



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Никола Давогић

Судски подсистем информационог система еУправе

СЕМИНАРСКИ РАД

Технологије и системи еУправе

Нови Сад, 2024

У овом раду је описан судски подсистем еУправе, као део већег информационог система еУправе. Његова сврха је да омогући лакши рад судовима и повећа EGDl [1] индекс.

За реализацију је коришћена клијент-сервер архитектура, где је клијент веб апликација, израђена помоћу React [2] библиотеке, а сервер микросервисна апликација, за коју је коришћен програмски језик Go [3] и његове библиотеке. За чување података је коришћена MongoDB [4] база података, која је заједно се серверском апликацијом смештена у интегрисаном окружењу, које је креирано помоћу Docker [5] платформе.

Применом предложеног решења се омогућује провера судских информација од користи за грађане, као и начин за потпомогнуто пословање у судским процесима.

Развијен је прототип поменутог подсистема и објашњено је коришћење кроз рад са судским саслушањима, налозима за хапшење и суспензијама возачких дозвола.

Кључне речи

- веб апликација
- еУправа
- судски информациони систем

I Увод

Полазећи од проблема да је правни систем оптерећен, за последицу може имати лошу интеракцију са грађанством. Чекање дужи временски период како би се обавиле неке процедуре може одвратити грађане од суда и у неким случајевима утицати на резултат судског процеса. Представљајући начин да грађани приступе информацијама од значаја преко мреже, са било ког уређаја, може олакшати интеракцију грађанин-суд. Омогућавањем приступа позивима за саслушање, као и увид у суспензије, грађанима даје опцију информисања поред стандардних медијума, као што је папирна пошта. Поред погодности за грађане, судски информациони систем може олакшати и пословање радницима суда, пружајући им униформан приступ обављају својих дужности кроз поменути информациони систем.

Остатак рада је организован као што је наведено у наставку. У другом поглављу су приказана и објашњена сродна решења из домена. У трећем поглављу су описане технологије које се користе у решењу. У четвртом поглављу се налази спецификација захтева, која појашњава шта је све предвиђено да се обухвати овим решењем. У петом поглављу је дата спецификација дизајна, са описом организације решења. У шестом поглављу је приказана имплементација решења, односно начин његове реализације. У седмом поглављу је демонстрирано коришћење решења. У осмом поглављу је дат закључак са завршним размишљањима.

II Сродна решења

У овом поглављу су описана сродна решења из области судских информационих система.

Први пример сродног решења је портал еУправе Републике Србије, конкретно услуга издавања уверења о вођењу кривичног поступка [6]. Веб апликација пружа корисницима приказ основних информација о услузи, као и одговоре на неке од честих питања, везано за плаћање и временски рок за добијање уверења. Поред информација о услузи, налазе се контакт информације Министарства правде и дугме за покретање услуге. Да би се приступило услузи, потребно је да корисник буде улогован. Кориснички интерфејс је добро организован, али може изгледати збуњујуће грађанима који су навикли на једноставније приказе. На слици 1 је дат приказ поменутог корисничког интерфејса.



Слика 1 - Страница услуге за издавање уверења о вођењу кривичног поступка

Други пример сродног решења је електронска огласна табла Министарства правде Републике Србије [7]. Корисницима су приказана поља за упис имена, презимена и ЈМБГ-а, као и дугме за претрагу. Након клика на дугме „Претрага“, приказује се табела са пронађеним резултатима за дате параметре. Табела укључује колоне са датумом објаве, именом суда и документима везаних за ту објаву. Ова претрага је омогућена за физичка лица, правна лица, као и везана лица. Једноставан дизајн изускује лако коришћење и врло је интуитиван. У случају да је корисницима потребна помоћ у

коришћењу апликације, постоји дугме на које се кликом приказују информације о употреби. На слици 2 је приказан изглед корисничког интерфејса за претрагу.

Упутство ?

☒ Физичко лице ☐ Правно лице

☐ Везано лице

Име* Презиме* ЈМБГ*

Пера Перић 1234567891234 Претрага

| Редни број | Датум објаве | Суд | Документ |
|----------------|--------------|-----|----------|
| Нема резултата | | | |

« < > »

УПОЗОРЕЊЕ:

Неовлашћено коришћење туђег Јединственог матичног броја грађана (ЈМБГ) у сврху претраге електронске огласне табле представља кривично дело „Неовлашћено прикупљање личних података“ из члана 146 Кривичног законика за које је предвиђена новчана казна или казна затвора до годину дана.

Слика 2 - Претрага електронске огласне табле

На слици 3 је приказано корисничко упутство за употребу.

Упутство

- ▶ Претрага се врши употребом ћириличног или латиничног писма, без обзира на изабрано писмо.
- ▶ Тачност претраге зависи од исправно унетих података у апликацијама у којима се воде предмети.
- ▶ Морате унети име, презиме и ЈМБГ, односно назив фирме или ПИБ за правно лице (користити карактере ћ,ч,ђ,ж,ш).
- ▶ Поља обележена са * су обавезна поља.

Слика 3 - Корисничко упутство

III Коришћене технологије

У овом поглављу је дат опис коришћених технологија за решавање проблема.

Софтверско решење је имплементирано преко клијент-сервер архитектуре, што омогућава скалабилност у раду и флексибилност у комуникацији између клијентске и серверске стране. Клијентска апликација је задужена за приказ информација и контрола за акције кориснику, док серверска апликација извршава добављање података и логику везану за корисничке акције. Серверска апликација је организована у микросервисној архитектури, где сваки подсистем, укључујући судски, управља својом пословном логиком и доменом, уз заједнички сервис за јединствену идентификацију корисника. Тиме је додатно потпомогнута скалабилност и побољшана отпорност на грешке, механизмом за поновно покретање у случају престанка рада подсистема. Сваком подсистему је придружена база података, што омогућује оптималне брзине читања и писања, за разлику од коришћења једне централизоване базе података. У решењу су коришћене нерелационе базе података, чија је предност скалабилност и оптимално чување велике количине података, што је потребно за било који систем еУправе, да би се омогућио рад са целим грађанством.

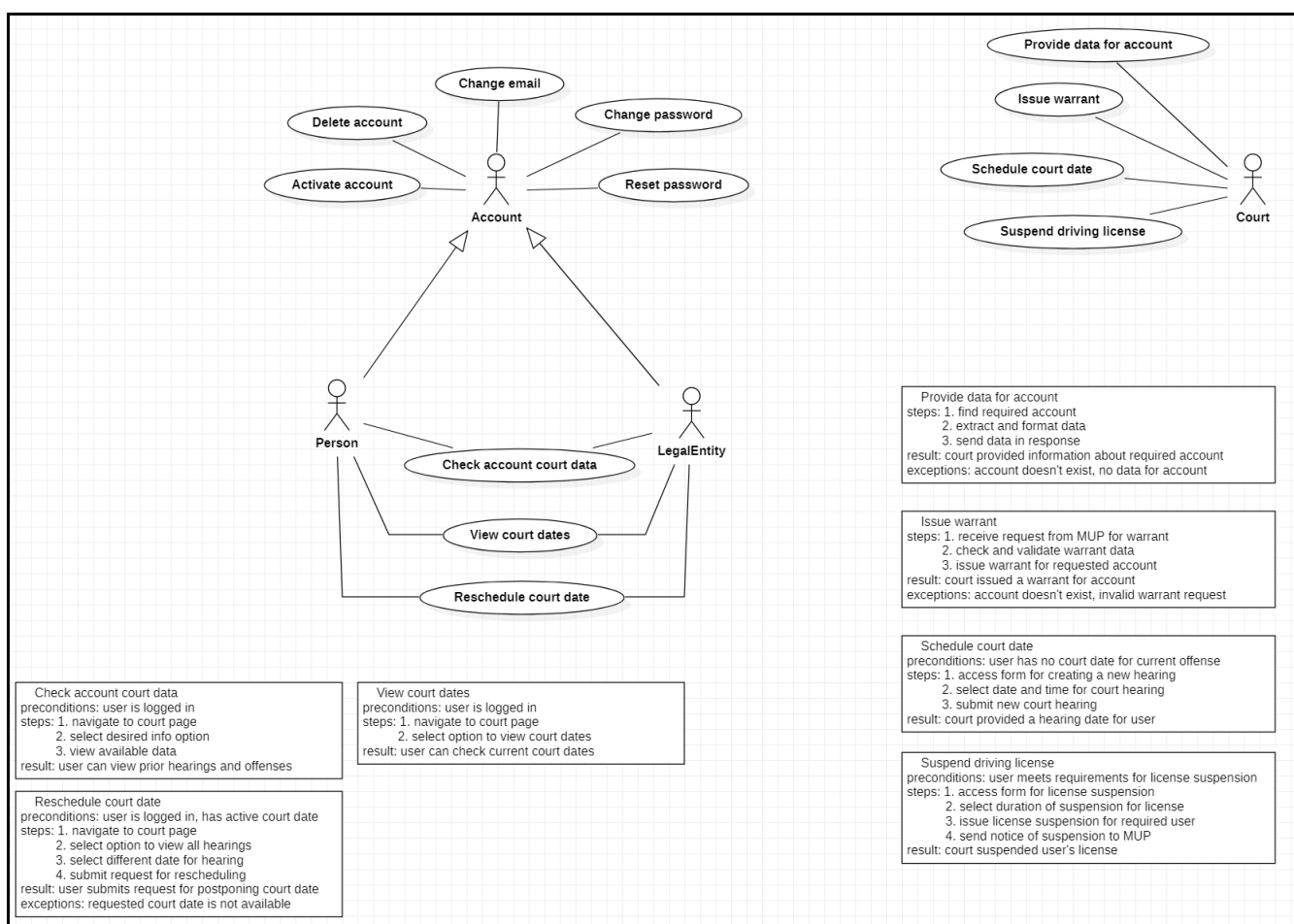
IV Спецификација захтева

У овом поглављу су објашњени функционални и нефункционални захтеви, обухваћени софтверским решењем овог рада.

Спецификација функционалних захтева

У овом одељку су представљени функционални захтеви, који су предвиђени за имплементацију кроз софтверско решење.

Подсистем подржава рад са физичким лицима и правним лицима. На дијаграму 1 су приказани случаји коришћења судског подсистема, заједно са дељеним случајевима подсистема за аутентификацију.



Дијаграм 1 - Случајеви коришћења судског подсистема

Табела 1 приказује опис случаја коришћења „Добављање судских корисничких података“.

| | |
|-------------------|---|
| Назив | Добављање судских корисничких података |
| Учесници | Корисник |
| Предуслови | Корисник је улогован |
| Кораци | <ol style="list-style-type: none"> 1. Корисник приступа страници суда 2. Корисник бира опцију за добављање информација 3. Кликом на дугме се шаље захтев серверу |
| Резултат | Кориснику су приказане одабране информације |
| Изузеци | / |

Табела 1 – Случај добављања судских података

Табела 2 приказује опис случаја коришћења „Одлагање датума саслушања“.

| | |
|-------------------|---|
| Назив | Одлагање датума саслушања |
| Учесници | Корисник |
| Предуслови | Корисник је улогован |
| Кораци | <ol style="list-style-type: none"> 1. Корисник приступа страници суда 2. Корисник бира опцију за приказ информација о саслушањима 3. У приказу листе саслушања, корисник врши клик на дугме за одлагање жељеног саслушања 4. Корисник бира нови датум саслушања |
| Резултат | Датум саслушања је успешно одложен |
| Изузеци | Датум није доступан, нови датум је пре већ постављеног датума |

Табела 2 – Случај одлагања датума саслушања

Табела 3 приказује опис случаја коришћења „Преглед датума саслушања“.

| | |
|-------------------|---|
| Назив | Преглед датума саслушања |
| Учесници | Корисник |
| Предуслови | Корисник је улогован |
| Кораци | <ol style="list-style-type: none"> 1. Корисник приступа страници суда 2. Корисник бира опцију за приказ информација о саслушањима |
| Резултат | Кориснику је приказана листа свих његових саслушања |
| Изузеци | Не постоје саслушања за корисника |

Табела 3 – Случај прегледа датума саслушања

Табела 4 приказује опис случаја коришћења „Достављање података за налог“.

| | |
|-------------------|---|
| Назив | Достављање података за налог |
| Учесници | Суд |
| Предуслови | / |
| Кораци | <ol style="list-style-type: none"> 1. Суд добија захтев за подацима од стране корисника или подсистема 2. Суд проналази тражене податке 3. Суд форматира пронађене податке и доставља их пошиљаоцу |
| Резултат | Суд је доставио тражене податке за налог |
| Изузеци | Налог не постоји, подаци за налог не постоје |

Табела 4 – Случај достављања података за налог

Табела 5 приказује опис случаја коришћења „Издавање налога за хапшење“.

| | |
|-------------------|--|
| Назив | Издавање налога за хапшење |
| Учесници | Суд |
| Предуслови | / |
| Кораци | <ol style="list-style-type: none"> 1. Суд добија захтев за издавање налога за хапшење од МУП-а 2. Суд проверава податке у захтеву и формира налог за хапшење 3. Суд издаје налог за хапшење за траженог корисника |
| Резултат | Суд је издао налог за хапшење |
| Изузеци | Корисник не постоји, подаци за налог нису валидни |

Табела 5 – Случај издавања налога за хапшење

Табела 6 приказује опис случаја коришћења „Заказивање судског саслушања“.

| | |
|-------------------|---|
| Назив | Заказивање судског саслушања |
| Учесници | Суд |
| Предуслови | Корисник нема заказано саслушање |
| Кораци | <ol style="list-style-type: none"> 1. Суд приступа форми за додавање новог саслушања 2. Суд уписује податке саслушања |
| Резултат | Суд је креирао ново саслушање |
| Изузеци | Уписани подаци нису валидни |

Табела 6 – Случај заказивања судског саслушања

Табела 7 приказује опис случаја коришћења „Суспендовање возачке дозволе“.

| | |
|-------------------|--|
| Назив | Суспендовање возачке дозволе |
| Учесници | Суд |
| Предуслови | Корисник испуњава услове за суспензију |
| Кораци | <ol style="list-style-type: none"> 1. Суд приступа форми за суспензију возачке дозволе 2. Суд бира период суспензије 3. Суд издаје суспензију за траженог корисника 4. Суд обавештава МУП о суспензији |
| Резултат | Суд је креирао суспензију за корисника и обавестио МУП о томе |
| Изузеци | Период суспензије није валидан, није било могуће обавестити МУП |

Табела 7 – Случај суспендовања возачке дозволе

Спецификација нефункционалних захтева

У овом одељку су представљени нефункционални захтеви, о којима софтверско решење треба да води рачуна.

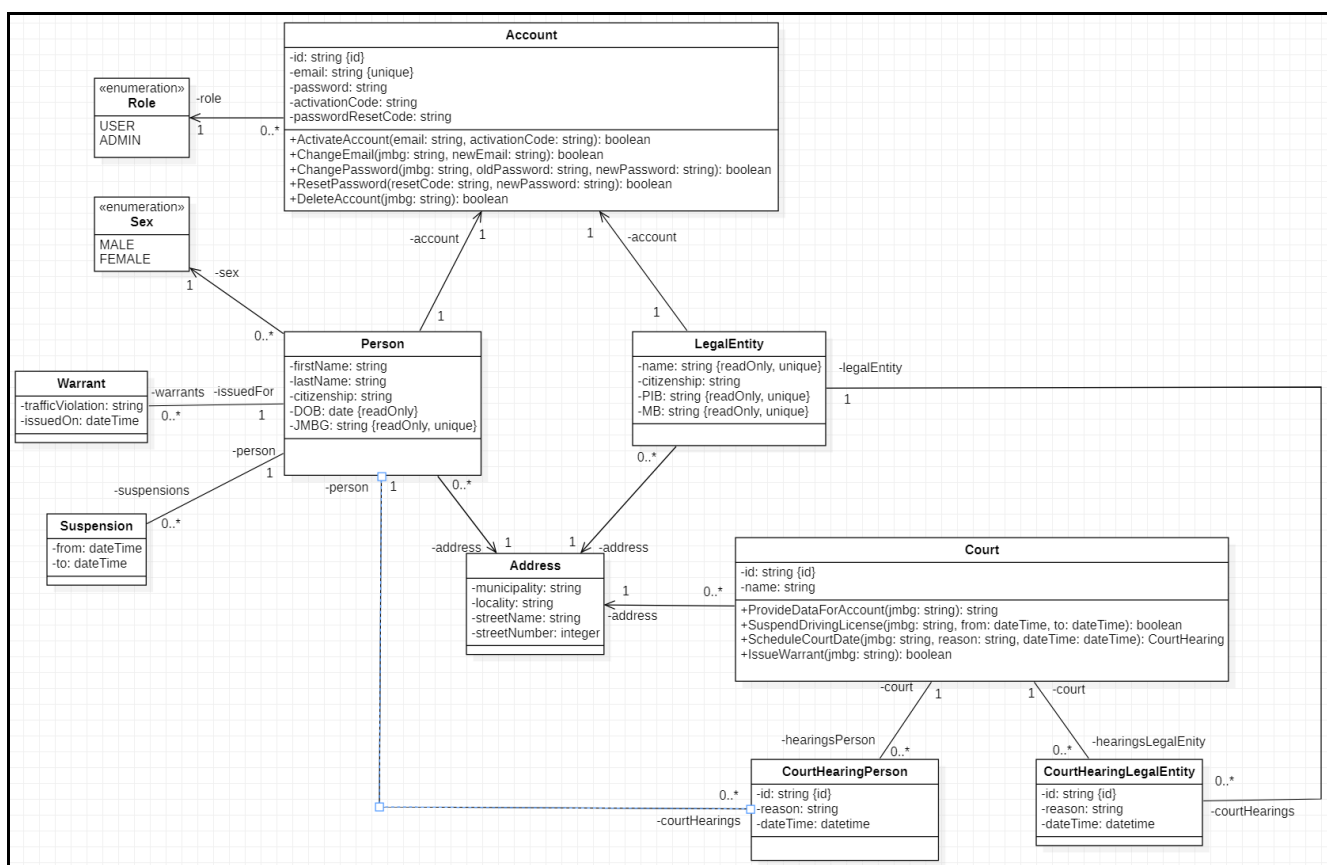
Брзина одзива система мора бити приближно реалном времену, како би корисницима искуство коришћења система било пријатно. Потребно је да систем буде скалабилан, како би подржао растући број корисника. Систем треба да омогући ефикасну употребу преко рачунарских, таблет и мобилних уређаја, кроз прилагодљив дизајн. Потребно је обезбедити приступачност за кориснике са слабијим видом. Системска отпорност на грешке мора бити на нивоу који не утиче на корисничко искуство. Осетљиве податке треба чувати на сигуран начин. Систем треба да обезбеди интуитиван дизајн и лако коришћење, како би корисничко искуство било оптимално.

V Спецификација дизајна

У овом поглављу је објашњен дизајн софтверског решења.

Комплетан информациони систем софтверског решења еУправе, организован је по подсистемима, сваки са својим задужењима и областима функционисања. Судски подсистем поред својих доменских функционалности, има и функционалности комуникације са осталим подсистемима, укључујући аутентификациони подсистем и подсистем МУП-а. Сви подаци о корисницима се налазе у аутентификационом подсистему, те је обавезна комуникација приликом сваке провере идентитета корисника и провере исправности примљених података о личности. Функционисање суда је уско везано са функционисањем МУП-а, где ова два подсистема врше међусобну размену информација и обавештавају један другог о потреби да се изврши нека акција.

На дијаграму 2 су преко класа представљене компоненте које спадају у судски подсистем, као и дељене компоненте са аутентификационим подсистемом.



Дијаграм 2 – Класни дијаграм судског подсистема

Корисник може бити физичко или правно лице. За сваког корисника је дефинисан налог, преко којег приступа услугама судског подсистема, као и адреса пребивалишта. За сваког корисника је могуће проверити стање саслушања, суспензија и налога за хапшење, којих не мора бити и може их бити више. Опционо, корисник може затражити одлагање саслушања. Суд има своје име и у придруженој бази података чува саслушања, суспензије и налоге за хапшење. Суд подржава акције манипулисања структурама из базе података, као и обавештавање подсистема МУП-а о њему битним променама у тим структурама.

VI Имплементација

У овом поглављу је приказано како су неке од функционалности подсистема имплементиране.

За добављање свих саслушања корисника, имплементирана је функција руковаоца HTTP [8] захтева *GetCourtHearingsByJMBG*. Код функције је приказан на слици 4.

```
func (ch *CourtHandler) GetCourtHearingsByJMBG(w http.ResponseWriter, r *http.Request) {
    params := mux.Vars(r)
    jmbg := params["jmbg"]

    log.Printf("Retrieving hearings for identifier '%s'", jmbg)

    hearings, err := ch.getHearings(jmbg)
    if err != nil && err.Error() != "hearings not found" {
        http.Error(w, "Failed to retrieve court hearings", http.StatusInternalServerError)
        log.Printf("Failed to retrieve court hearings: %s", err.Error())
        return
    }

    w.Header().Set("Content-Type", "application/json")
    w.WriteHeader(http.StatusOK)
    if err := json.NewEncoder(w).Encode(hearings); err != nil {
        http.Error(w, "Error while encoding body", http.StatusInternalServerError)
        log.Printf("Error while encoding court hearings: %s", err.Error())
    }

    if len(hearings) > 0 {
        log.Println("Successfully retrieved requested hearings")
    } else {
        log.Printf("No hearings found for '%s'", jmbg)
    }
}
```

Слика 4 - Функција за добављање саслушања

Функција прима ЈМБГ као URL [9] параметар, затим позива помоћну функцију за проналажење саслушања на основу ЈМБГ-а. Уколико је претрага базе података прошла без грешке, резултат се енкодира у JSON [10] структуру и поставља у тело HTTP захтева. На почетку и крају функције се исписују лог записи ради евиденције рада подсистема, као и у случајевима појављивања грешке. Код помоћне функције је приказан на слици 5.

Помоћна функција прима ЈМБГ као параметар и позива функцију репозиторијума за проналажење саслушања унутар базе података. Креирају се два низа саслушања и попуњавају се динамички, зависно од типа пронађених саслушања. Саслушања физичких и правних лица се чувају у одвојеним колекцијама. Уколико су пронађена саслушања, враћа се попуњени низ, иначе враћа грешку за непроналажење саслушања за дати ЈМБГ.

Функција репозиторијума добавља колекцију која садржи саслушања за физичка лица. Креира филтер на основу прослеђеног ЈМБГ-а. Креира се и context [11], који служи да ограничи време извршавања функције. Позива се Find [12] функција колекције, која у датом времену, са датим филтером, враћа показивач у ком је садржан резултат претраге. Итерацијом кроз показивач се декодирају инстанце саслушања и додавају у листу пронађених саслушања. На крају, уколико није било грешки приликом извршења функције, враћа се листа пронађених саслушања. Код функције репозиторијума је приказан на слици 6.

```

func (ch *CourtHandler) getHearings(jmbg string) ([]data.CourtHearing, error) {
    hearings, err := ch.repo.GetHearingsByJMBG(jmbg)
    if err != nil {
        return nil, err
    }

    var chp []data.CourtHearing
    var chle []data.CourtHearing

    for _, hearing := range hearings {
        if hearingPerson, ok := hearing.(*data.CourtHearingPerson); ok {
            chp = append(chp, hearingPerson)
        } else if hearingLegalEntity, ok := hearing.(*data.CourtHearingLegalEntity); ok {
            chle = append(chle, hearingLegalEntity)
        } else {
            return nil, fmt.Errorf("could not convert retrieved court hearing to any type")
        }
    }

    if len(chp) > 0 {
        return chp, nil
    } else if len(chle) > 0 {
        return chle, nil
    }

    return nil, fmt.Errorf("no hearings found for the given JMBG")
}

```

Слика 5 - Помоћна функција за добављање саслушања

```

func (cr *CourtRepo) GetHearingsByJMBG(jmbg string) ([]CourtHearing, error) {
    hearingsPerson := cr.getHearingsPersonCollection()

    filterPerson := bson.M{"person": jmbg}

    ctx, cancel := context.WithTimeout(context.Background(), 5*time.Second)
    defer cancel()

    var courtHearings []CourtHearing

    cursorPersons, err := hearingsPerson.Find(ctx, filterPerson)
    if err != nil {
        return nil, err
    }

    for cursorPersons.Next(ctx) {
        var chp CourtHearingPerson
        if err := cursorPersons.Decode(&chp); err != nil {
            return nil, err
        }
        courtHearings = append(courtHearings, &chp)
    }

    if err := cursorPersons.Err(); err != nil {
        return nil, err
    }

    if len(courtHearings) > 0 {
        return courtHearings, nil
    }

    return nil, errors.New("hearings not found")
}

```

Слика 6 - Функција репозиторијума за добављање саслушања

Функција клијентске апликације која обезбеђује приказ пронађених саслушања је *checkHearings*. Унутар ње се позива функција сервиса, који шаље HTTP захтев серверској апликацији за добављање листе саслушања. Функција сервиса враћа Promise [13], на основу чијег резултата се врши исцртавање интерфејса. Уколико је Promise успешан, приказује се листа саслушања или у случају да не постоје, порука о непостојању саслушања. Уколико је Promise неуспешан, приказује се порука о грешци приликом добављања листе саслушања. У случају успеха, функција подешава варијабле *Modal* компоненте, њеног садржаја и наслова и приказује *Modal* постављајући видљивост на *true*. Када је садржај листе саслушања већи од нуле, листа се прослеђује компоненти *HearingList*, која је задужена за приказ саслушања. Код функције је приказан на слици 7.

Функција сервиса за добављање саслушања је *getHearingsByJMBG*. Функција узима JWT [14] токен из *localStorage* [15] и декодира га, како би извукла ЈМБГ улоганог корисника. Тада уз помоћ *axios* [16] библиотеке шаље HTTP GET [17] захтев ка серверу, на URL конструисан уз помоћ глобалне варијабле *BASE_URL_COURT*, која садржи основни део путање ка серверу. У заглављу захтева се поставља и токен корисника, како би захтев прошао ауторизациону проверу на серверу. На крају враћа податке садржане у одговору. Код функције сервиса је приказан на слици 8.

```
const checkHearings = () => {
  getHearingsByJMBG().then((hearings: CourtHearing[]) => {
    if (hearings.length > 0) {
      setModalContent(<HearingList hearings={hearings} closeParent={closeModal} />);
      setModalHeading("Your hearings");
    } else {
      setModalContent("");
      setModalHeading("No court hearings for your account");
    }
    setModalVisible(true);
  }).catch(error => {
    toast.error("Failed to fetch hearings");
    console.error(error);
  });
};
```

Слика 7 - Функција клијентске апликације за добављање саслушања

```
export async function getHearingsByJMBG() {
  const token = localStorage.getItem("token");
  const jmbg = decodeJwtToken(token!).sub;

  try {
    const response = await axios.get(`${BASE_URL_COURT}/hearings/${jmbg}`, {
      headers: {
        Authorization: `Bearer ${token}`
      }
    });
    return response.data;
  } catch (error: any) {
    throw new Error(error.response.data.message || 'Failed to retrieve hearings');
  }
};
```

Слика 8 - Функција сервиса за добављање саслушања

Компонента *Modal* служи за динамички приказ жељеног садржаја и наслова. Могуће је контролисати њену видљивост, као и акцију која се извршава приликом затварања приказа компоненте. Код *Modal* компоненте је приказан на слици 9.

```
interface ModalProps {
  heading: string;
  content: any;
  isVisible: boolean;
  onClose: () => void;
}

const Modal = ({ heading, content, isVisible, onClose }: ModalProps) => {
  return isVisible ? (
    <ModalStyled>
      <ModalContentBoxStyled>
        <CloseButtonStyled onClick={() => onClose()}>
          <IoIosCloseCircle />
        </CloseButtonStyled>
        <HeadingStyled>{heading}</HeadingStyled>
        <section>{content}</section>
      </ModalContentBoxStyled>
    </ModalStyled>
  ) : null;
};
```

Слика 9 - Modal компонента

Компонента *HearingList* служи да на униформан начин прикаже листу саслушања. Код *HearingList* компоненте је приказан на слици 10.

```
interface HearingsProps {
  hearings: CourtHearing[];
  closeParent: () => void;
};

const HearingList = ({hearings, closeParent}: HearingsProps) => {
  const content = hearings.map(hearing =>
    <HearingCardStyled>
      <h1>{hearing.reason}</h1>
      <h6>{hearing.id}</h6>
      <h3>Date and time: {hearing.dateTime.replace("T", " ").replace("Z", "")}</h3>
      <p><b>Court: {hearing.court}</b></p>
      <Button
        buttonType="button"
        label="Reschedule"
        onClick={() => newReschedule(hearing.id!, hearing.dateTime, hearing.reason)} />
      </HearingCardStyled>
    );

  return (
    <>
      <HearingListStyled>{content}</HearingListStyled>
    </>
  );
};
```

Слика 10 - HearingList компонента

Компонента *HearingCardStyled* је styled [18] компонента, односно стилизована компонента, која служи да прикаже једно саслушање. За конфигурисање изгледа искоришћене су константе глобалне теме апликације. Код *HearingCardStyled* компоненте је приказан на слици 11.

```
import styled from "styled-components";

const HearingCardStyled = styled.section`
  background-color: ${(props) => props.theme.colors.base};
  border: ${(props) => props.theme.borders.standardBlack};
  margin: ${(props) => props.theme.margins.standard};
`;

export default HearingCardStyled;
```

Слика 11 - *HearingCardStyled* компонента

За одлагање саслушања имплементирана је функција руковаоца HTTP захтева *UpdateHearingPerson*. Код функције је приказан на слици 12.

```
func (ch *CourtHandler) UpdateHearingPerson(w http.ResponseWriter, r *http.Request) {
    log.Println("Rescheduling court hearing for a person")

    var rescheduledHearing data.RescheduleCourtHearing
    if err := json.NewDecoder(r.Body).Decode(&rescheduledHearing); err != nil {
        http.Error(w, InvalidRequestBody, http.StatusBadRequest)
        log.Println(InvalidRequestBodyError)
        return
    }

    rescheduledDateTime, err := time.Parse("2006-01-02T15:04:05", rescheduledHearing.DateTime)
    if err != nil {
        http.Error(w, "Error while decoding rescheduled date and time", http.StatusBadRequest)
        log.Println(InvalidRequestBodyError)
        return
    }

    courtHearing, err := ch.getHearing(rescheduledHearing.HearingID)
    if err != nil {
        http.Error(w, "Failed to retrieve court hearing", http.StatusInternalServerError)
        log.Printf("Failed to retrieve court hearing: %s", err.Error())
        return
    }

    if rescheduledDateTime.Before(courtHearing.GetDateTime()) {
        http.Error(w, "Court hearing can't be rescheduled before set date and time", http.StatusBadRequest)
        log.Println(InvalidRequestBodyError)
        return
    }

    err = ch.repo.RescheduleCourtHearingPerson(rescheduledHearing)
    if err != nil {
        http.Error(w, "Failed to update court hearing", http.StatusInternalServerError)
        log.Printf("Failed to update court hearing: %s", err.Error())
        return
    }

    w.WriteHeader(http.StatusOK)
    log.Println("Successfully rescheduled court hearing")
}
```

Слика 12 - Функција одлагања саслушања

Функција декодира тело захтева, које садржи структуру за ажурирање саслушања и парсира текстуалну репрезентацију датума и времена. Проналази већ постојеће саслушање у бази података, проверава валидност датума и времена из захтева и уколико је валидно, ажурира постојеће саслушање, кроз функцију репозиторијума *RescheduleCourtHearingPerson*. Код функције репозиторијума је приказан на слици 13.

```
func (cr *CourtRepo) RescheduleCourtHearingPerson(rescheduledHearing RescheduleCourtHearing) error {
    collection := cr.getHearingsPersonCollection()

    objID, err := primitive.ObjectIDFromHex(rescheduledHearing.HearingID)
    if err != nil {
        return err
    }

    filter := bson.M{"_id": objID}

    dateTime, err := time.Parse("2006-01-02T15:04:05", rescheduledHearing.DateTime)
    if err != nil {
        cr.logger.Println("Error while parsing date")
        return err
    }

    update := bson.M{
        "$set": bson.M{
            "dateTime": dateTime,
        },
    }

    ctx, cancel := context.WithTimeout(context.Background(), 5*time.Second)
    defer cancel()

    result, err := collection.UpdateOne(ctx, filter, update)
    if err != nil {
        cr.logger.Println("Error while updating collection")
        return err
    }

    if result.ModifiedCount > 0 {
        cr.logger.Println("Successfully updated collection")
        return nil
    }

    cr.logger.Printf("Invalid court hearing ID: %s", rescheduledHearing.HearingID)
    return errors.New("invalid hearing ID")
}
```

Слика 13 - Функција репозиторијума за ажурирање саслушања

Функција репозиторијума добавља колекцију са саслушањима физичких лица. Формира филтер на основу прослеђене идентификације саслушања, парсира датум и време и формира *update* акцију, која ће бити извршена над базом података. Акција мења вредност *dateTime* поља на нову, прослеђену вредност датума и времена. Уз временско ограничење од пет секунди, акција се извршава над базом података и чува се њен резултат. У резултату се провера колико докумената је промењено. Ако је промењено више од нула докумената, акција је успешно извршена, иначе се враћа грешка о невалидној идентификацији саслушања.

Функција клијентске апликације која врши ажурирање саслушања је *rescheduleHearing*. Функција позива помоћну функцију, која одређује да ли се врши ажурирање саслушања физичког или правног лица. Помоћна функција позива сервисну функцију, која враћа Promise као резултат, на основу којег се одређује успешност одлагања саслушања. Уколико је саслушање успешно, приказује се порука о успешном одлагању, иначе се приказује порука о грешци приликом одлагања. Код функције која врши ажурирање, помоћне функције и сервисне функције су приказани на сликама 14, 15 и 16.

```
const rescheduleHearing = (formData: any) => {
  const retVal = updateHearing(formData["hearingID"], formData["dateTime"]);
  retVal.then(() => {
    toast.success("Successfully rescheduled court hearing");
    closeModal();
    closeParent();
  }).catch((error) => {
    toast.error("Failed to reschedule court hearing");
    console.error(error);
  });
};
```

Слика 14 - Функција клијентске апликације за ажурирање саслушања

```
export function updateHearing(hearingID: string, dateTime: string): Promise<any> {
  const token = localStorage.getItem("token");
  const jmbg = decodeJwtToken(token!).sub;

  const hearingUpdate = {
    hearingID: hearingID,
    dateTime: dateTime + ":00"
  };

  let response;
  if (jmbg.length > 8) response = updateHearingPerson(hearingUpdate);
  else response = updateHearingLegalEntity(hearingUpdate);

  return response;
};
```

Слика 15 – Помоћна функција за ажурирање саслушања

```
export async function updateHearingPerson(hearing: RescheduleCourtHearing) {
  const token = localStorage.getItem("token");

  try {
    const response = await axios.put(`${BASE_URL_COURT}/update-hearing-person`, hearing,
    {
      headers: {
        Authorization: `Bearer ${token}`
      }
    });
    return response.data;
  } catch (error: any) {
    throw new Error(error.response.data.message || 'Failed to update person hearing');
  }
};
```

Слика 16 – Функција сервиса за ажурирање саслушања физичког лица

VII Демонстрација

VIII Закључак

У овом раду је представљено софтверско решење за судски подсистем еУправе. Приказано решење пружа корисницима лакши приступ судским подацима од интереса, кроз различите врсте уређаја. Решење такође олакшава пословање судским службеницима, пружајући им униформан приступ обављању својих послова.

Софтверско решење врши аутоматску комуникацију са осталим подсистемима, те у подржаним функционалностима није потребно слање дописа и обавештења. Решењем нису обухваћене функционалности издавања потврда и уверења, те се оне морају обављати ван пројектоване апликације. Даљим усавршењем, кроз интеграцију преосталих судских функционалности, било би омогућено и издавање потврда и уверења, што би омогућило комплетно корисничко искуство преко пројектоване апликације.

У поређењу са сродним решењима, пројектована апликација такође нуди једноставно и интуитивно корисничко искуство, са јасно дефинисаним интерфејсом. Употребом HTTP захтева за комуникацију између клијента и сервера, оставља се простор за даљу интеграцију са осталим сервисима, која може повећати интероперабилност целокупног система еУправе.

Комплетан код имплементираног софтверског решења је доступан на GitHub репозиторијуму [19].

Литература

- [1] United Nations. 2024. E-Government Development Index (EGDI). Преузето 29.06.2024. са <https://tinyurl.com/un-egdi>
- [2] Meta Open Source. 2024. React. Преузето 29.06.2024. са <https://react.dev>
- [3] Google. 2024. Go. Преузето 29.06.2024. са <https://go.dev/doc>
- [4] MongoDB, Inc. 2024. MongoDB. Преузето 29.06.2024. са <https://www.mongodb.com>
- [5] Docker Inc. 2024. Docker. Преузето 29.06.2024. са <https://docs.docker.com>
- [6] Канцеларија за информационе технологије и електронску управу. 2024. еУправа. Преузето 29.06.2024. са <https://euprava.gov.rs/usluge/6521>
- [7] Министарство правде. 2024. Електронска огласна табла. Преузето 29.06.2024. са <https://etabla.sud.rs>
- [8] Mozilla. 2024. HTTP. Преузето 30.06.2024. са <https://tinyurl.com/mdn-http>
- [9] Mozilla. 2024. What is a URL? Преузето 30.06.2024. са <https://tinyurl.com/mdn-url>
- [10] Douglas Crockford. 2024. Introducing JSON. Преузето 30.06.2024. са <https://www.json.org>
- [11] Google. 2024. context package. Преузето 30.06.2024. са <https://pkg.go.dev/context>
- [12] MongoDB, Inc. 2024. db.collection.find(). Преузето 30.06.2024. са <https://www.mongodb.com/docs/manual/reference/method/db.collection.find>
- [13] Mozilla. 2024. Promise. Преузето 30.06.2024. са <https://tinyurl.com/mdn-promise>
- [14] Okta, Inc. 2024. Introduction to JSON Web Tokens. Преузето 30.06.2024. са <https://jwt.io/introduction>
- [15] Mozilla. 2024. Window: localStorage property. Преузето 30.06.2024. са <https://tinyurl.com/mdn-localstorage>
- [16] Matt Zabriskie. 2024. Getting Started. Преузето 30.06.2024. са <https://axios-http.com/docs/intro>
- [17] Mozilla. 2024. GET. Преузето 30.06.2024. са <https://tinyurl.com/mdn-http-get>
- [18] Glen Maddern and Maximilian Stoiber. 2024. Documentation. Преузето 30.06.2024. са <https://styled-components.com/docs>
- [19] ND-SR7. 2024. eUprava. Преузето 30.06.2024. са <https://github.com/ND-SR7/eUprava>