



СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Учебен Проект

по Системи за е-бизнес,
зимен семестър 2022/2023

НА ТЕМА: "KenTrack" - Инструмент за управление на
задачите към един проект

Изготвил:

Кенан Юсеин

фак. № 71947

спец. Информационни системи , 4 курс

София,

Януари 2023

СЪДЪРЖАНИЕ

Описание на учебния проект.....	2
Архитектура и дизайн на системата.....	2
Потребителски интерфейс (UI):.....	3
Бизнес логически слой:.....	3
Слой Бази данни:.....	3
Външна система за удостоверяване.....	4
Диаграма на потребителските случаи (Use case diagram).....	5
Етапи на разработка.....	5
Описание на е-бизнеса / е-услуга.....	6
Обхват.....	6
Описание на цялостната реализация.....	6
Описание на Базата от Данни.....	11
E/R диаграма на базара данни.....	11
Домейн модел.....	13
Описание на категориите и отделите.....	16
Описание на съхранените процедури.....	16
Допълнение.....	17
Продукция.....	17

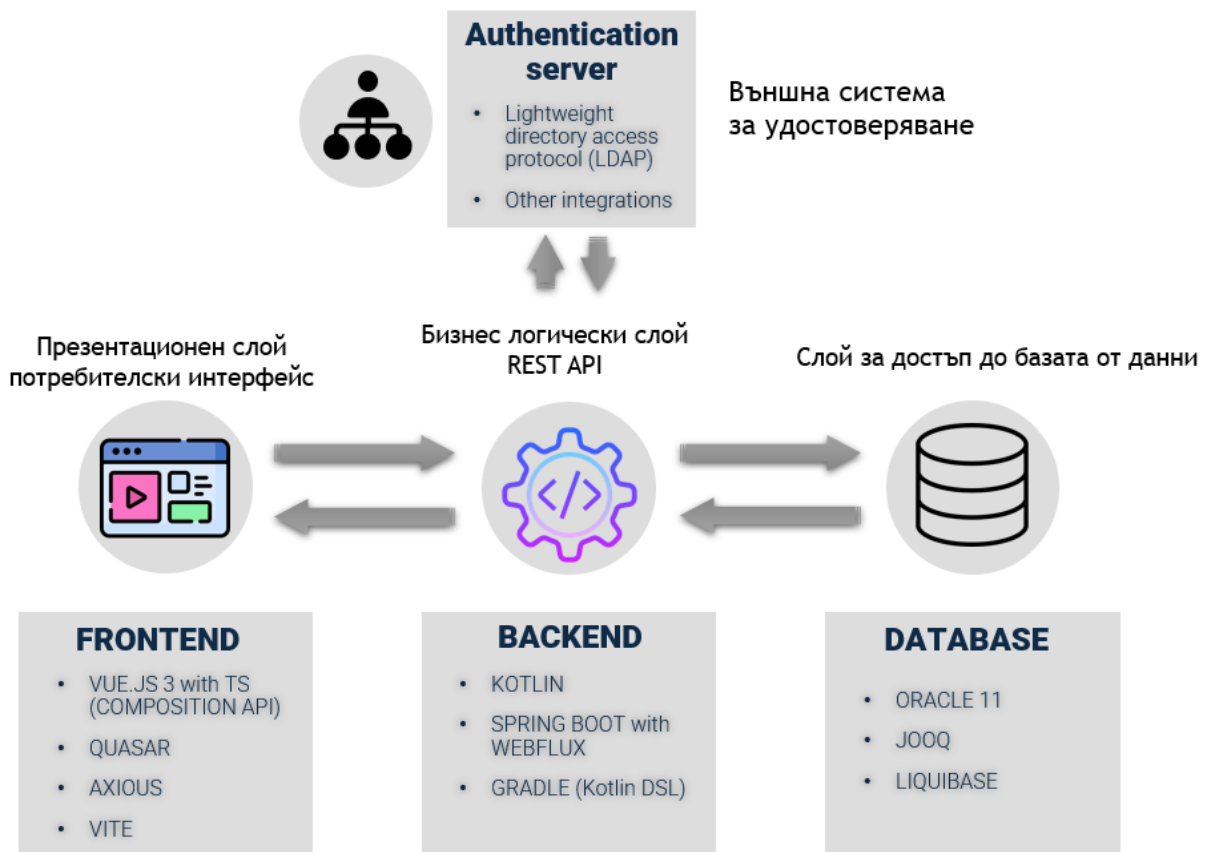
Описание на учебния проект

Текущият проект, KenTrack, представлява "Инструмент за управление на задачите към един проект". Идеята е базирана на "Канбан" подходът - проектите на дадена организация(фирма) да са представени върху табла, които съдържат списъци от задачи за изпълнение, които се разработват от съответните потребители в системата.

Потребителите могат да създават/редактират табла и създават/редактират задачи, които са част от тези табла. Задачите могат да бъдат давани за разработка на потребителите, които участват в съответното табло. Всяка задача има статус, който показва текущото положение на съответната задача. Пази се история за всички промени.

Архитектура и дизайн на системата

За разработката на KenTrack е използван пакет от съвременни технологии, които предоставят гъвкавост и лесна интегрируемост на продукта. За всеки от слой от архитектурата на проекта са избрани следните технологии описани по-долу:



Потребителски интерфейс (UI):

1. **VUE.JS 3 (Composition API) with Typescript**
2. **Quasar** - Quasar е базирана на Vue.js и е UI библиотека, която съдържа набор от инструменти за бързо и лесно създаване на уеб приложения и уеб сайтове. Състои от три компонента: CSS, Компоненти на потребителския интерфейс и Javascript/Typescript.
3. **Axios** – Axios е HTTP клиентска библиотека за браузъри. Използва се за пращане на HTTP заявки към Spring сървър и получаване на отговор.
4. **Vite** – Frontend Build tool - Инструмент за бързо разработване на съвременни уеб проекти. Фокусира се върху скоростта и производителността чрез подобряване на опита за разработка. Vite има HMR (hot module replacement) което позволява промените по Vue компонентите да се визуализират веднага без нужда от re-build.

Бизнес логически слой:

1. **Kotlin** – избраният език за разработка е Kotlin – той е статично типизиран език за програмиране. Kotlin е проектиран да взаимодейства напълно с Java и JVM, което позволява да се използва Spring boot.
2. **Spring Boot with WEBFLUX**
Spring Boot framework е базирана на Java, използва се за създаване на микро услуга и за изграждане на самостоятелни и готови за производствено приложения. Spring създава и деплоува чрез TomCat нашия сървър, който прилага REST API-то и бизнес логиката на приложението. То е връзката между потребителския интерфейс и базите от данни, също така позволява интеграция с външната система за удостоверяване. WEBFLUX позволява конкурентно използване на приложението от повече потребители.
3. **Gradle (Kotlin DSL)** – build tool, dependency management...

Слой Бази данни:

1. **Oracle 11** – Избраната база от данни е Oracle 11, защото тя е обектно-реляционна

система за управление на база данни и прилага обектно-ориентирани характеристики като дефинирани от потребителя типове, наследяване и полиморфизъм.

2. **JOOQ Typesafe querying** - Object Oriented Querying, по-известен като jOOQ, е лека софтуерна библиотека за картографиране на бази данни в Java/Kotlin, която прилага активния модел на запис. Целта му е да бъде едновременно реляционен и обектно ориентиран, като предоставя специфичен за домейн език за конструиране на заявки от класове, генерирани от схема на база данни.

3. LIQUIBASE

- Liquibase е независима от база данни библиотека за проследяване, управление и прилагане на промени в схемата на базата данни. Позволява по-лесно проследяване на промените в базата данни, особено в гъвкава среда за разработка на софтуер където разработчиците са много на брой.
- Държи всички инстанции на базите от данни в синхронизация (среда за разработка, тестване, интеграция, продукция..)

Външна система за удостоверяване

Имплементирана LDAP интеграция – при предоставен LDAP сървър, приложението може да използва външна система за удостоверяване на потребители.

Целта е продуктът да може да бъде предоставен на вече съществуващи компании, които вече имат система за удостоверяване на служителите си и продуктът ни да може да се интегрира лесно с техния набор от потребители(служители).

Lightweight Directory Access Protocol е отворен, неутрален по отношение на доставчика индустриален стандартен протокол за приложение за достъп и поддържане на информационни услуги за разпределени директории през мрежа с интернет протокол

Диаграма на потребителските случаи (Use case diagram)



Етапи на разработка

1. Планиране:

- Определяне на обхват и цели
- Определяне на нефункционални и функционални изисквания,
- Изготвяне на първоначален списък на потребителските случаи.

2. Детайлизиране:

- детайлизиране на потребителските случаи
- изготвяне на прототип на потребителския интерфейс.
- Изготвяне на план за управление на качеството и риска.
- Определяне на изискванията за системната архитектура и избиране на технологии, които ще се ползват за разработка

3. Разработка:

- Конфигурация на необходимите среди за имплементация на проекта (среда за разработка, тестване, интеграция)
- Разработване на софтуерната архитектура
- Създаване на инстанция на базите от данни – създаване на домейн модел, нормализация на базата от данни, създаване на таблиците и връзките между тях, конфигурация на инструмент за мигриране на бази данни.
- Разработване на системните функционалности - Създаване на REST API, реализиране на категориите на услугите
- Тестване – премахване на потенциално появили се грешки

4. Завършване

- Извършване на UAT и отстраняване на новооткрити грешки
- Изготвяне на документация и ръководство за употреба

- Качване на продуктова среда

Описание на е-бизнеса / е-услуга

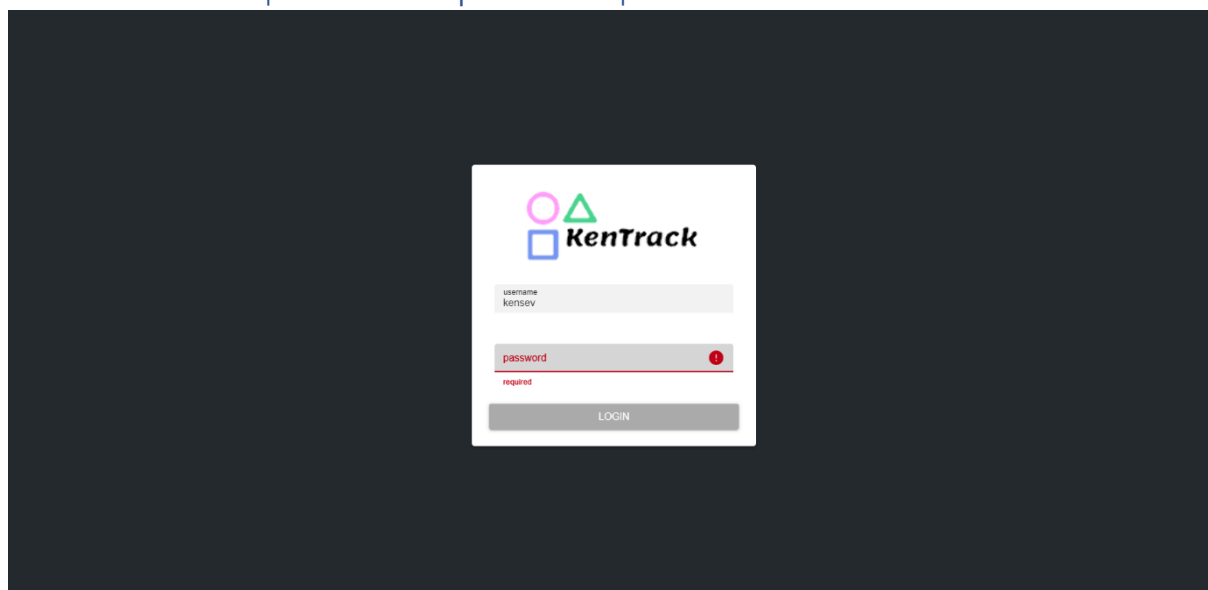
KenTrack, Инструмент за управление на задачите към един проект, е изключително полезен продукт за планиране и контрол на времето в една компания. Time management (управление на времето) е ключов за успеха на една компания, защото позволява на бизнеса да планира правилно ресурсите си. Ресурсите са време, хора и пари и когато фирмата знае какво прави, спестява разходи и чрез KenTrack може да го получи. WEB Приложението предоставя на ръководителите на компанията да следят продуктивността на всеки служител, също така да следи успеха и прогреса. Освен това мениджъра може да следи колко време се изпълняват задачите и колко задачи може да изпълни дадения служител за определен период от време. KenTrack е подходящ и за всеки, който иска да контролира, планира и приоритизира лични своите задачи или бизнес цели.

KenTrack предлага безплатната версия, при която потребителя може да създава до 3 табла и до 20 задачи, които не са „DONE“. Има само секциите TO DO, IN DEVELOPMENT, DONE. При нужда от повече boards и създаване на собствени допълнителни секции, трябва да се закупи платена версия.

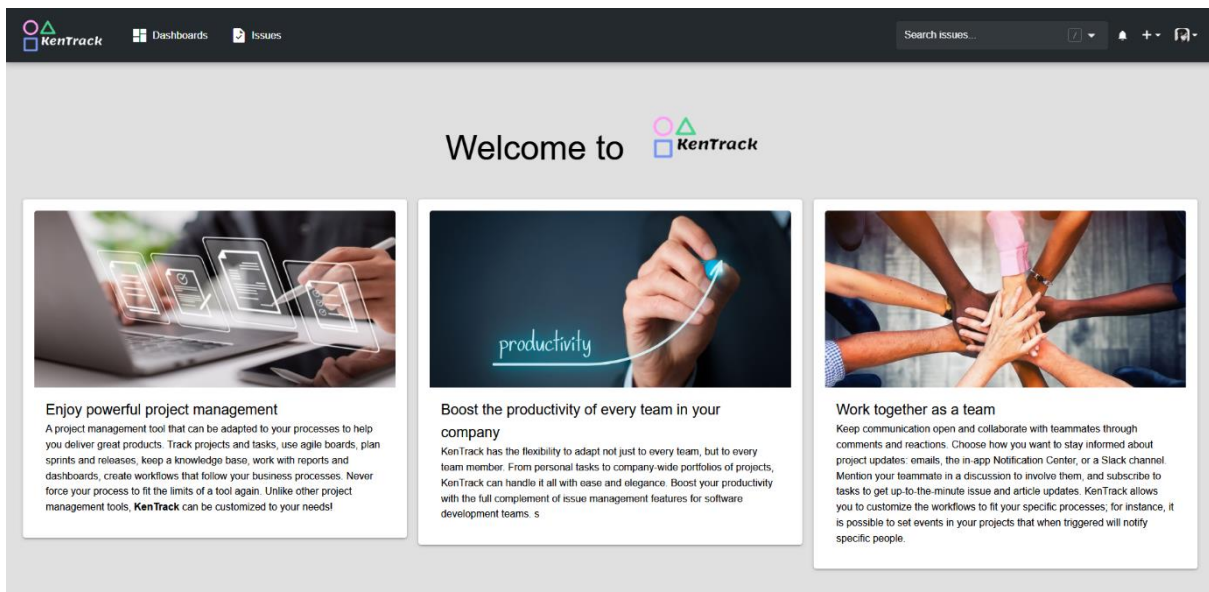
Обхват

- Вход в профила(login) чрез външна система за удостоверяване (LDAP) на съответната компания, която ползва приложението.
- Изход от профила (logout)
- Създаване на задача – по категории (нисък приоритет, висок приоритет, среден приоритет)
- Редактиране на задача
- Изтриване на задача
- Създаване на коментари под съответна задача
- Редактиране на коментар под съответна задача от оторизиран потребител
- Изтриване на коментар под съответна задача от оторизиран потребител
- Редактиране на секция от потребител от оторизиран потребител
- Създаване на борд от потребител
- Редактиране на борд от потребител от оторизиран потребител
- Добавяне на потребители към борд от оторизиран потребител

Описание на цялостната реализация



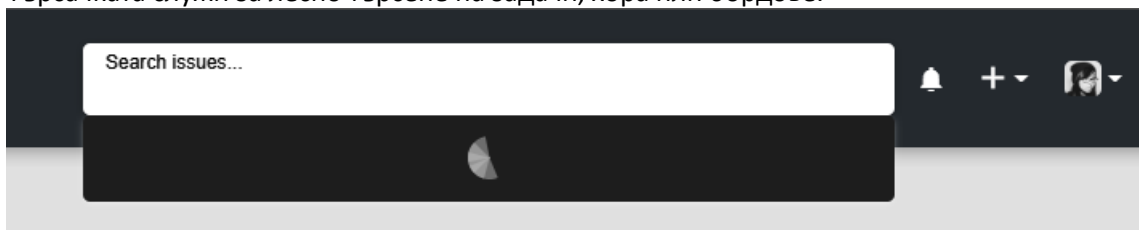
При отваряне на приложението пред потребителя се показва LOGIN страницата, чрез която може да се направи LDAP удостоверяване. Съществуват валидации за съответните полета. При коректни въведени username и password потребителят се пренасочва към началната страница на KenTrack.



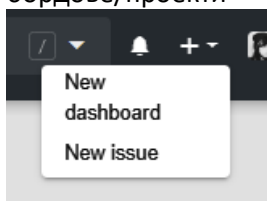
Началната страница съдържа цялостната информация за проекта KenTrack. Също така и главния 'layout' меню, откъдето може да се използват останалите функционалности на приложението.

Първият бутон с лого 'KenTrack' пренасочва към началната страница.

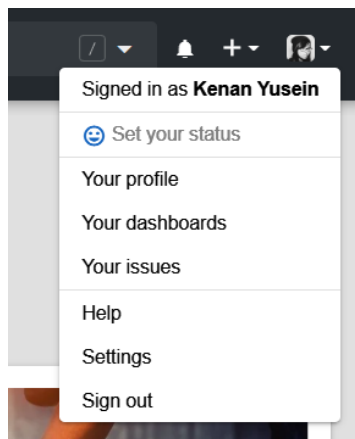
Търсачката служи за лесно търсене на задачи, хора или бордове.



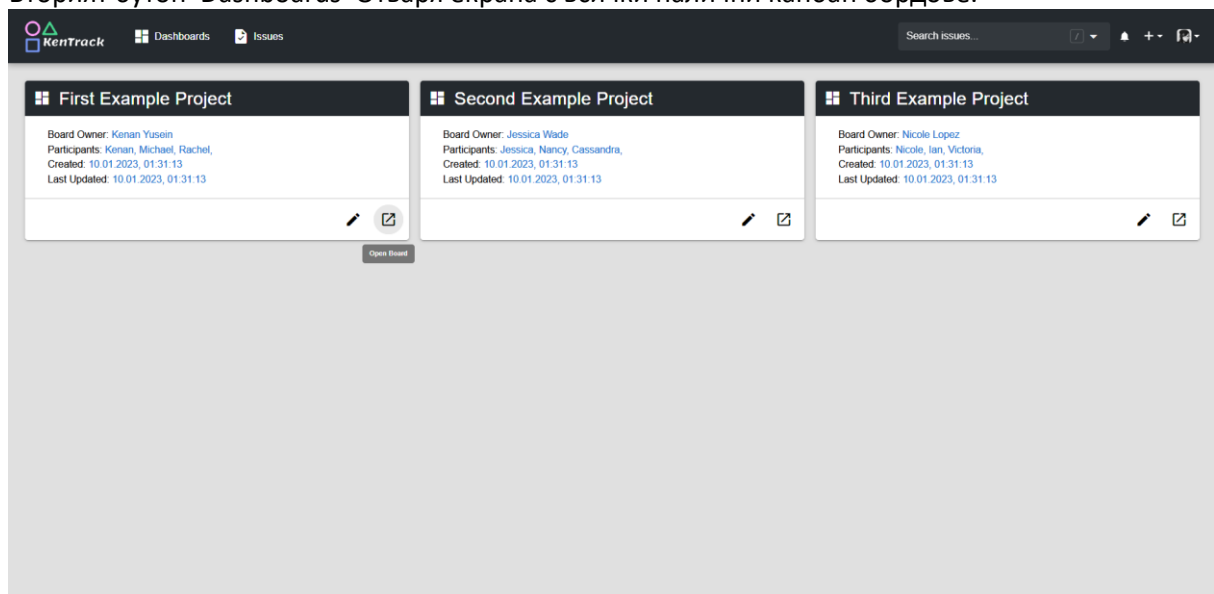
Чрез '+' може да се създаде нов борд или нова задача към някой от съществуващите вече бордове/проекти



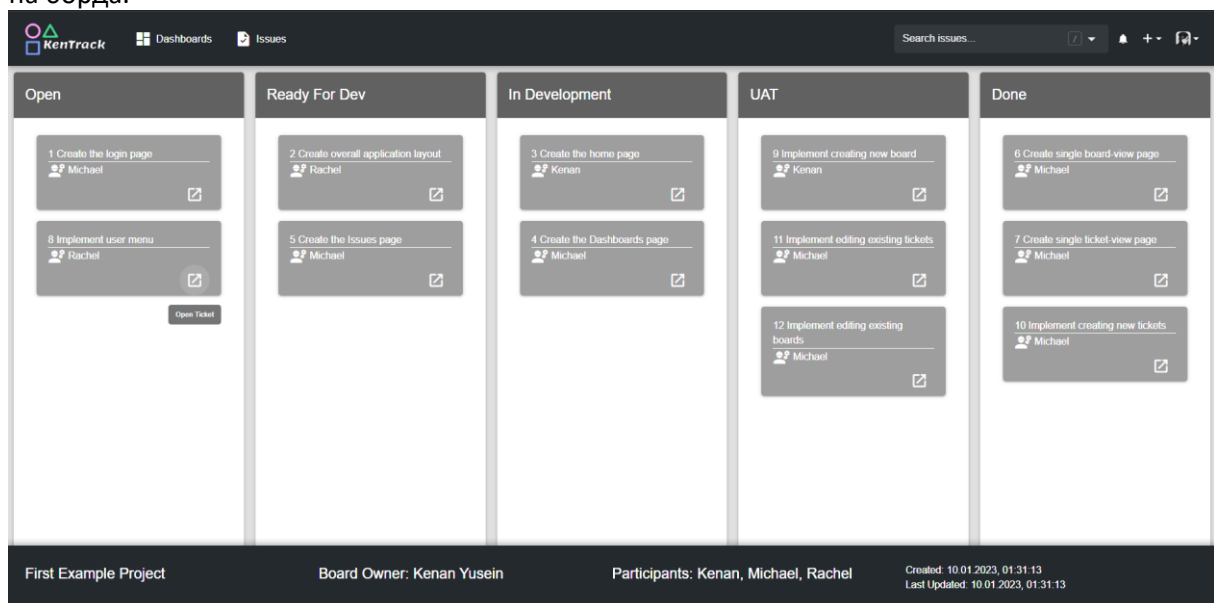
Натискането на профила отваря следното меню със следните функционалности:



Вторият бутон 'Dashboards' Отваря екрана с всички налични канбан бордове.



Тук може да видим основната информация за всяка от наличните бордове/проекти за дадената компания, която ползва KenTrack. Съответните бутони за отваряне на борда и редактиране на борда.



Така изглежда един отворен борд. Най-отдолу пак може да видим информацията за съответния

отворен борд, а по средата виждаме разграфени всички налични секции за отворения борд и съответните задачи, които се намират в съответната секция.
За всяка задача се вижда в какъв статус е, кой го разработва и ID със заглавие на задачата.
Всяка задача може да се отвори за повече информация:

The screenshot shows the KenTrack interface with the 'Issues' tab selected. A modal window displays the details for 'Issue 1 - Create the login page'. The description states: 'The login page must be created with the following criterias: It should following the base applicaiton layout. It should have a box in the middle with two text input fields - one for username and one for password. A login button must be placed underneath them. Validations for both input fields must be implemented. Clicking login must send login request to the LDAP and if the user is authenticated it should proceed to the home page. IF not an error message must be displayed.' Below the description are two comments by 'Kenan Yusein' from 10.01.2023, 01:40:28, labeled 'Test comment 1' and 'Test comment 2'. An 'ADD COMMENT' button is at the bottom.

След като задачата бъде отворена, допълнителните полета за нея се показват в следния екран по-горе. Виждаме подробното описание на задачата и коментарите направени по нея подредени по хронологичен ред. Чрез бутона горе в дясно задачата може да се редактира.

Следващият бутон 'Issues' от главното меню отваря екрана с всички задачи по всички проекти и бордове.

The screenshot shows the KenTrack interface with the 'Issues' tab selected. A table lists 12 tasks. Each row includes an ID, Title, Owner, Assignee, Status, Board, Created, and Updated. Action icons (edit, delete, open) are in the rightmost column.

ID	Title	Owner	Assignee	Status	Board	Created	Updated	
1	Create the login page	Kenan	Michael	Open	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
2	Create overall application layout	Kenan	Rachel	Ready For Dev	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
3	Create the home page	Kenan	Kenan	In Development	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
4	Create the Dashboards page	Kenan	Michael	In Development	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
5	Create the Issues page	Kenan	Michael	Ready For Dev	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
6	Create single board-view page	Kenan	Michael	Done	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
7	Create single ticket-view page	Kenan	Michael	Done	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
8	Implement user menu	Kenan	Rachel	Open	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
9	Implement creating new board	Kenan	Kenan	UAT	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
10	Implement creating new tickets	Kenan	Michael	Done	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
11	Implement editing existing tickets	Kenan	Michael	UAT	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	
12	Implement editing existing boards	Kenan	Michael	UAT	First Example Project	10.01.2023, 01:40:28	10.01.2023, 01:40:28	

Бутоните в дясно служат съответно за редактиране на задача, изтриване на задача и отваряне на съответната задача, което отваря страницата за задачи показано по-горе.

Диалог за създаване и редактиране на задачи:

Manage Issue

×

ID

1

Title

Create the login page

B

I

U

Align

Formatting

Font Size

Default Font

Second Example Project

Board

Partic

Creat

Last U

Manage Board

ID

1

Title

First Example Project

Owner

Kenan

Participants

Kenan Yusein

Michael Avery

Rachel Elliott

Created

10.01.2023

Updated

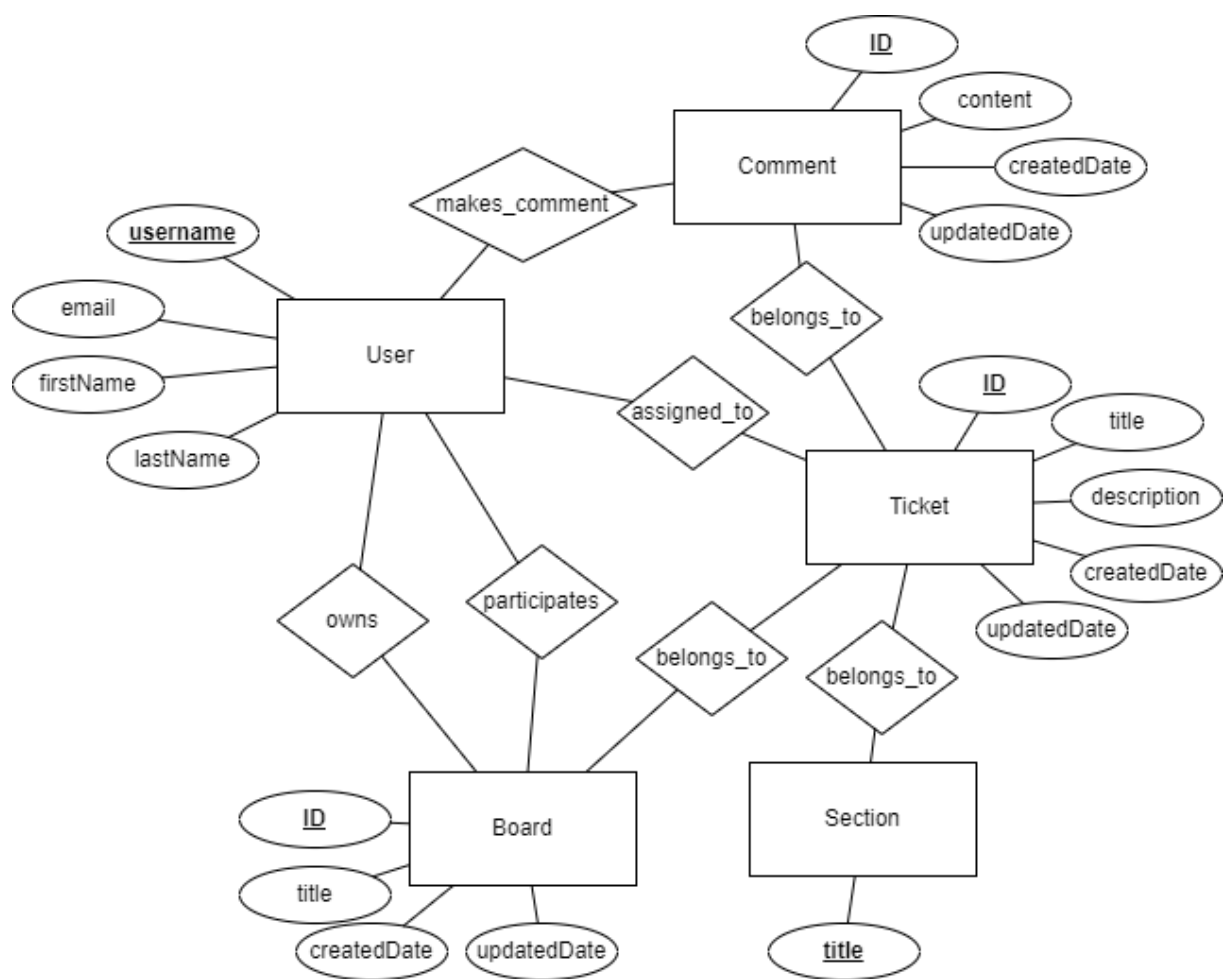
10.01.2023

SUBMIT

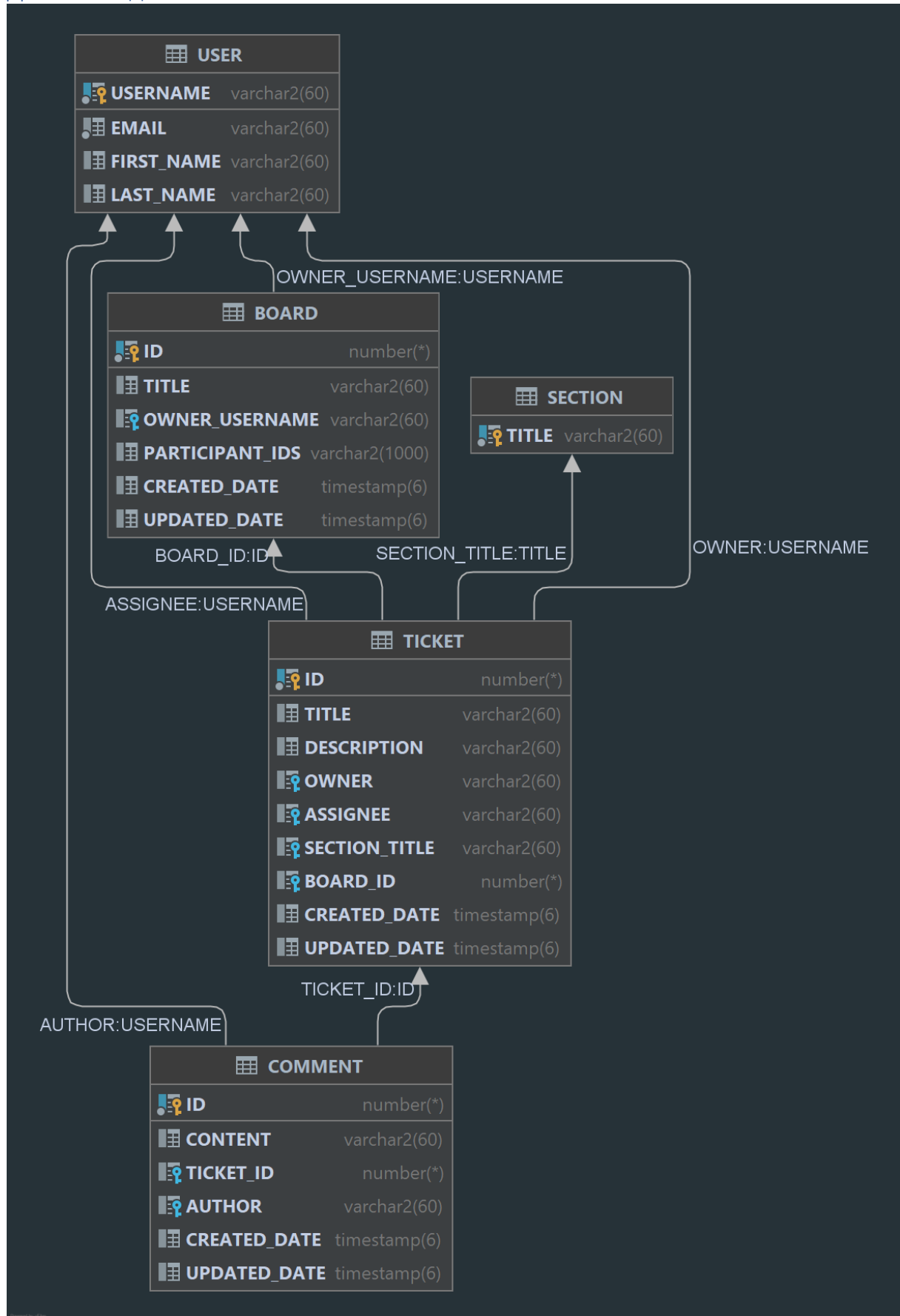
CANCEL

Описание на Базата от Данни

Е/Р диаграма на базара данни.



Домейн модел



Потребител

Колона	Тип	Стойност по подразбиране	Описание
username	String	N/A	Тази стойност е уникална за всеки потребител. Взима се от LDAP
firstName	String	N/A	Име на потребител в LDAP
lastName	String	N/A	Фамилия на потребител в LDAP
email	String	N/A	Емайл на потребител в LDAP
createdDate	Date	Датата в момента на създаване.	Датата на създаване на потребителския профил.
updateDate	Date	Датата в момента на редакция.	Датата от последната редакция, която е извършена от конкретния потребителски профил.

TaskSchema

Колона	Тип	Стойност по подразбиране	Описание
Id	String	N/A	Тази стойност се генерира автоматично от Oracle Sequence
title	String	N/A	Име на задачата, което се вписва от конкретния потребител при създаването ѝ.
description	String	N/A	Допълнително описание на задачата, което се попълва от конкретния потребител при създаване на задачата.
owner	String	Приема идентификационния номер на потребителя автоматично при създаване на задачата.	Идентификационния номер на потребителя, който се взима автоматично при създаване на задачата. FOREIGN KEY към таблицата USER
assignee	String	N/A	Име на потребител, което се избира от конкретен owner от падащото меню, като по подразбиране може да не бъде избран такъв FOREIGN KEY към таблицата USER
section	String	TO DO/NOT READY	Приема се идентификационен номер на съответната секция. FOREIGN KEY към таблицата SECTION
board	String	Приема по подразбиране идентификационен номер на съответния борд	Борда, в който се намира съответната задача. FOREIGN KEY към таблицата BOARD
createdDate	Date	Датата на създаване на задачата.	Датата на създаване на задачата.
updatedDate	Date	Датата на последната редакция.	Датата от последната редакция на задачата.

SectionSchema

Колона	Тип	Стойност по подразбиране	Описание
id	String	N/A	Тази стойност се генерира автоматично от Oracle Sequence
title	String	N/A	Име на секцията, което се задава от конкретния потребител при създаване на секцията.
createdDate	Date	Датата в момента на създаване.	Датата в момента на създаване на секцията от съответния потребител.
updatedAt	Date	Датата на последната редакция.	Датата на последната редакция на секцията.

CommentSchema

Колона	Тип	Стойност по подразбиране	Описание
id	String	N/A	Тази стойност се генерира автоматично от Oracle Sequence
content	String	N/A	Показва текстовото съдържание на коментара, задължително поле.
author	String	Идентификационен номер на потребителя, въвел коментара.	Идентификационен номер на конкретния потребител, който е създал коментара. FOREIGN KEY към таблицата USER
taskId	String	Идентификационния номер на задачата, за която се отнася коментара.	Идентификационния номер на конкретната задача, за която се отнася коментара. FOREIGN KEY към таблицата TICKET
createdDate	Date	Датата в момента на създаване.	Датата в момента на създаване на коментара от съответния потребител.
updatedAt	Date	Датата на последната редакция.	Датата на последната редакция на коментара.

BoardSchema

Колона	Тип	Стойност по подразбиране	Описание
id	String	N/A	Тази стойност се генерира автоматично от Oracle Sequence.
title	String	N/A	Име на борда, което се задава от конкретния потребител при създаване на борда.
owner	String	Идентификационен номер на потребителя, който е създал board - а.	Идентификационен номер на потребителя, който е създал board - а. FOREIGN KEY към таблицата USER
createdDate	Date	Датата в момента на създаване.	Датата в момента на създаване на секцията от съответния потребител.

updatedAt	Date	Датата на последната редакция.	Датата на последната редакция на секцията.
participants[user_ids]	Array	N/A	Списък на потребителите, които участват в този борд.

Описание на категориите и отделите

Системата разполага с пет отделни категории секции за всеки борд, всяка, от които се състои от конкретен брой задачи:

- **OPEN** - секция на задачите от проекта, които предстои да бъдат започнати
- **READY FOR DEVELOPMENT** - секция на задачите от проекта, които са готови да разработване, но все още не са започнати (изискванията са определени)
- **IN DEVELOPMENT** – задачите, които в момента са в процес на разработка
- **UAT** – User acceptance testing – задачи, които са разработени, но чакат тестване от тестери.
- **DONE** - секция на задачите от проекта, които вече са записани в системата като готови

Категории за лицата, имащи отношение към системата:

o Standard user - потребител, който има достъп в системата

o Project manager – PM – има права да редактира проекти и бордове

Категории за задачите:

- Low priority - задача с нисък приоритет
- Medium priority - задача със среден приоритет
- High priority - задача с висок приоритет

Описание на съхранените процедури

Comments:

o Create-comment - създаване на коментар като POST заявка, взема се taskId (идентификационен номер на задачата, по която ще се извърши коментар), author(потребителят) - чрез REST API се изпраща заявка към сървъра и се създава коментар в базата

o Delete-comment – изтриване на коментар като DELETE заявка- чрез REST API се изпраща заявка към сървъра и се изтрива коментар в базата

o Update-comment – редактиране на коментар като POST заявка- чрез REST API се изпраща

заявка към сървъра и се редактира коментар в базата

Users:

o Login – вход за профила като POST заявка - чрез REST API се изпраща заявка към сървъра за влизане на потребител в системата, който се обръща към външната система за удостоверяване LDAP

o Logout – изход от профила като GET заявка - чрез REST API се изпраща заявка към сървъра за излизане на потребител от системата.

Task:

o Create-task – създаване на задача в секция като POST заявка, като се задава uniqueIdentifier (уникален идентификационен номер), title(заглавие), description(описание), estimatedTime (максимално време за приключване на задачата) и се взема owner (идентификационния номер на потребителя), също и assignee(потребител, който се избира от конкретен owner от падащото меню, като по подразбиране може да не бъде избран такъв), както и board (проекта, върху който ще се изпълнява задачата в съответната секция) - чрез REST API се изпраща заявка към

сървърът към съответен assignee и задачата се създава в базата
o Delete-task – изтриване на задача като DELETE заявка - чрез REST API се изпраща заявка към сървъра и се изтрива задача в базата
o Update-task – актуализиране на задача в секция като POST заявка - чрез REST API се изпраща заявка към сървъра и се редактира задачата в базата

Section:

o Create-section – създаване на секция в Board - а (Проекта) като POST заявка - чрез REST API се изпраща заявка към сървъра и се създава секция като част от проекта в базата
o Delete-section – изтриване на секция в Board - а (Проекта) като GET заявка - чрез REST API се изпраща заявка към сървъра и се изтрива секция като част от проекта в базата
o Update-section – актуализиране на секция в Board - а (Проекта) като POST заявка - чрез REST API се изпраща заявка към сървъра и се редактира секция като част от проекта в базата

Board:

o Create-board – създаване на проект като POST заявка, като се взема идентификационният номер на потребител и се задава заглавие/title на проекта - чрез REST API се изпраща заявка към сървъра и се създава проект в базата
o Read-board – визуализиране на проект като POST заявка - чрез REST API се изпраща заявка към сървъра и се визуализира проект в базата
o Update-board – актуализиране на борд като POST заявка- чрез REST API се изпраща заявка към сървъра и се извършва редакция на проекта в базата

Допълнение

За успешно стартиране на WEB приложението:

1. Клонирайте репото
2. Създайте Oracle user KENTRACK с парола KENTRACK.
3. npm install на front end модула.
4. Npm run dev на front end модула (чрез VITE)
5. Пуснете gradle build на backend и jooq модула.
6. Стартирайте Spring приложението (Liquibase автоматично ще създаде всички необходими таблици и връзки в базата от данни)
7. Отворете <http://localhost:3000/>

Продукция

За качване на продукцията се използва FAT JAR – пакет, който съдържа билднат всички необходими модули – Jooq класовете, backend spring модула и front end модула.

За целта е нагласен task в gradle – BOOT JAR, който пакетира всичко в един deployable jar.

Това се случва като се билдва първо front-end-а чрез VITE в един JS файл, който съдържа всички VUE компоненти и техните dependency-та. След това той се копира в статична папка в бекенда и чрез gradle се пакетират заедно като предварително са нагласени порта на сървъра да връща UA-а.

След като е получен артефактът, той трябва само да се изпълни като се използва java версия 11+: java -jar new-kentrack.jar, който ще пусне WEB апликацията в порт 8080 на съответната машина.