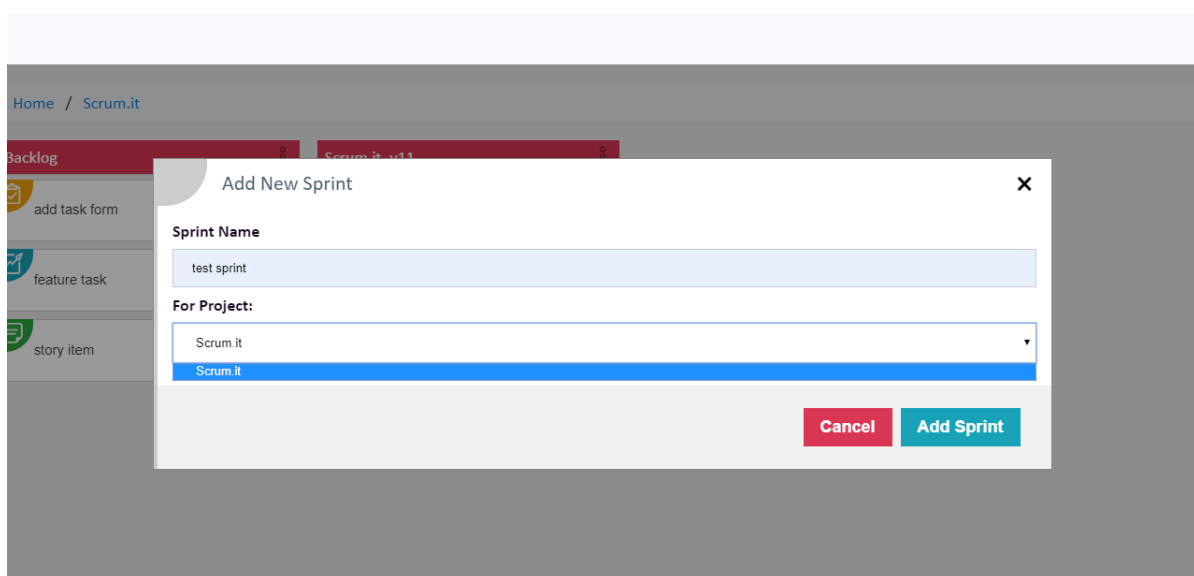


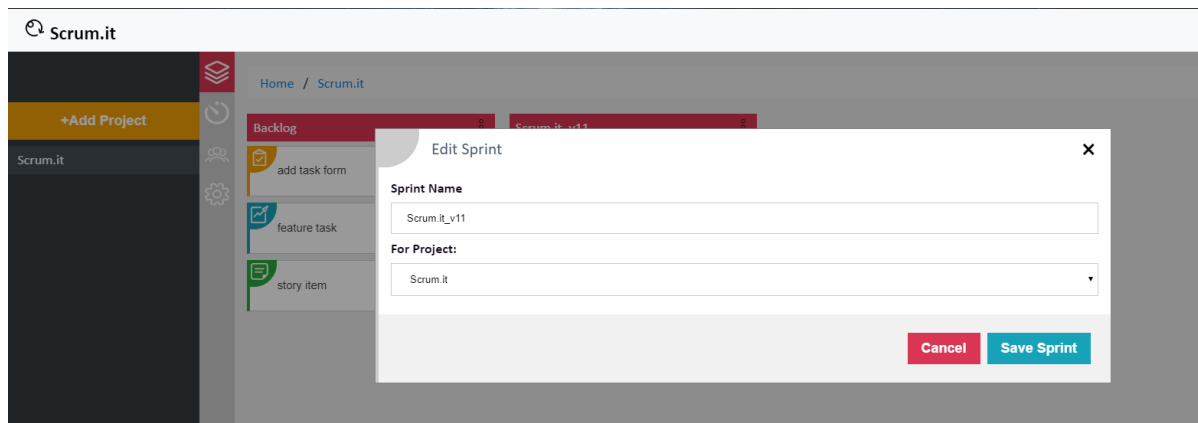
- **Добавяне на проект / редакция на проект + изтриване**
 - форма, в която се попълва име на проект
 - добавяне от бутона за добавяне в главната странична навигация
 - добавяне/редактиране/изтриване от конфигурационното меню а проект от таб менюто в страницата на съответния проект



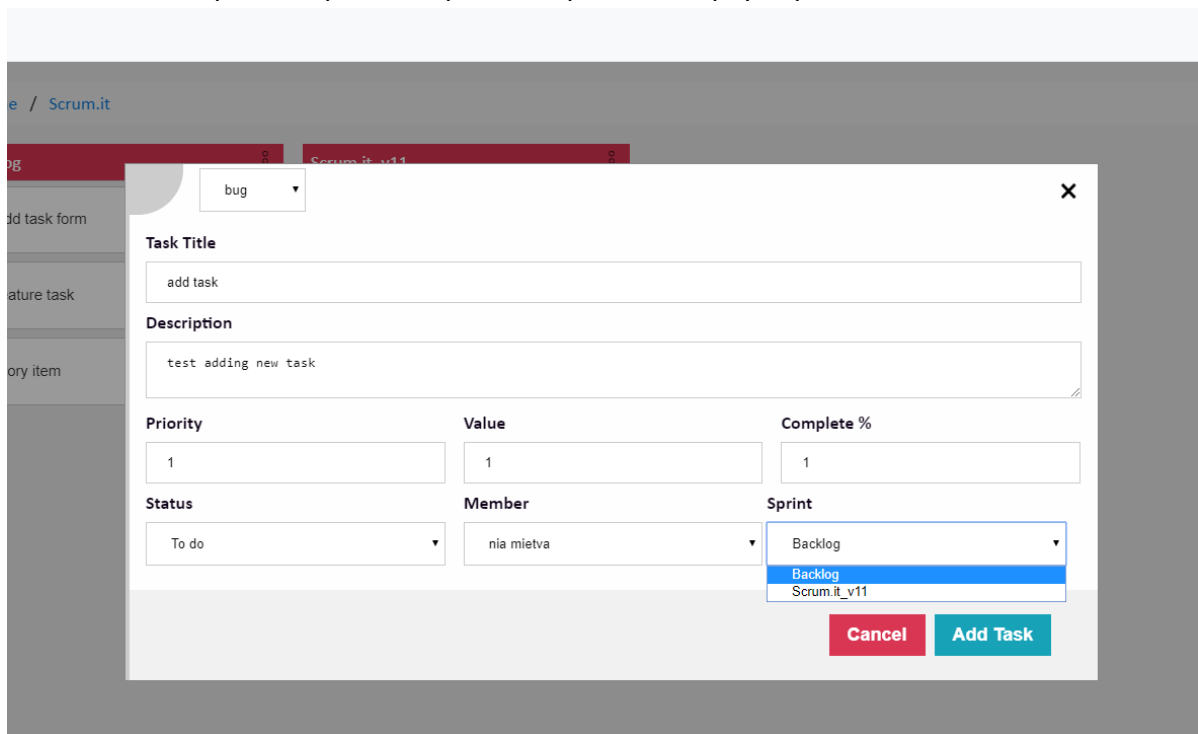


- **Добавяне/редактиране на спринт + изтриване**
 - форма, в която се попълва име на спринт и се избира проект от падащо меню, към който ще принадлежи спринтът
 - добавяне от конфигурационното меню на проект (таб менюто в страницата на съответния проект)
 - добавяне/редактиране/изтриване от конфигурационното меню в десния ъгъл на заглавната част на сприн списъка





- **Добавяне на задача + редактиране + изтриване**
 - форма, в която се попълват всички данни нужни за създаване на задача
 - добавяне от конфигурационното меню на проект
 - добавяне от конфигурационното меню на спринт цикъл
 - редактиране/изтриване при клик върху картичка на задача



Scrum.it

Home / Scrum.it

Backlog

add task form

feature task

story item

Task Title

feature task

Description

asd

Priority

1

Value

1

Complete %

1

Status

To do

Member

nia miteva

Sprint

Backlog

Cancel Save Delete

- **Добавяне на член на екип + редактиране + изтриване**
 - форма, в която се попълват всички данни нужни за добавяне на член на екипа
 - добавяне от конфигурационното меню на проект
 - редактиране/изтриване при клик върху картичка на член на екипа

Scrum.it / Team

niya miteva

Member Name

niya miteva

Email

asd@asd.com

Project

Scrum.it

Position

Developer

Developer

PM

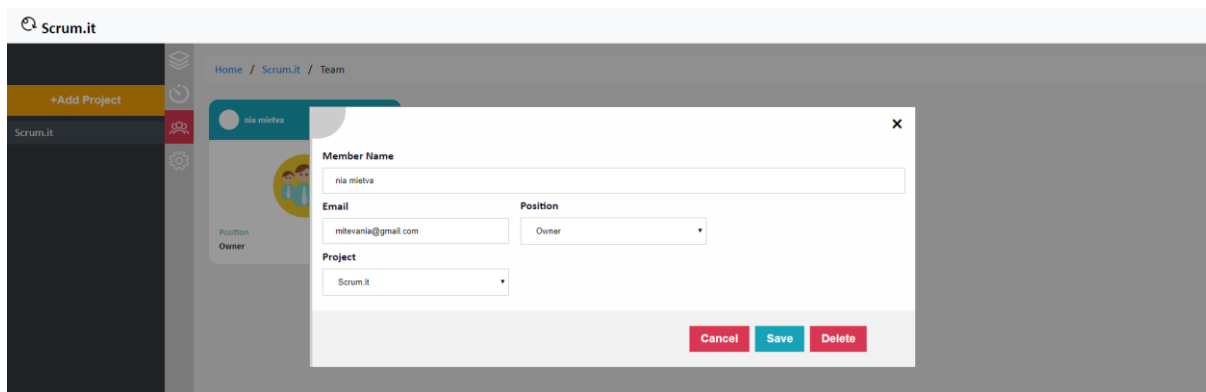
Owner

Designer

QA

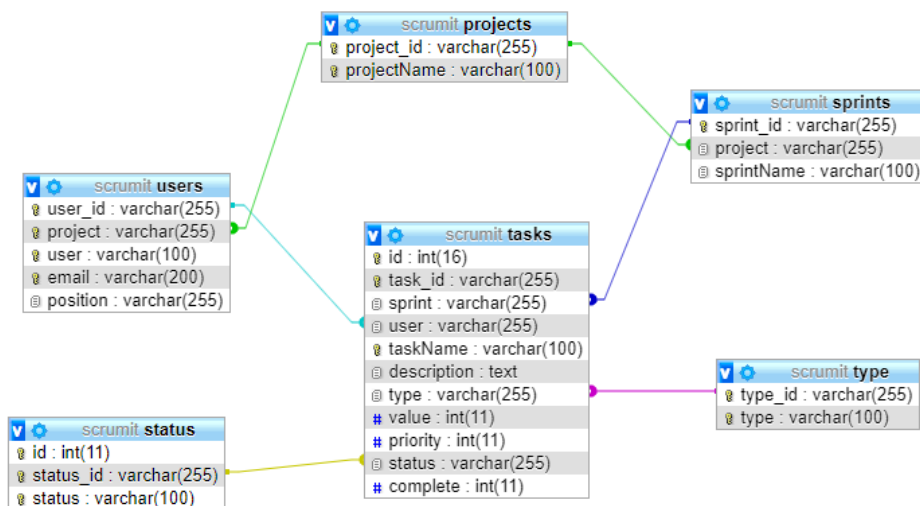
Software Architect

Cancel Add Member



7. Описание на програмния код

Базата от данни се състои от таблици, подходящо именувани, като в тях да се съхраняват данните на потребителите, данните за проектите, техните sprint-ове и задачи и т.н.



Файлова архитектура:

```

scrum_it
|--app
|---|---- config
|---|---- controllers
|---|---- models
|---|---- view
|---framework
|---|---- core
|---|---- database
|---src
  
```

```
|----|---- img
|----|---- styles
|----|---- scripts
|---- index.php
```

Папката **framework** представлява съвсем прост **MVC framework**. Тя съдържа базовите класове за **Model, View, Controller** частите, които се намират във вътрешната папка **core**, където е и конфигурационния файл **framework.php**, в който са дефинирани главните пътища към отделните файлове. **BaseModel.php** съдържа всички основни заявки, ползващи се за всички обекти. **BaseView.php** играе роля на „лепило“ за всички части на една страница (header, nav, main, footer). Вътрешната папка **database** съдържа файла **connection.php**, където се осъществява връзката с базата данни.

BaseModel.php:

- Съдържа основните заявки към базата данни за всички обекти
- Получава връзка към базата данни
- Взима всички записи за даден обект
- Взима запис по ID за даден обект
- Взима всички записи по FK за даден обект
- Изтрива запис по ID от даден обект
- Вмъква запис към даден обект
- Променя запис по ID за даден обект

BaseController.php

- Получава записи от базата данни за обектите съдържащи се във всяка страница.
- Събира всички части на страницата и ги изпраща на BaseView за визуализиране

Папката **app** съдържа същинската част на приложението. Връзката с използваните файлове/класове се осъществява във **core** файловете в папката **config**. В **controllers**, се намират отделните **Controller** класове за всеки обект, където се модифицират или събират данните, които ще се показват на потребителя. В **models**, се намират отделните **Model** класове, които дефинират обектите използвани за приложението. В тези класове се съдържат и заявките към базата данни за съответните обекти. В **views**, се съдържат **View** класовете с темплейтите за отделните обекти. Model, View, Controller класовете наследяват съответните си базови класове.

Всяка страница има **header** с **navbar**, **sidenav**, **main** (съдържанието на съответната страница), **footer**. Съдържанието в **main** е **view** класът на съответния вид обект за който се отнася страницата. **Controller**-ът има метод **display**, където се задава съдържанието, което ще бъде представено и визуализирано в съответните части на страниците. Този

метод извиква **display** метода на **View** класовете, които „попълват“ данните в **html** шаблоните.

CSS стилът и JavaScript функциите се съдържат в папката **src**, където се намират и снимки, иконки и картинки използвани в приложението.

8. Приноси на студента, ограничения и възможности за бъдещо разширение

Преди да започна да разработвам приложението, направих щателен research на различни подобни приложения за това кои функционалности е задължително да има, за да е полезно при управление на scrum процесите. Събрах си общи изисквания за системата и нарисувах mockup-и за отделните страници и елементи. Всички това може да се види в прикачения файл [brainstorm_scrumit.pdf](#).

Начинът, по който е реализирано уеб приложението, предоставя възможност за лесно внедряване на нови функционалности.

За в бъдеще са предвидени следните функционалности и възможности на системата:

- Подобряване на визуалната част на проекта
- Оптимизации по кода
- Валидация на данни
- Да се добави вписване в системата
- Да се добавят сесии
- Стартиране и прекратяване на спринт цикъл
- Да се добавят права за достъп за различните потребители
- Да се добави администраторски панел , през който:
 - Да се добавят потребители (при добавяне да се генерира и парола, с която добавения потребител да се вписва в системата със съответни права за достъп). За добавяне на потребител към проект ще се премахне съществуващата форма и ще се добави списък с всички потребители на системата, от който да се избират потребители работещи по проекта.
 - Да се добавят нови типове задачи
 - Да се добавят нови длъжности на потребители
- да се ползва от различни организации:
 - регистрационна форма за организация
 - всяка организация си има отделни потребители, т.е. хората работещи във фирмата
 - всяка организация си има собсвен админ/админи, които се грижат за добавянето на потребители и дали информацията която се добавя и обработка е изрядна

- организация няма достъп до данните на друга организация
- да се добави Planning Poker функционалност
- да се добави календар за организация на scrum срещите
- да се добавят филтри и търсачки за задачи и потребители
- всеки проект да има така наречени „дъски“ за отделен екип разработващ съответна част от проекта:
 - примерни дъски/екипи: „Bisnes intelligence“, „Security“, „Front-end“, „Mobile Version“
 - за всяка „дъска“ да има отделен backlog списък и sprint цикли.
- Да се добавят reminders и notifications за потребителите на системата
- Мобилно приложение на системата

9. Какво научих

Благодарение на работата по проекта:

- Научих подробно как работи MVC модела:
 - Да се изолира презентационни слой от бизнес логиката, за да може всяко от тях да е обособено самостоятелно и да са независими части от архитектурата, които единствено си комуникират.
- Успях да разработя MVC framework, което още повече задълбочи знанията ми за модела
- Научих как се създава приложение с подходяща архитектура, така че в бъдеще да е лесно добавянето на нови функционалности. Както и при промяна на визията на приложението тя да не зависи от логиката.
- Обогатих знанията си по PHP, CSS, HTML и JavaScript
- Научих какви са основните особености на scrum процесите за управление на проекти
- Имам собствено приложение, което разработих from scratch. Разработката ми ми показва, че за тази цел са необходими много познания, постоянство и труд.
 - Front-end
 - Back-end
 - Дизайн на база данни
 - Дизайн на архитектура на проект
 - Събиране на изисквания за определяне на функционалностите на системата
 - Създаване на mockup прототипи на приложението, които да се използват като план при разработката

10. Използвани източници

1. <https://www.w3schools.com/>
2. <https://www.codeofaninja.com/2017/02/create-simple-rest-api-in-php.html>
3. <https://stackoverflow.com/>
4. <https://www.php.net/docs.php>
5. <https://medium.com/@noufel.gouirhate/create-your-own-mvc-framework-in-php-af7bd1f0ca19>
6. <https://medium.com/shecodeafrica/building-your-own-custom-php-framework-part-1-1d24223bab18>
7. <https://www.sitepoint.com/the-mvc-pattern-and-php-1/>
8. <https://www.codeproject.com/Articles/1080626/Code-Your-Own-PHP-MVC-Framework-in-Hour>
9. <https://www.oreilly.com/library/view/learning-php-mysql/9781449337452/> - Learning-PHP-MySQL-JavaScript-and-CSS-2nd-Edition-1
10. <https://www.oreilly.com/library/view/php-cookbook-3rd/9781449363741/> - PHP Cookbook, 3rd Edition