

ДОКУМЕНТАЦИЈА ЗА ПРОЕКТНА ЗАДАЧА ПО
ПРЕДМЕТОТ ИМПЛЕМЕНТАЦИЈА НА
СИСТЕМИ СО СЛОБОДЕН И ОТВОРЕН КОД

Име и презиме: Александар Крстески

Индекс: 186002

Наслов на проектната задача

**Разработка на FastAPI развојната рамка во
Python**

1. Опис на технологиите користени за изработка на проектната задача

1.1 Опис на структура на веб апликации

Секоја веб апликација се состои од две главни програмски компоненти:

- Frontend (делот од апликацијата кој го контролира приказот кон корисникот);
- Backend (делот од апликацијата кој комуницира со базата на податоци и извршува бизнис логика врз податоците).

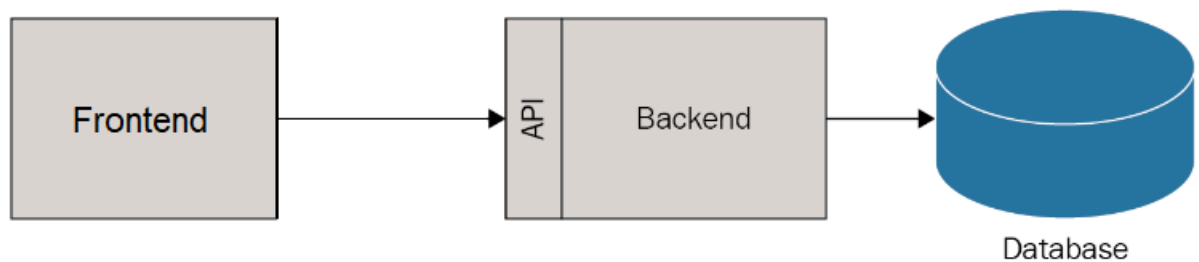
Целокупната функционалност која ја овозможува веб апликацијата се постигнува преку постојана комуникација помеѓу овие две компоненти.

Вообичаен тек на извршување на активности при користење на веб апликации е следниот:

- Корисникот стапува во интеракција со корисничкиот интерфејс на апликацијата, а со цел да испраќа барања за одредени услуги.
 - Контролата на приказот на корисничкиот интерфејс се врши од страна на Frontend делот од апликацијата. Корисничкиот интерфејс на корисникот му овозможува да испрати најразлични барања, како на пример барање за приказ на одредени ресурси, креирање нови ресурси, измена на постоечки ресурси, бришење на ресурси итн.
- Кога корисникот ќе испрати одредено барање, Frontend делот од апликацијата воспоставува комуникација со Backend делот, при што бара да се преземат одредени податоци од базата и да се изврши соодветна бизнис логика врз нив за да се задоволи барањето на корисникот.

- Кога Backend делот ќе заврши со процесирање на податоците, истите ги испраќа назад до Frontend делот, кој што генерира соодветен приказ до корисникот.

Шематски приказ на клиент-сервер архитектурата кај веб апликациите е прикажан на Слика 1 подолу:



Слика 1: Генерална архитектура на веб апликации

1.2 Опис на FastAPI како развојна рамка за креирање на Backend делот (серверскиот дел) на веб апликациите

Компонентата од Backend делот која “слуша” за барања испратени од Frontend делот (односно од корисникот) се нарекува Application Programming Interface (API). Со други зборови, прв посредник при остварување на комуникација помеѓу Frontend и Backend е API-то.

Постојат најразлични стандарди кои налагаат запазување на одредена архитектура при развој на API (на пример REST, GraphQL, SOAP итн).

Во оваа проектна задача ја употребив FastAPI open-source развојната рамка во програмскиот јазик Python, која овозможува креирање на Backend делот од веб апликациите, со API што го следи REST стандардот (уште наречено и RESTful API).

При креирање на API, REST стандардот налага да се следат долунаведените чекори:

- Запазување на клиент – сервер архитектурата, каде корисникот (Frontend – от) го претставува клиентот, додека Backend – от го претставува серверот;
- Користење на HTTP/HTTPS протоколот за комуникација помеѓу Frontend и Backend, со специфицирање на HTTP метод соодветен на акциите кои корисникот бара да бидат извршени од страна на апликацијата;
- Дефинирање “рути”, односно уникатни патеки на кои клиентот (Frontend) ќе праќа барања до серверот (Backend), каде секое барање ќе има свој “тип” (тука наречен “метод”) соодветен на акцијата што треба да се изврши од страна на серверот. Секое барање може да содржи и дополнителни податоци пратени од страна на клиентот до серверот, специфицирани во JSON формат.

Но, FastAPI не е лимитирана исклучиво на развој на API. Таа е интегрирана да работи заедно со библиотеката Pydantic, која нуди голем број функции за манипулација на податоци (со особен акцент на валидацијата на податоци, како оние кои доаѓаат од формите). Поспецифично, при развој на Backend апликации со FastAPI, библиотеката Pydantic се користи за креирање на компоненти уште наречени и модели, кои претставуваат објектни репрезентации на ентитетите во базата на податоци. Овие модели овозможуваат манипулација со податоците во базата (додавање, читање, изменување и бришење на податоци), како и извршување на сервисна логика врз податоците.

Исклучителните перформанси на Backend апликациите развиени со користење на FastAPI развојната рамка во голема мера се должат на употребата на библиотеката Pydantic, која е високо оптимизирана во поглед на перформансите.

FastAPI како open-source развојна рамка за креирање на Backend апликации во Python нуди:

- Креирање Backend апликации со највисоки перформанси во однос на сите останати рамки за развој на Backend апликации во Python;
- Брз и конзистентен развој на Backend апликации со значително поретка појава на грешки во кодот, бидејќи:
 - вклучува готови функционалности за речиси се што е потребно при развој на Backend апликации;
 - вклучува функционалност за автоматско дополнување на код, со што дополнително се убрзува развојот на Backend апликации и се намалува бројот на грешки.
- Брзо запознавање со различните функционалности кои ги нуди доколку се има познавање од Python, бидејќи:
 - FastAPI претставува склоп од функции напишани во јазикот Python;
 - Нуди кратка, но доволно прецизна документација.

1.3 Опис на Angular како развојна рамка во програмскиот јазик TypeScript за развој на Frontend апликации

За развој на Frontend делот во оваа проектна задача ја искористив развојната рамка Angular.

Angular е open-source component-базирана развојна рамка во програмскиот јазик TypeScript, чија намена е развој на скалабилни клиентски (Frontend) апликации.

Основните градебени единици на секоја Angular клиентска апликација се наречени компоненти. Секоја компонента во Angular ги вклучува сите потребни елементи, стилови и функционалности потребни за приказот и однесувањето на различните составни делови (целини) на корисничкиот интерфејс кој му се презентира на корисникот.

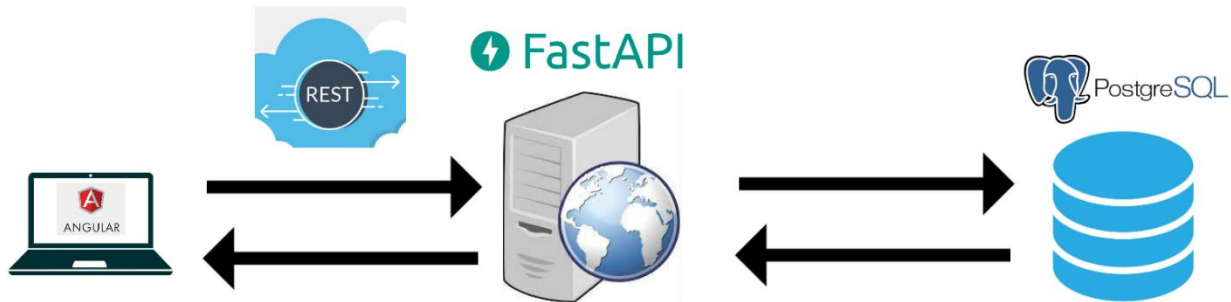
1.4 Опис на PostgreSQL системот за менаџирање на релациони бази на податоци

За чување на податоците потребни за изработка на оваа проектна задача употребив релациона SQL база на податоци, со користење на PostgreSQL како open-source систем за менаџирање на релациони бази на податоци (Relational Database Management System – RDBMS).

PostgreSQL е еден од најпопуларните системи за менаџирање на релациони бази на податоци, бидејќи:

- овозможува користење на “надградена” верзија од јазикот SQL која нуди дополнителни функционалности за обработка на податоците во базата;
- поддржува голем број типови на податоци и нуди можност за креирање кориснички типови на податоци;
- се базира врз така наречени “ACID” трансакции (Atomic, Consistent, Isolated, Durable) кои гарантираат надежно, конзистентно и стабилно извршување операции врз базата на податоци, дури и кога се побарува паралелно извршување на повеќе операции;
- има голем број extensions и plugins за надградување на функционалноста.

Архитектурата на апликацијата што ја развив за потребите на оваа проектна задача е прикажана на Слика 2 подолу:



Слика 2: Архитектура на апликацијата

2. Опис на идејата

За потребите на оваа проектна задача, развив веб апликација за менаџирање и резервација на ресурси.

Оваа апликација не е врзана стриктно со одредени ресурси (како на пример, компјутерска опрема), туку претставува еден генерализиран систем за менаџирање и резервација на ресурси кој овозможува работа со најразлични видови ресурси.

Апликацијата ја развив со цел да може било која компанија да ја користи за да ги менаџира и доделува на користење своите ресурси. Имено, оваа апликација овозможува ADMINISTRATOR да додаде најразлични ресурси кои е потребно да се менаџираат и распределуваат помеѓу корисници.

Администраторот може во секој момент да додава ресурси, да врши измена на карактеристиките на ресурсите, да брише ресурси итн. Дополнително, овозможено е администраторот да може да ги групира ресурсите во категории, при што може да се додаваат и поткатегории. Односно, администраторот може да изгради едно дрво (хиерархија) од

категории, при што ресурсите може да припаѓаат исклучиво на категории кои немаат поткатегории (листови во дрвото од категории).

Секоја категорија на ресурси може да има свои својства, различни од својствата на останатите категории. Поткатегориите на дадена категорија ги наследуваат нејзините својства, при што кај нив може да се додаваат и дополнителни својства.

Секој ресурс е опишан со својствата на категоријата на која ѝ припаѓа. При креирање на ресурс потребно е да се специфицираат вредности за овие својства.

Администраторот не може да резервира ресурси.

Корисниците кои побаруваат резервација на ресурси не може да додаваат ресурси, не може да ги изменуваат карактеристиките на постоечките ресурси, ниту пак може да бришат ресурси.

Секој корисник и администратор може да има увид во резервациите што биле направени до тековниот момент.

За користење на системот секој корисник треба да биде автентизиран. Автентикацијата е реализирана со користење на JWT (JSON Web Token), што претставува безбеден начин на пренос на податоци помеѓу клиент и сервер во JSON формат (во JWT токен се чуваат информациите за автентикација на корисникот, преку кои се проверува дали корисникот е автентизиран). Процесот на автентикација се одвива така што кога корисникот ќе се регистрира/најави, серверот генерира JWT токен кој ги содржи информациите за автентикација на корисникот и го испраќа токенот до корисникот (клиентот). Токенот се чува кај клиентот одредено време, така што во секое праќање на ново барање до серверот се праќа и токенот. Штом ќе се испрати токенот до серверот, серверот валидира дали тој корисник постои во табелата со регистрирани корисници во базата, и ако тоа е случај, му дозволува

пристап на корисникот до сите ресурси за кои е потребно да се биде автентизиран кога се побаруваат.

Исто така, бидејќи има два типови на корисници на апликацијата (администратор и корисник кој резервира ресурси) кои имаат различни права при употреба на апликацијата, имплементирана е авторизација.

3. Опис на имплементацијата

3.1 Опис на табелите во базата на податоци

- За овозможување на функционалностите опишани во претходната точка, во базата на податоци чувам информации за следните ентитети: Корисник (User), Ресурс (Resource), Категорија (Category) и Резервација (Reservation).

За секој корисник се чуваат следните својства:

- id: Уникатен идентификатор на корисникот
- username: Корисничко име
- email: Електронска пошта
- role: Улога на корисникот (може да биде администратор или корисник за резервација на ресурси)
- password: Лозинка за најава

За секоја категорија се чуваат следните својства:

- id: Уникатен идентификатор на категоријата
- name: Име на категоријата
- description: Краток опис на категоријата
- parent: Име на родителската категорија (доколку таа постои)
- properties: Својства на категоријата

За секој ресурс се чуваат следните својства:

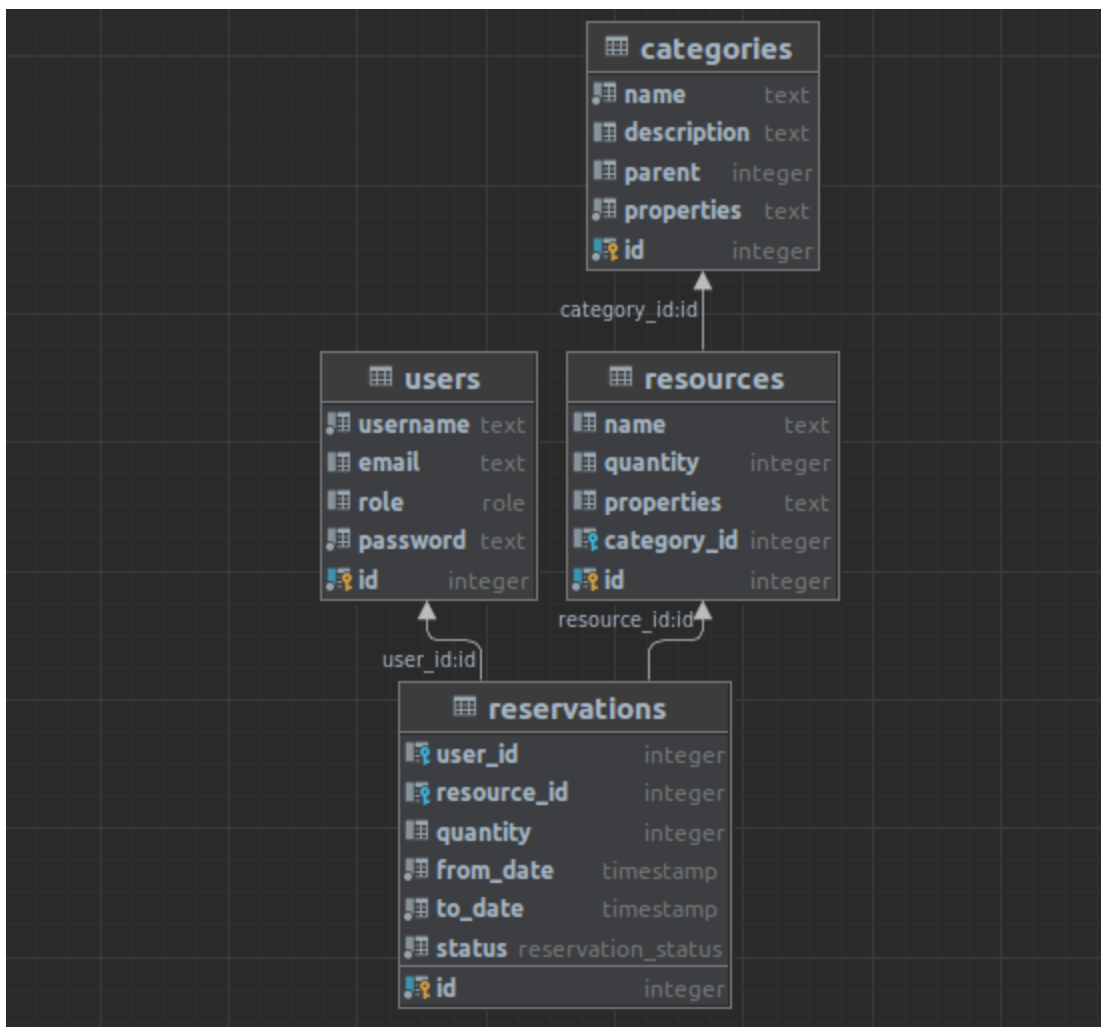
- id: Уникатен идентификатор на ресурсот
- name: Име на ресурсот

- quantity: Број на достапни единици од ресурсот
- properties: Својства на ресурсот (соодветни на категоријата на која тој ѝ припаѓа)
- category_id: Уникатен идентификатор на категоријата на која ресурсот ѝ припаѓа

За секоја резервација се чуваат следните својства:

- id: Уникатен идентификатор на резервацијата
- user_id: Уникатен идентификатор на корисникот кој ја направил резервацијата
- resource_id: Уникатен идентификатор на ресурсот кој е резервиран
- quantity: Број на единици од ресурсот што се резервираат
- from_date: Почеток на важење на резервацијата
- to_date: Завршеток на важење на резервацијата
- status: Статус на резервацијата (може да биде PENDING, IN PROGRESS или DONE)

Дијаграм со шемите на табелите во базата на податоци и релациите помеѓу нив е прикажан на Слика 3 подолу:



Слика 3: Дијаграм со табелите во базата на податоци и релациите помеѓу нив

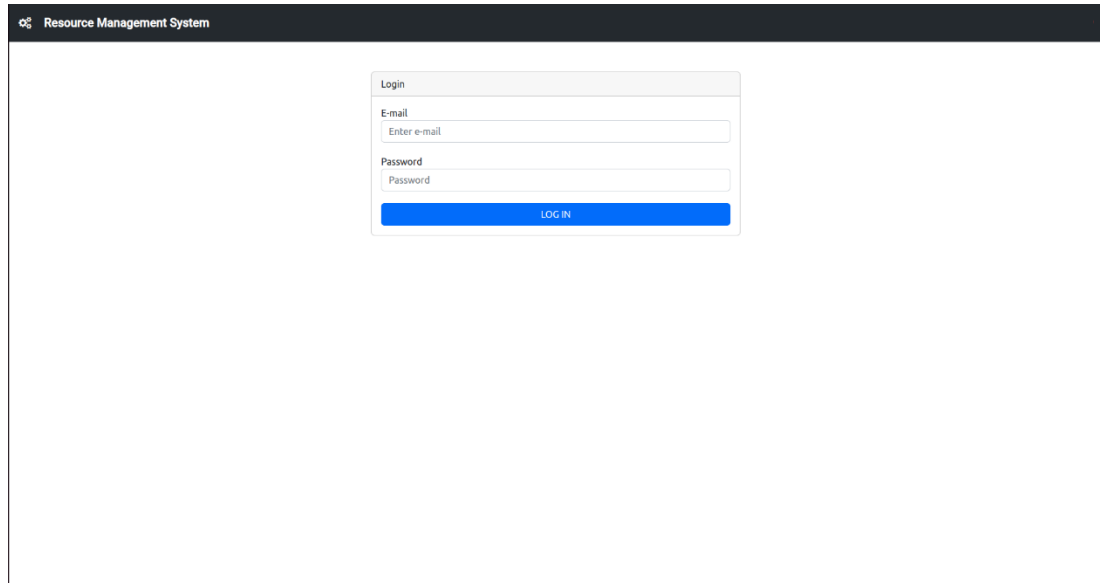
3.2 Опис на корисничкиот интерфејс

Во оваа апликација, како што беше опишано погоре, постојат два типови на корисници: администратор и корисник кој побарува резервација на ресурси. Администраторот и корисникот кој резервира ресурси имаат различни права за пристап до одделните компоненти на апликацијата. Со други зборови, во оваа апликација постојат два типови на кориснички интерфејс:

- Кориснички интерфејс за администратор
- Кориснички интерфејс за корисник кој резервира ресурси

3.2.1 Опис на корисничкиот интерфејс за администратори

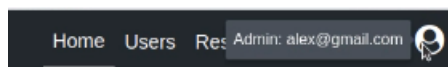
За да може да се користи оваа апликација од страна на било кој корисник, истиот најпрво мора да се автентичира. При првичен пристап до апликацијата, доколку корисникот не е автентичиран, се прикажува интерфејсот на Слика 5 подолу:



Слика 5

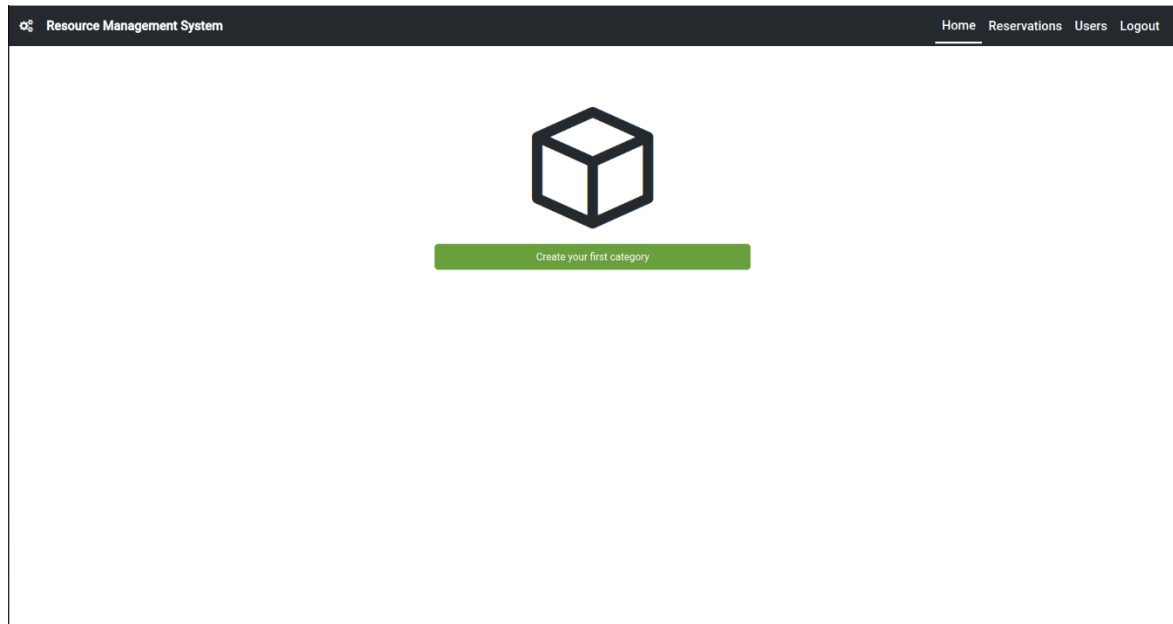
Во оваа првична верзија на апликацијата, во базата постои еден од напред креиран администратор, кој мора задолжително да се најави на почетокот.

По најавувањето на корисник, во десниот дел на навигацискиот бар горе се појавува икона за профил. Кога ќе се направи hover на оваа икона, се прикажува текст од облик ROLE: email за најавениот корисник (пример ADMIN: alex@gmail.com). Тоа може да се види на Слика 6 подолу:



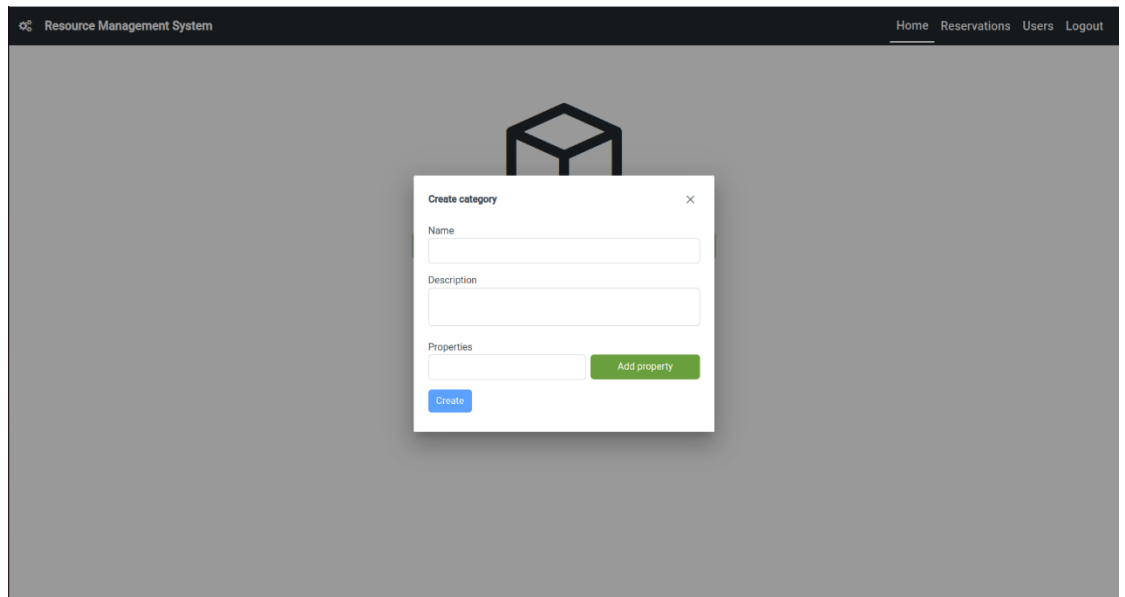
Слика 6

Кога на системот ќе се најави администратор, доколку не постои ниту една категорија на ресурси (а во тој случај нема да постои и ниту еден ресурс) ќе се прикаже интерфејсот на Слика 7:



Слика 7

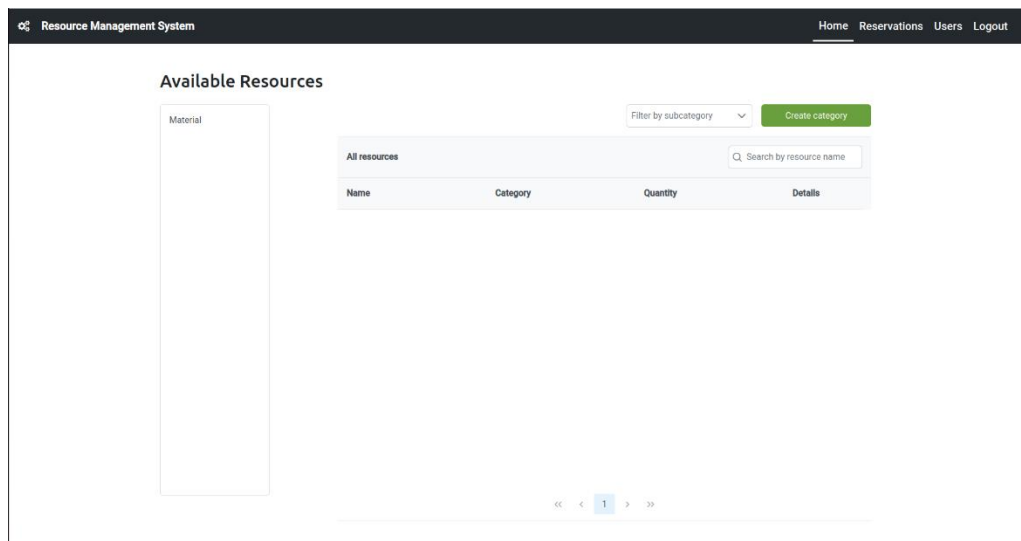
Тука, администраторот може да притисне на копчето “Create your first category” за да креира категорија, при што ќе се прикаже форма за пополнување информации за категорија како на Слика 8:



Слика 8

Тука администраторот задолжително треба да внесе име на категоријата и имиња на својства на категоријата (Properties), додека описот на категоријата (Description) е опционален.

Кога во системот ќе постои барем една категорија на ресурси и администраторот е најавен, ќе се прикаже интерфејсот на Слика 9:



Слика 9

Како што беше опишано погоре, може да се додаваат и поткатегории на дадена категорија. Од левата страна во овој приказ се прикажуваат исклучиво категориите кои се деца на тековно избраната категорија (така што, кога ќе се додадат деца на дадена категорија, администраторот може да притисне на таа категорија за да се прикажат нејзините деца).

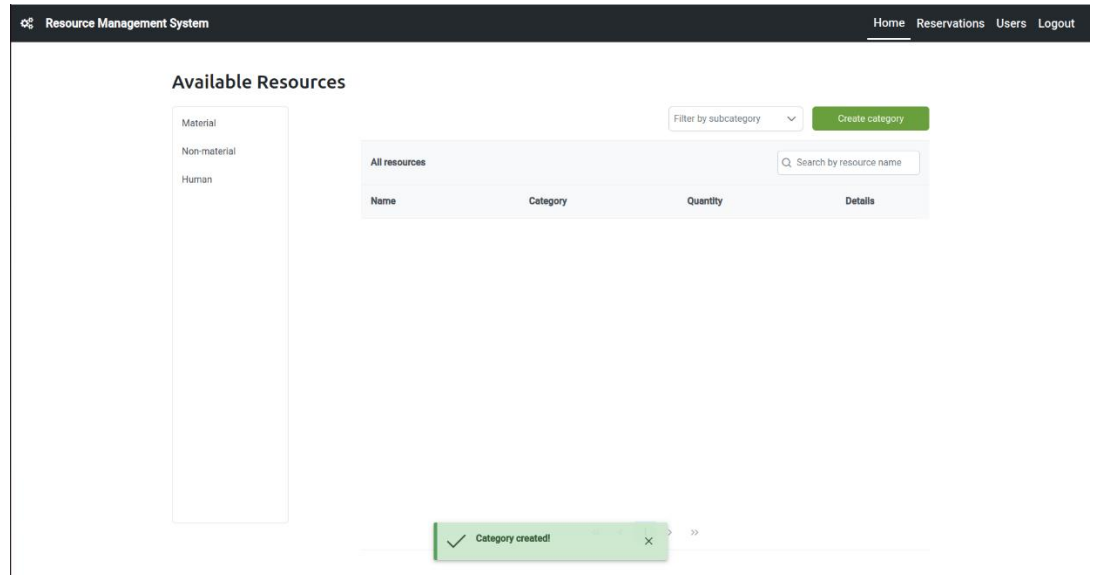
Од десна страна во овој интерфејс е прикажана табела од сите ресурси на кои заеднички предок им е категоријата која е тековно избрана (кога се наоѓаме на најгорното ниво во дрвото од категории, се прикажуваат сите ресурси креирани во системот).

Исто така, над табелата со ресурси има drop-down листа за филтрирање на ресурсите кои се прикажуваат во табелата според поткатегорија на тековно избраната категорија. Тука, може да се изберат само поткатегориите “листови” (оние кои немаат деца категории), бидејќи за да се добие приказ на сите ресурси кои припаѓаат во поткатегорија која има деца, може само да се влезе во категоријата со притискање на истата од левата страна на приказот.

Дополнително, администраторот може и да филтрира приказ на ресурси според нивното име, со внесување на дел од нивното име во полето чиј placeholder е “Search by resource name”.

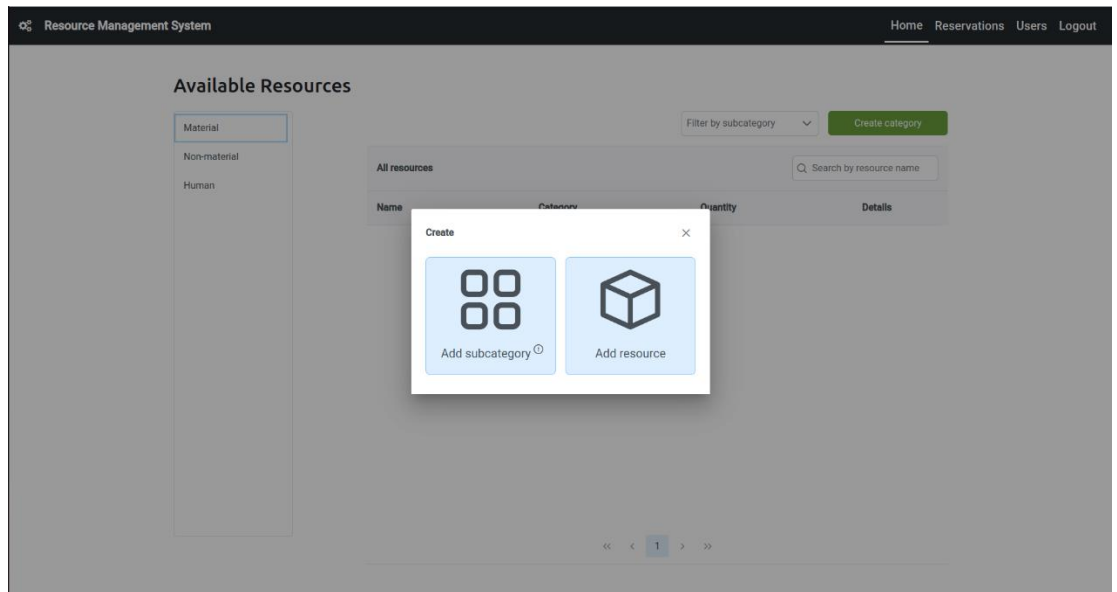
Копчето “Create category” служи за да му овозможи на администраторот да креира поткатегорија на тековно избраната категорија (оваа поткатегорија ќе биде “sibling” на категориите кои тековно се гледаат во левиот дел на интерфејсот). Доколку администраторот притисне на копчето “Create category” ќе се прикаже истата форма за креирање категорија која е прикажана на Слика 8.

Кога администраторот ќе додаде неколку “siblings” категории кои се деца на тековно избраната категорија, ќе биде прикажан интерфејсот на Слика 10:



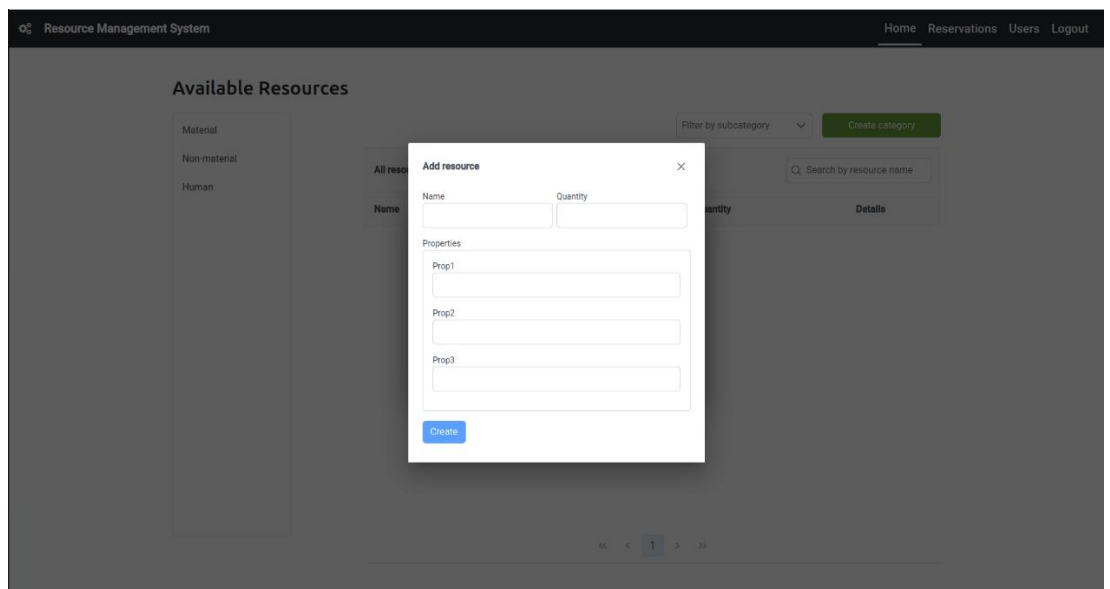
Слика 10

Доколку администраторот сака да додаде ресурс или поткатегорија на категорија која тековно нема поткатегории, потребно е само да притисне на името на таа категорија во левиот дел на приказот (категириите се копчиња). Кога администраторот ќе го стори тоа, ќе биде прикажан интерфејсот на Слика 11:



Слика 11

За додавање ресурс, администраторот треба да притисне на копчето “Add resource”. Кога администраторот ќе го стори тоа, ќе биде прикажана форма за внес на информации за ресурсот кој се креира како на Слика 12:

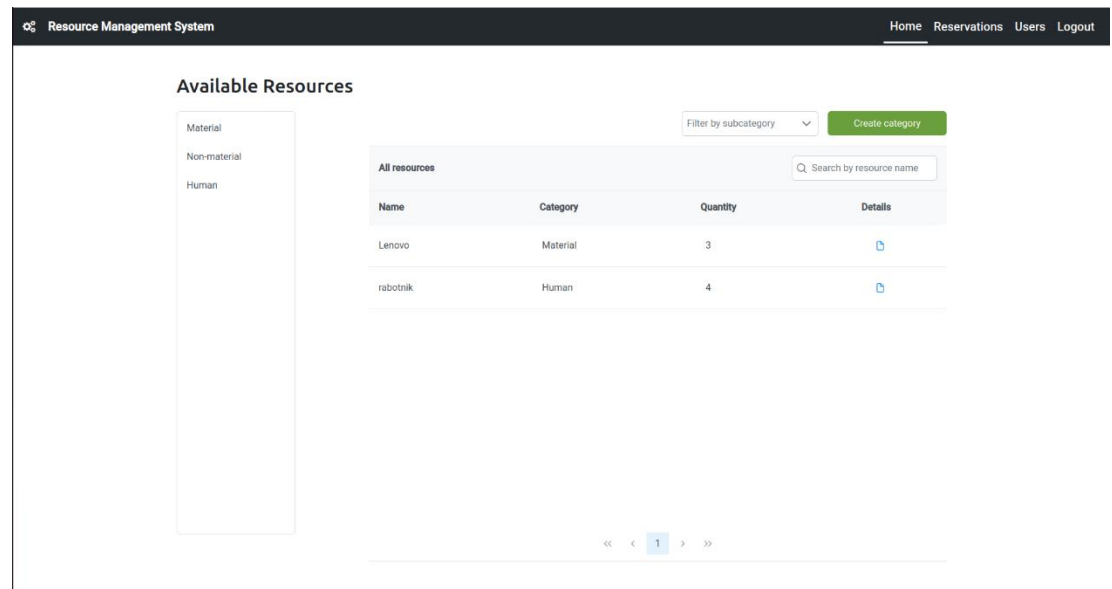


Слика 12

За секој ресурс, администраторот треба да специфицира име на ресурсот, број на достапни единици од ресурсот, како и вредности за сите својства кои ѝ припаѓаат на неговата

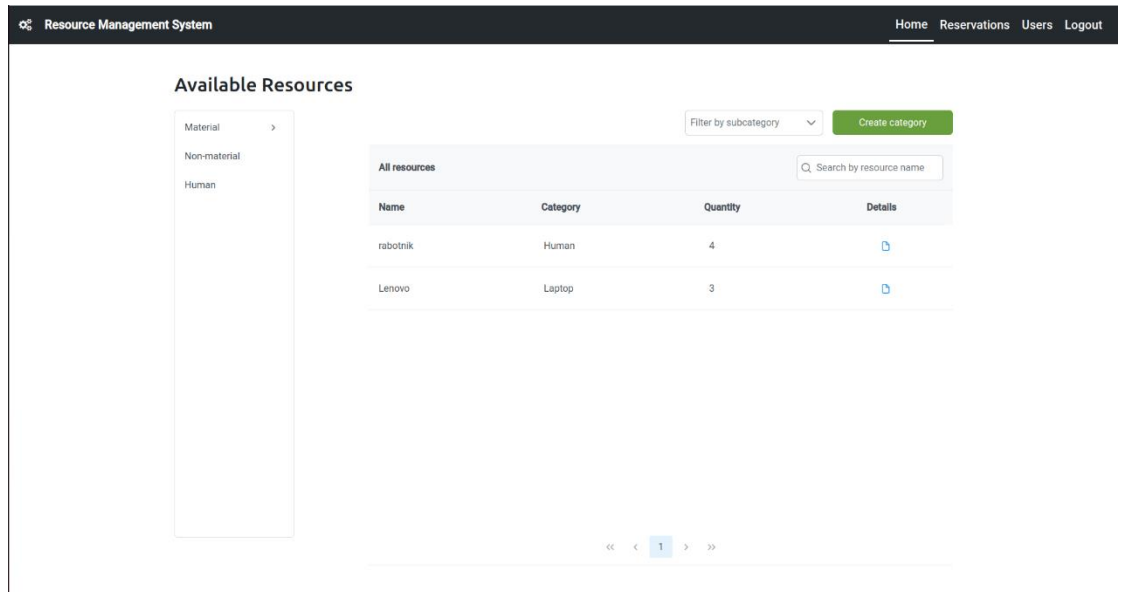
категорија (својствата на категоријата ги опишуваат ресурсите кои ѝ припаѓаат).

Кога администраторот ќе додаде ресурси во системот, ќе биде прикажан интерфејсот на Слика 13:



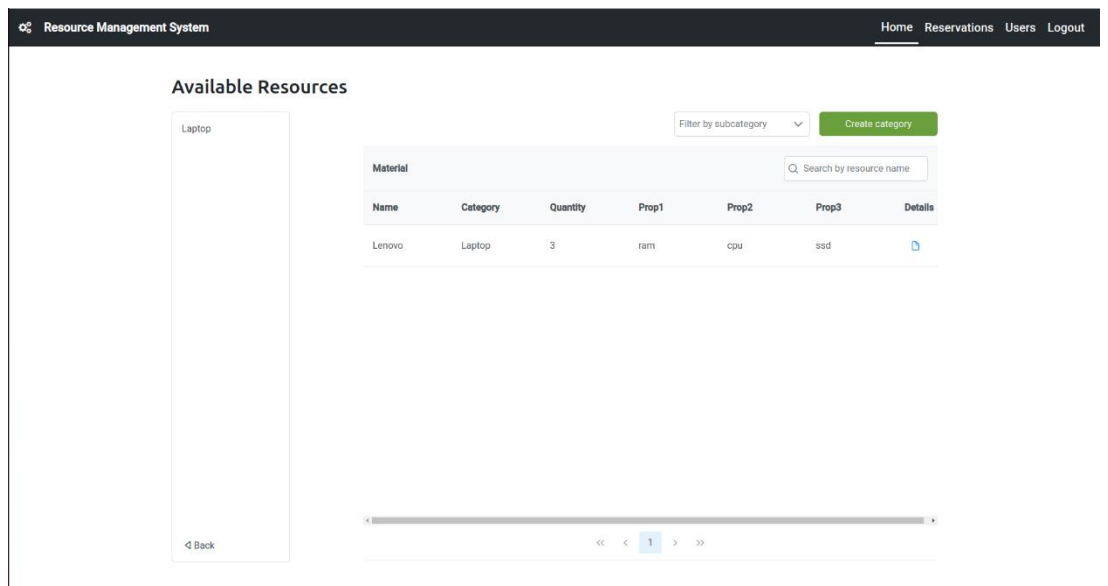
Слика 13

Доколку администраторот во интерфејсот прикажан на Слика 11 избере да додаде поткатегорија, се прикажува формата за додавање категорија од Слика 8. Кога администраторот ќе додаде поткатегории на некоја од категориите, десно од името на таа категорија ќе биде прикажана стрелка (тоа означува дека кога ќе се притисне на таа категорија, ќе може да се видат нејзините поткатегории). По додавање на поткатегории во некои од категориите, интерфејсот ќе изгледа како на Слика 14:



Слика 14

При интерфејс како на Слика 14, доколку администраторот избере да влезе во категоријата “Material”, ќе биде прикажан интерфејс како на Слика 15:



Слика 15

Кога тековно е избрана дадена категорија во левиот дел на приказот, се прикажуваат сите ресурси кои им припаѓаат на “листови” категории во поддрвото од категории чиј корен е

тековно избраната категорија. Поради тоа што овие ресурси може да имаат различни множества од својства, во табелата се прикажуваат само оние својства кои се заеднички за сите ресурси (а тоа се својствата на категоријата која тековно е избрана). Доколку администраторот сака да ги види останатите својства на ресурсот, потребно е да притисне на копчето во колоната “Details”. Кога администраторот ќе притисне на копчето “Details”, ќе биде прикажана формата за додавање ресурс прикажана на Слика 12, при што во неа соодветно ќе бидат пополнети полињата со постоечките вредности на својствата на ресурсот. Тоа е прикажано на Слика 16 подолу:

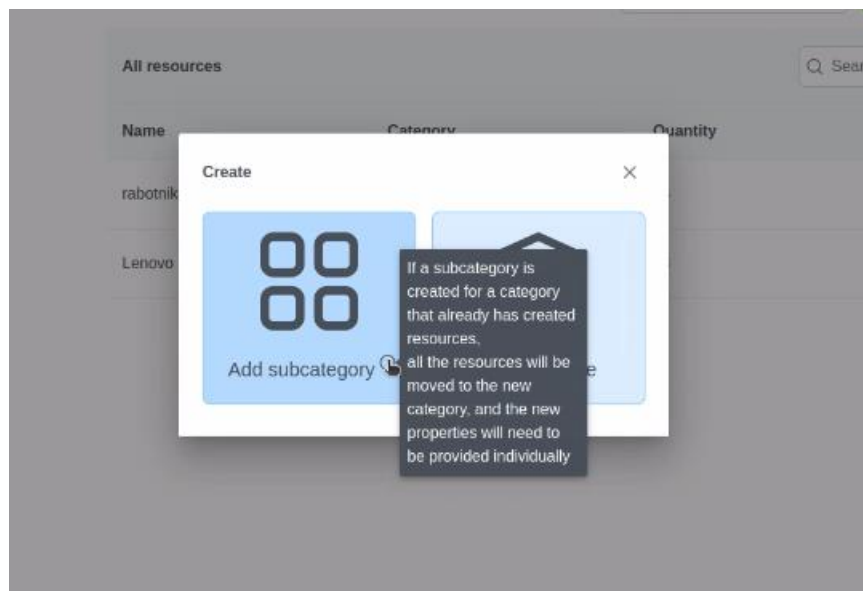
The screenshot displays the 'Resource Management System' interface. A modal window titled 'Update resource' is open, showing a form for updating a resource. The form includes fields for 'Name' (containing 'Lenovo') and 'Quantity' (containing '3'). Below these are five 'Properties' fields: 'Prop1' (containing 'ram'), 'Prop2' (containing 'cpu'), 'Prop3' (containing 'ssd'), 'Prop4' (empty), and 'Prop5' (empty). At the bottom of the modal, there is a 'Category' dropdown menu set to 'Laptop', a link 'Load all available categories', and two buttons: 'Delete' (red) and 'Update' (blue). The background shows a table of 'Available Resources' with columns for 'Material', 'Name', and 'Details'. The 'Details' column for the 'Lenovo' resource has a blue circular icon with a plus sign, which is the 'Details' button mentioned in the text.

Слика 16

Притоа, администраторот може да ги измени својствата на ресурсот и да притисне на копчето “Update”, а може и да го избрише ресурсот со притискање на копчето “Delete”.

Забелешка: Во овој случај, својствата “Prop4” и “Prop5” не се пополнети со вредности. Тоа се случува кога во дадена категорија најпрво ќе се додадат ресурси, а потоа ќе се додаде поткатегорија. Имено, доколку категоријата “Material” е

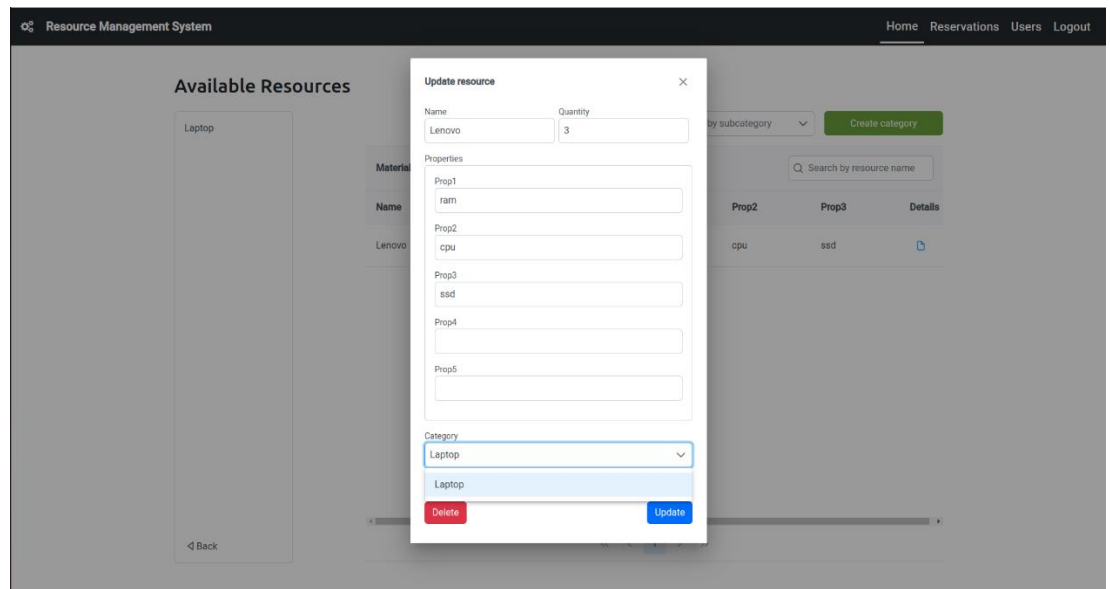
опишана само со својствата “Prop1”, “Prop2” и “Prop3” и ресурсот “Lenovo” прикажан на Слика 15 бил креиран во категоријата “Material” пред да биде креирана нејзината поткатегорија “Laptop” (која ја опишуваат две дополнителни својства “Prop4” и “Prop5”), тогаш ресурсот “Lenovo” ќе мора да биде преместен во категоријата “Laptop” (бидејќи категоријата “Material” веќе нема да биде “лист” категорија). Но, во овој случај, по неговото преместување во категоријата “Laptop”, тој нема да има вредности за својствата на категоријата “Laptop” (тоа се “Prop4” и “Prop5”). Поради тоа, во овој случај, администраторот ќе мора да притисне на копчето за приказ на деталите од ресурсот, каде ќе може да внесе вредности за својствата “Prop4” и “Prop5”. Поради овој случај, десно од текстот “Add subcategory” на Слика 11 стои извичник за информации, како што е прикажано на Слика 17 подолу:



Слика 17

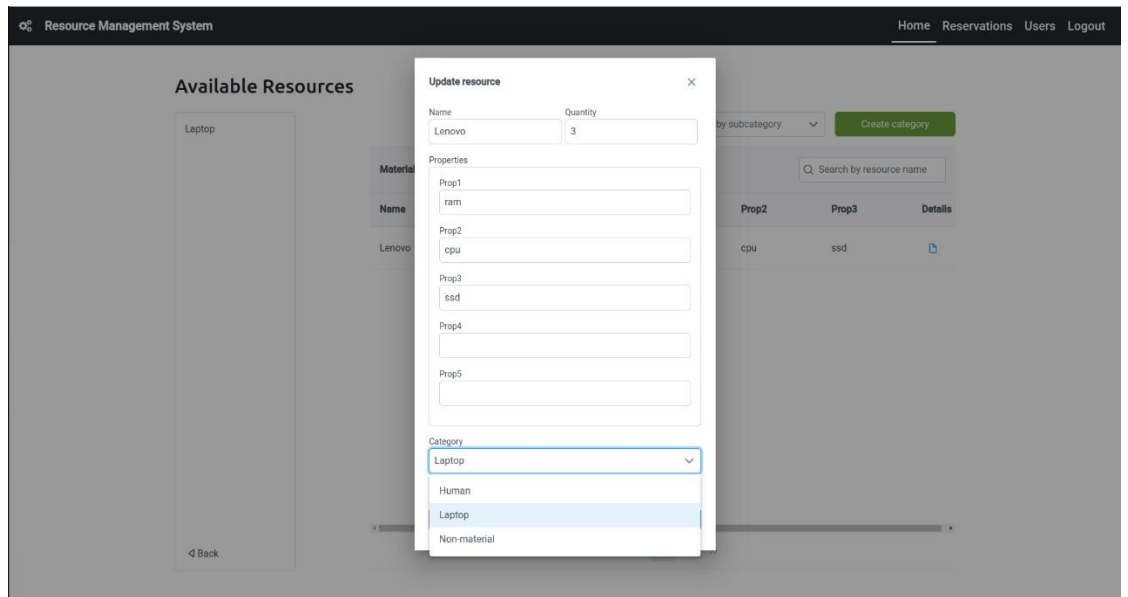
Дополнително, во приказот на деталите за ресурсот (Слика 16), администраторот може да ја промени категоријата на која ѝ припаѓа ресурсот. За почеток, се прикажуваат само “лист” категориите кои претставуваат “siblings” на категоријата на

која ресурсот ѝ припаѓа моментално, како што може да се види на Слика 18 подолу:



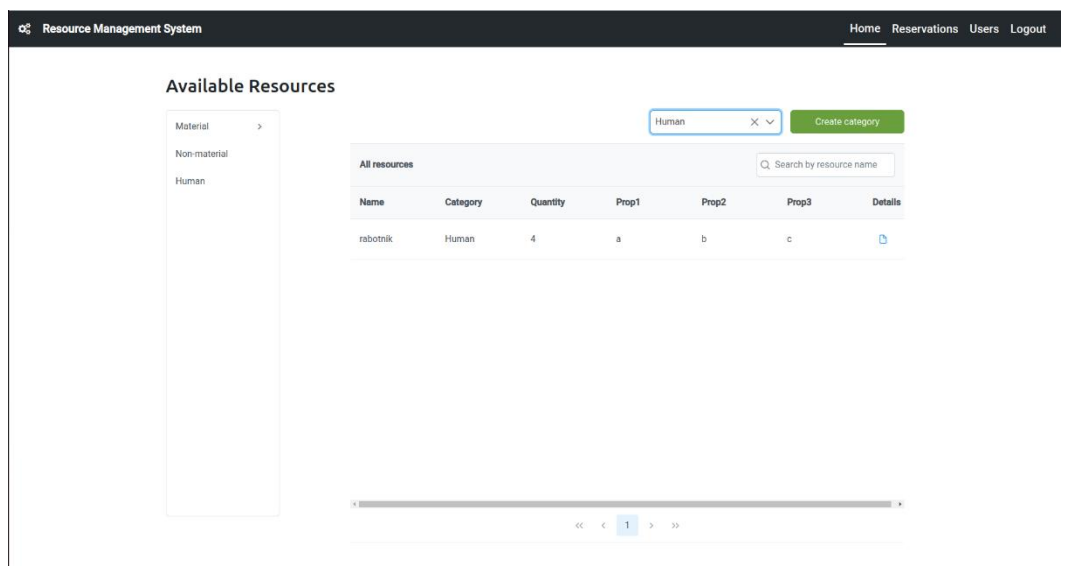
Слика 18

Доколку администраторот сака да го премести ресурсот во “лист” категорија која не е “sibling” на категоријата на која моментално ѝ припаѓа ресурсот, тогаш најпрво администраторот мора да притисне на копчето “Load all available categories”, прикажано на Слика 16. Кога администраторот ќе го стори тоа, може да избере нова категорија за ресурсот од листата на сите “лист” категории во дрвото од категории, како што е прикажано на Слика 19 подолу:



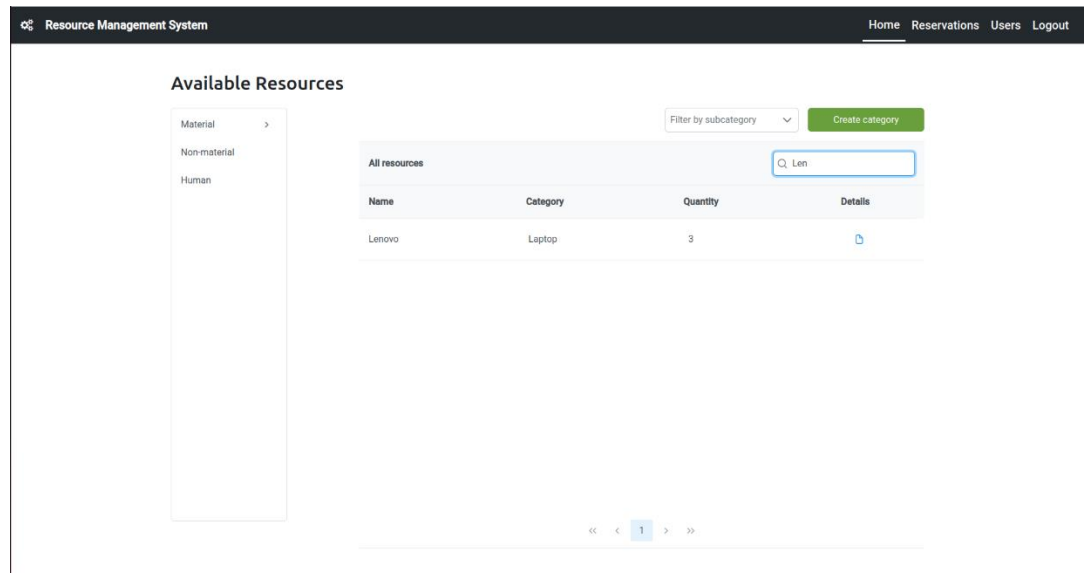
Слика 19

Доколку администраторот сака да ги исфилтрира ресурсите прикажани во табелата според поткатегорија “лист”, тоа може да го стори со избор на поткатегоријата од drop-down листата лево од копчето за креирање нова категорија. По извршено филтрирање според поткатегорија, интерфејсот изгледа како на Слика 20 подолу:



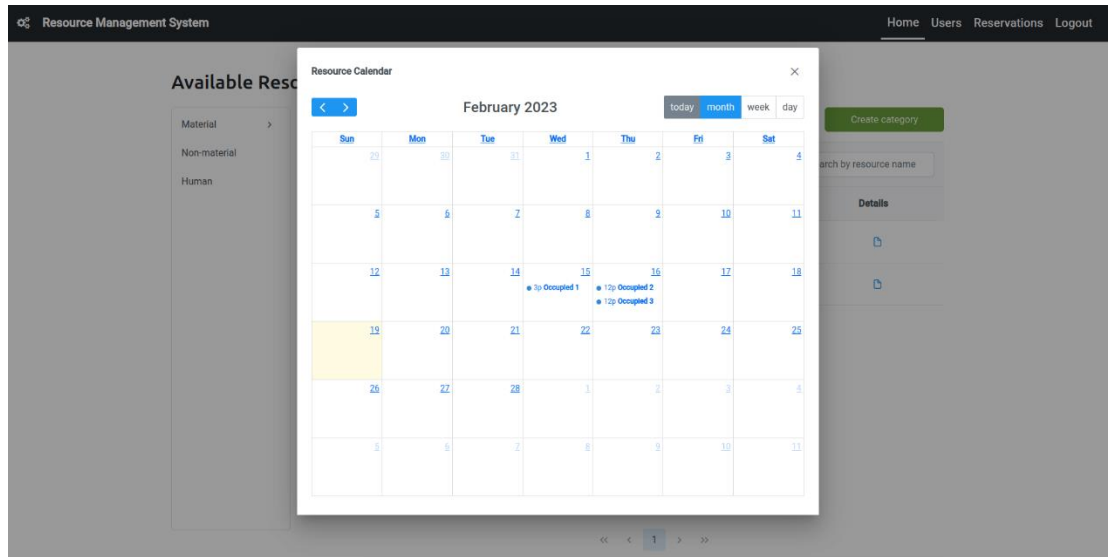
Слика 20

Од друга страна, доколку администраторот сака да филтрира приказ на ресурси во табелата според име на ресурсот, може да внесе дел од името на ресурсот во полето чиј placeholder е “Search by resource name”. По извршено филтрирање на ресурси според име, интерфејсот изгледа како на Слика 21 подолу:



Слика 21

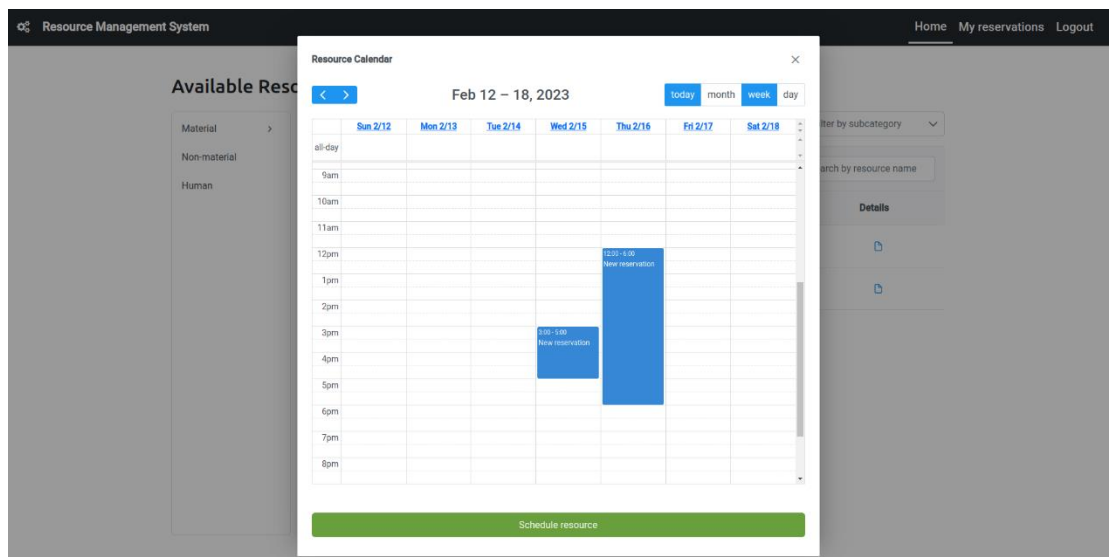
Доколку администраторот сака да види календар со обележани резервации за даден ресурс, потребно е да притисне било каде во редицата за тој ресурс од табелата со ресурси (целата редица е копче). Кога администраторот ќе го стори тоа, се прикажува интерфејс како на Слика 22 подолу:



Слика 22

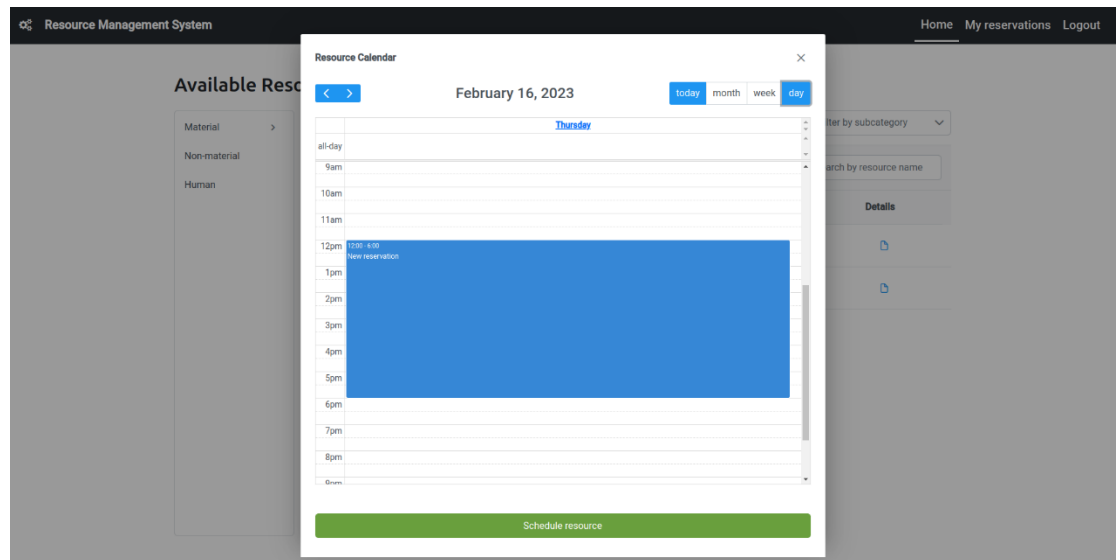
Дополнително, во календарот има опции за приказ на резервациите по ден, недела и месец, како што се гледа во горниот десен агол на Слика 22 (овој изглед на календарот е за приказ на резервации по месец).

Доколку администраторот избере да ги прикаже резервациите на календарот по недела, ќе се прикаже интерфејс како на Слика 23:



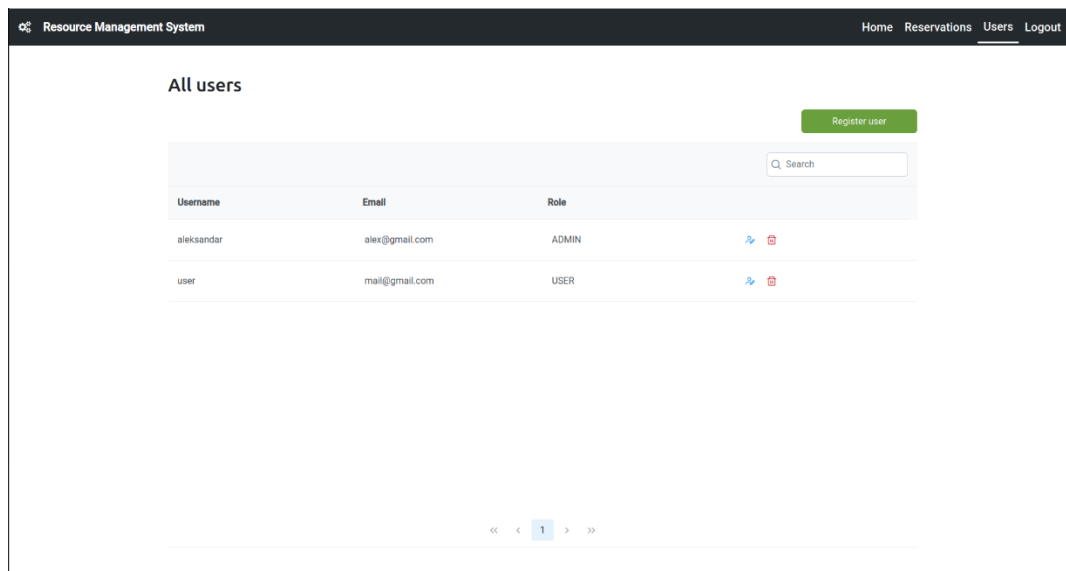
Слика 23

Додека пак, доколку администраторот избере да ги прикаже резервациите на ниво на ден, ќе се прикаже интерфејсот на Слика 24:



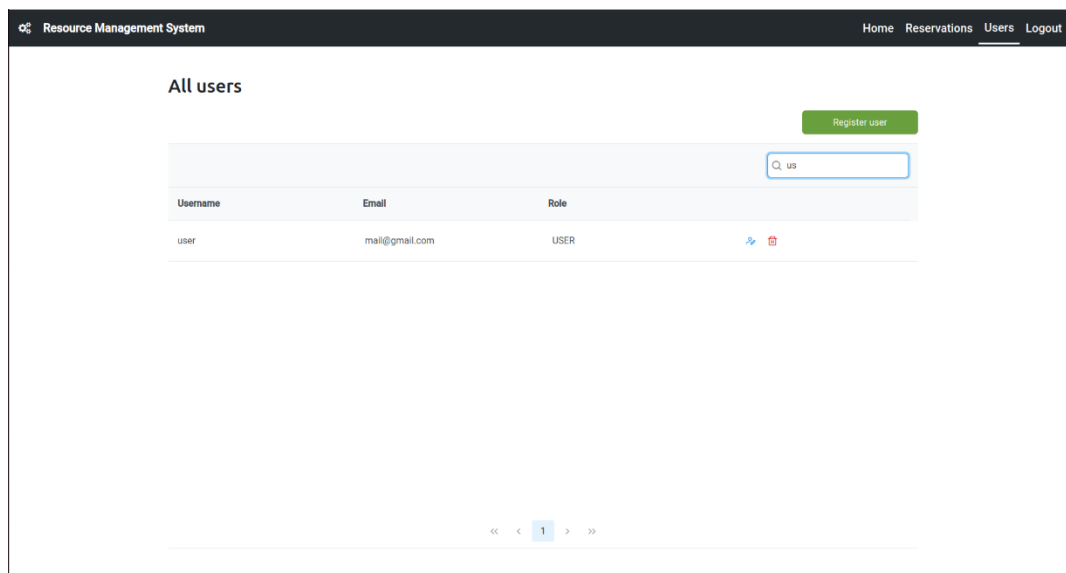
Слика 24

Доколку администраторот сака да види информации за сите корисници регистрирани на системот, тоа може да го стори со притискање на копчето "Users" горе десно во навигацискиот бар. Листата од регистрирани корисници се прикажува како на Слика 25 подолу:



Слика 25

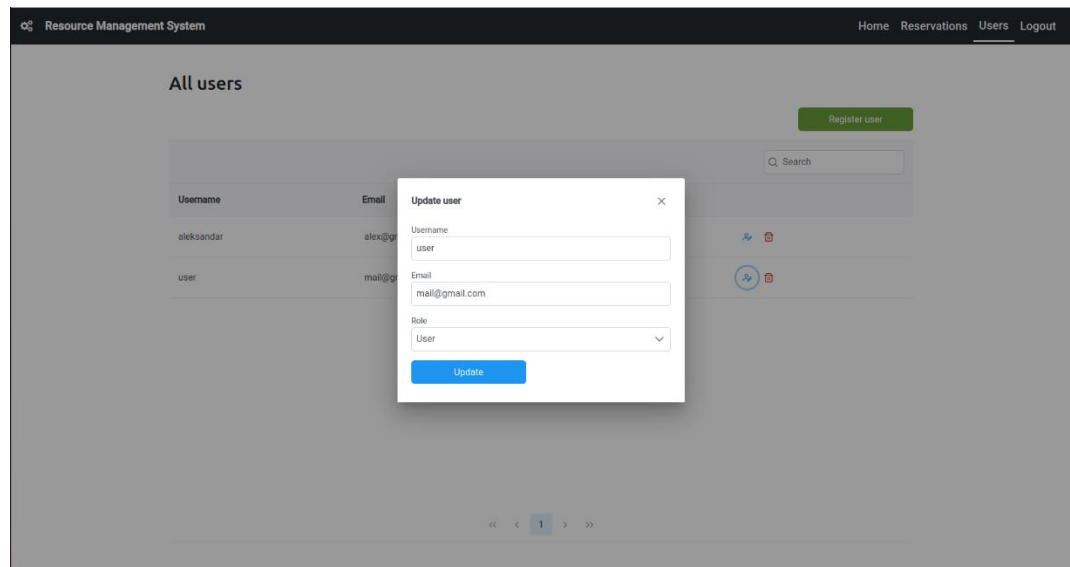
Администраторот може да пребарува корисници според username внесено во полето “Search”. При вакво пребарување, интерфејсот што се прикажува изгледа како на Слика 26 подолу:



Слика 26

Администраторот има право да врши промена на информации за корисници, со притискање на синото копче

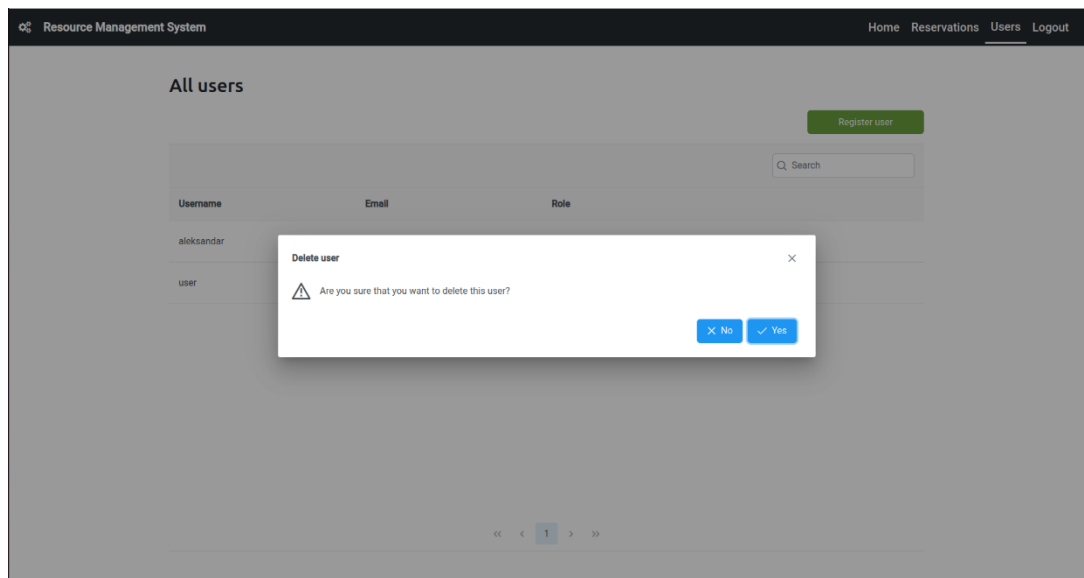
десно од информациите за корисникот. Кога ова копче ќе биде притиснато, се прикажува форма со полиња за специфицирање на информации за корисникот, која е веќе пополнета со тековните информации за корисникот. Тоа може да се види на Слика 27 подолу:



Слика 27

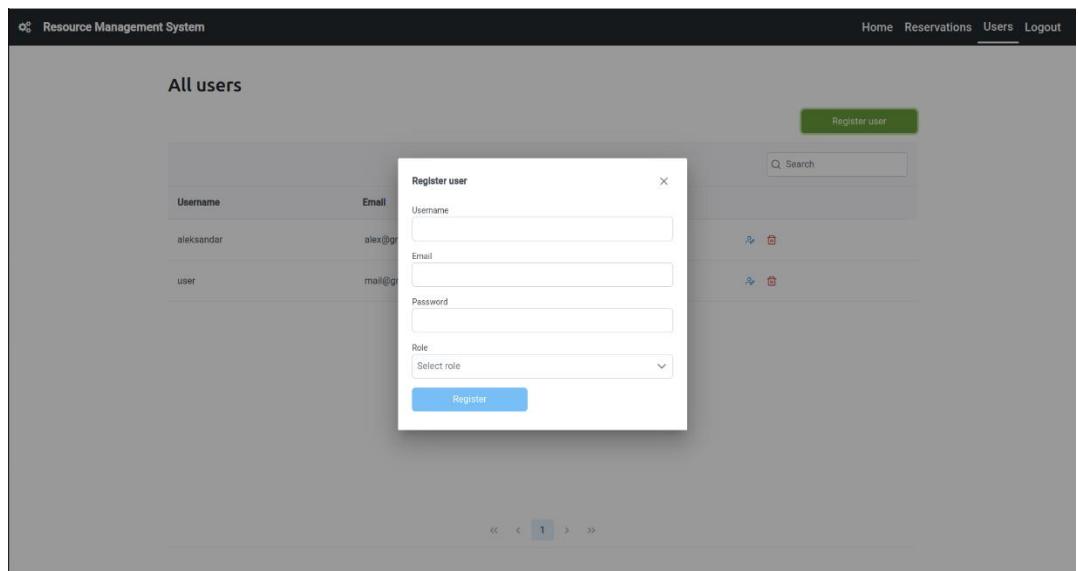
Забелешка: Администраторот може да ја промени улогата на секој од корисниците, но не и лозинката.

Администраторот има право и да избрише даден корисник, со притискање на црвеното копче десно од информациите за корисникот прикажани на интерфејсот на Слика 26. По притискање на ова копче, се појавува дополнителен prompt кој бара од администраторот да потврди дека сака да го избрише корисникот (со цел да се избегне ненамерно бришење на корисници). Изгледот на овој prompt е прикажан на Слика 28 подолу:



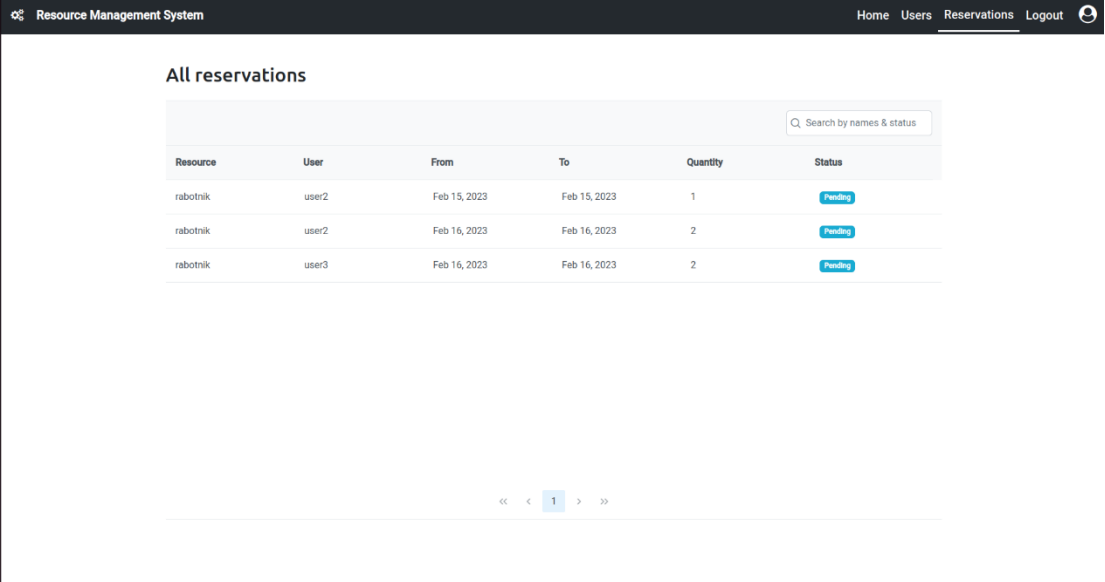
Слика 28

Администраторот може да регистрира нов корисник со притискање на зеленото копче “Register user” десно над табелата. По притискање на ова копче се појавува форма за внес на информации за корисникот, каде покрај username, e-mail и role, може да се специфицира и password (бидејќи ново креиранiot корисник нема да може да се најави без почетен password). Оваа форма е прикажана на Слика 29 подолу:



Слика 29

Доколку администраторот сака да ги прикаже сите резервации кои биле направени на системот (од сите корисници), истото може да го стори со притискање на копчето “Reservations” во десниот дел на навигацискиот бар горе. Кога тоа ќе се стори, се прикажува интерфејс како на Слика 30 подолу:



Resource Management System

Home Users Reservations Logout

All reservations

Search by names & status

Resource	User	From	To	Quantity	Status
robotnik	user2	Feb 15, 2023	Feb 15, 2023	1	Pending
robotnik	user2	Feb 16, 2023	Feb 16, 2023	2	Pending
robotnik	user3	Feb 16, 2023	Feb 16, 2023	2	Pending

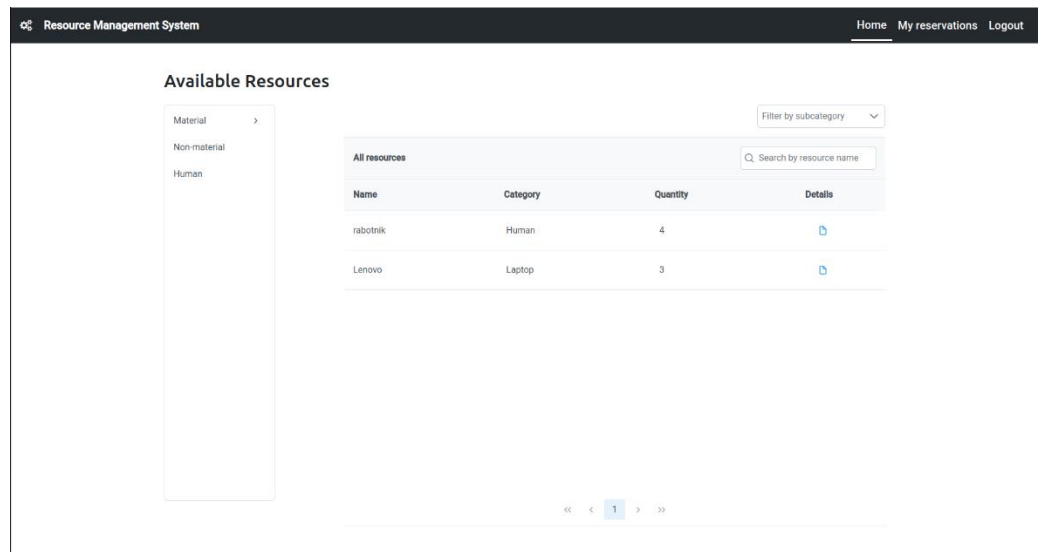
« < 1 > »

Слика 30

3.2.2 Опис на интерфејсот за корисник кој резервира ресурси

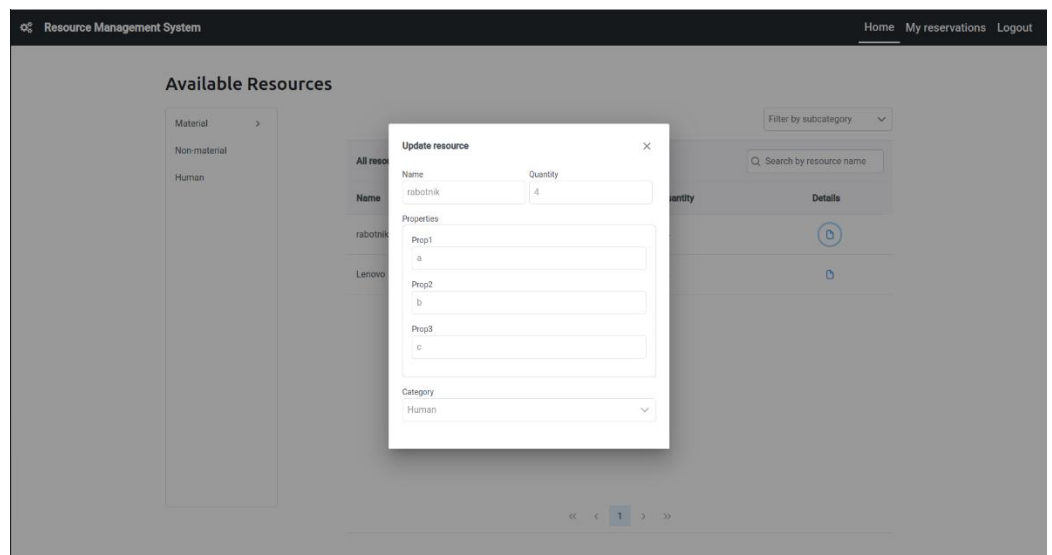
Голем дел од функционалностите кои може да се пристапат од страна на администраторот, исто така може да се пристапат и од корисник кој резервира ресурси, со неколку важни разлики:

- Во приказот на табелата со ресурси, кај корисник кој резервира ресурси го нема копчето за додавање нови категории (зеленото копче “Create category”). Исто така, притискање на категориите во левиот дел од приказот може единствено да го носи овој корисник низ дрвото од категории, а не може да се додаваат нови ресурси и категории. Тоа може да се види на Слика 31 подолу:



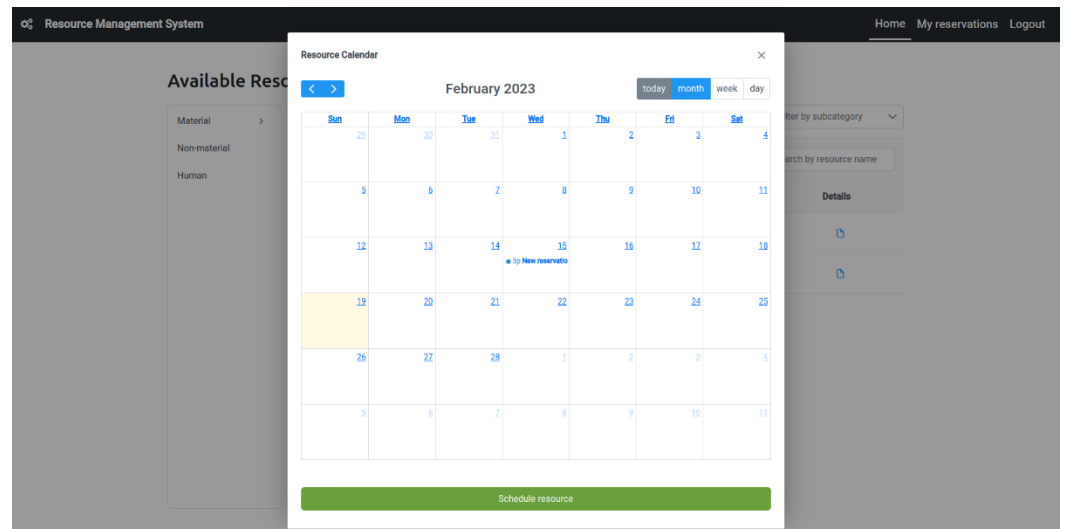
Слика 31

- Кога корисник кој сака да резервира ресурс ќе притисне на копчето во колоната “Details” за да се прикажат повеќе детали за ресурсот, овој корисник нема право да врши измени во својствата на ресурсот, ниту пак да го брише ресурсот. Тоа може да се види на Слика 32 подолу:



Слика 32

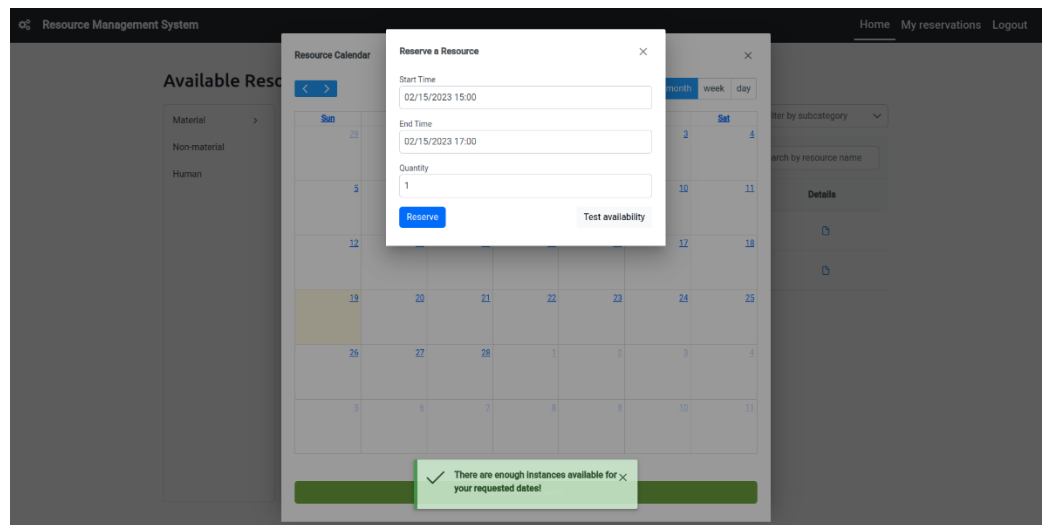
- Доколку корисник сака да резервира ресурс, потребно е да притисне било каде во редицата за тој ресурс во табелата со ресурси (целата редица е копче). Кога тоа ќе се стори, се прикажува календар со тековни резервации на ресурсот како на Слика 33 подолу:



Слика 33

Притоа, за разлика од администраторот, во овој приказ кај корисник кој не е администратор има копче за резервација на ресурсот.

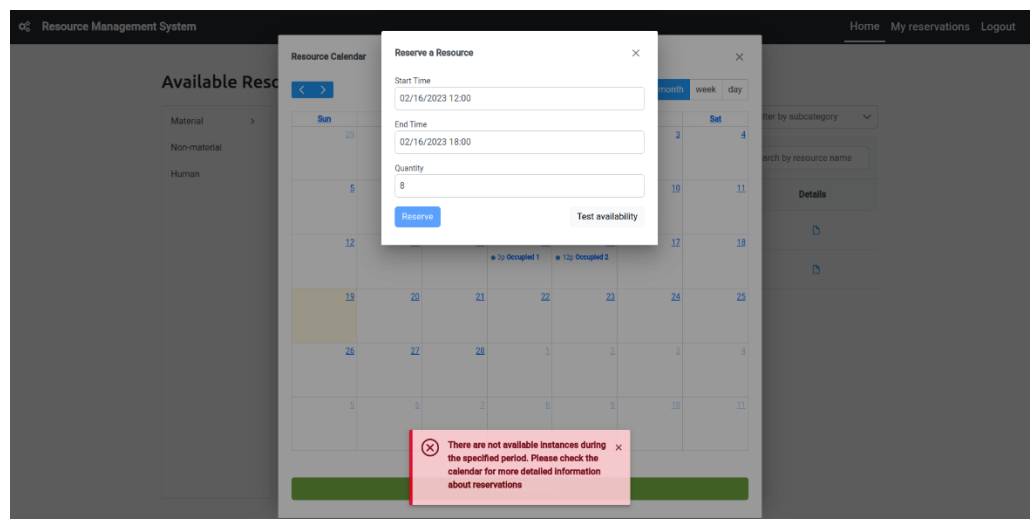
- Доколку корисник кој сака да резервира ресурс притисне на копчето “Schedule resource” при интерфејс како на Слика 33, ќе се прикаже форма за внес на информации за резервација (од кое време до кое време се резервира ресурсот, како и колку инстанци се резервираат). Изгледот на оваа форма при успешна резервација на ресурс е прикажан на Слика 34 подолу:



Слика 34

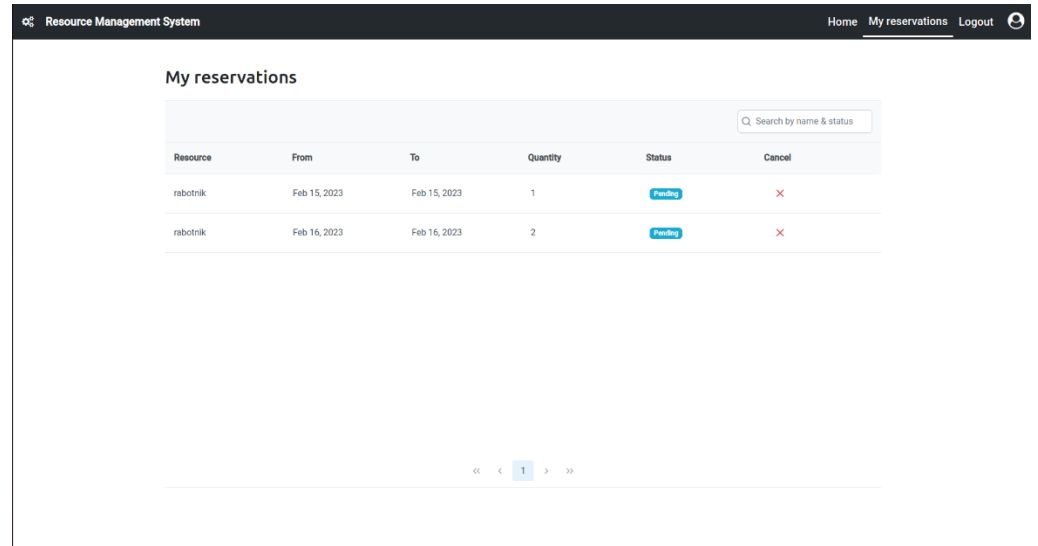
Притоа, доколку овој корисник притисне на копчето “Test availability”, ќе биде извршена проверка за тоа дали е можно да се изврши резервацијата (проверка дали има доволно слободни инстанци од тој ресурс во тој временски период).

Доколку оваа проверка не е успешна, ќе се прикаже информација дека не е можно да се изврши резервацијата, како што е прикажано на Слика 35 подолу:



Слика 35

- Доколку корисник кој резервира ресурси сака да има увид во сите свои резервации, истото може да го стори со притискање на копчето “My reservations” во десниот дел на навигацискиот бар горе. Кога тоа ќе се стори, се прикажува интерфејс како на Слика 36 подолу:



Слика 36

4. Користена литература

- <https://fastapi.tiangolo.com/>
- <https://restfulapi.net/>
- <https://angular.io/docs>
- <https://www.postgresql.org/docs/>
- <https://jwt.io/introduction>
- <https://frontegg.com/blog/jwt-authentication>