

Spletna aplikacija za dostop do medicinskih slik v standardu DICOM

Žan Regoršek¹, Samo Penič¹, Miha Fošnarič², Janez Žibert²

¹Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška 25, SI-1000 Ljubljana, Slovenija

²Zdravstvena fakulteta, Univerza v Ljubljani, Zdravstvena pot 5, SI-1000 Ljubljana, Slovenija

E-pošta: samo.penic@fe.uni-lj.si

Web application for accessing medical DICOM images

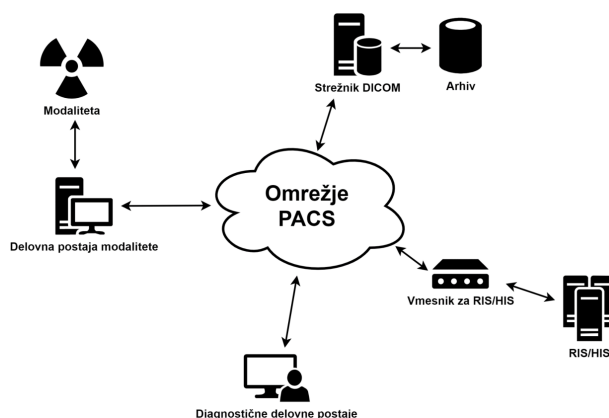
Medical imaging is today mostly digitized. There are many picture archiving and communication systems (PACS) available and are often integrated with hospital information system (HIS) and/or radiological information system (RIS). In practice, however, patients still receive their medical images as a printed hardcopy; on a CD, DVD or USB drive, where they are often saved in a DICOM (Digital Imaging and Communications in Medicine) format. This requires appropriate software and hardware to access them.

In this work we present a lightweight application for remote access to medical images from a DICOM server. It comes with user management and medical image preview and is built on open-source components. It was developed for educational and scientific purposes at the University of Ljubljana, and can be used as a basis for an affordable solution, suitable for medical facilities performing diagnostic imaging, for example small medical clinics with ultrasonography.

1 Uvod

Vse bolj prisotna digitalizacija omogoča tudi v medicinski slikovni diagnostiki posredovanje slik in drugih podatkov o preiskavah pacientom in strokovnemu osebju na hiter, učinkovit in varen način. Pri večini medicinskih slikovnih preiskav se podatki danes zajemajo v digitalni obliki in tudi zdravniki ter inženirji radiologije slike pogosto pregledujejo in obdelujejo na računalniku. V ta namen so na voljo mnogi informacijski sistemi za pošiljanje in arhiviranje slik (*angl.* Picture Archiving and Communication System – PACS). PACS je, predvsem v večjih zdravstvenih ustanovah, pogosto povezan z bolnišničnim informacijskim sistemom (*angl.* hospital information system – HIS) in/ali radiološkim informacijskim sistemom (RIS) (slika 1) [1, 2].

V praksi pacient še vedno večinoma dobi slike kar natisnjene v fizični obliki ali pa shranjene na zgoščenko ali USB ključek. Te so navadno shranjene v formatu standarda digitalnega slikanja in komunikacije v medicini (*angl.* Digital Imaging and Communications in Medicine – DICOM) [3]. Tako za pregled slik še vedno potrebujemo ustrezno strojno in programsko opremo, pri čemer smo praviloma omejeni na uporabo računalnika, saj na



Slika 1: Primer integracije sistema PACS v celostno rešitev, ki nudi zajem medicinskih diagnostičnih slik, njihovo hrambo in nadaljnjo obdelavo ter integracijo v druge informacijske sisteme, kot sta HIS in RIS [1].

mobilnih napravah primerne programske opreme za pregled slik nimamo. Ob tem se postavlja tudi vprašanje hrambe in varnosti podatkov ter obremenjevanja okolja. Nenazadnje prenosni mediji predstavljajo tudi dodaten strošek.

V prispevku predstavljamo preprosto spletno aplikacijo za dostop do medicinskih slik, ki so zbrane v centralnem strežniku v DICOM formatu. Aplikacija je s svojimi odprtokodnimi komponentami primerna za pedagoške in raziskovalne namene Univerze v Ljubljani in je osnova za razvoj cenovno dostopne storitve predvsem v manjših medicinskih diagnostičnih ustanovah, kot so na primer ambulante z ultrazvočno diagnostiko.

Prispevek je organiziran tako, da v naslednjem poglavju opišemo zasnovano aplikacijo, njen uporabniški vmesnik in platformo na kateri je zgrajena, čemur sledi kratka razprava o razvoju aplikacije in zaključki.

2 Aplikacija

2.1 DICOM

Standard DICOM organizira medicinske slike po hierarhiji pacient → študija → serija → instanca. Aplikacija obravnava omenjene ravni po naslednji ideji:

- **Pacient:** unikatna identiteta pacienta;

- **Študija:** specifično področje/razlog zdravljenja (npr. zlom kosti, nosečnost, neznani glavoboli ipd.), ki poteka dlje časa in je skupek več individualnih pregledov – serij;
- **Serijski:** posamezen pregled v ambulanti, ki je del postopka zdravljenja. Pregled lahko sicer razdelimo na dve seriji, v kolikor sta bili uporabljeni dve različni modalnosti. Tipično poimenovanje bi bilo npr.: 1. pregled celjenja kosti - rentgen, 16. 5. 2020; 2. pregled celjenja kosti - rentgen, 18. 6. 2020; 2. pregled celjenja kosti - MRI, 18. 6. 2020;
- **Instanca:** ob posameznem pregledu se zdravnik odloči posneti sliko iz več zornih kotov ali pa z različnimi nastavitvami slikovne aparature. Vsaka posamezna slika predstavlja eno instanco, ki nadomesti fizično sliko ali posamezno datoteko na prenosnem mediju. V primeru CT posnetka lahko ena instanca predstavlja posamezno rezino.

2.2 Uporabniški vmesnik

Uporabnika najprej pričaka vpisno okence (slika 2). Vsak uporabnik ima unikatno identifikacijsko številko, v primeru pacienta morda kar številko zdravstvenega zavarovanja in osebno geslo. Uporabnik je lahko pacient, ki ima dostop le do lastnih medicinskih slik, ali zdravnik, ki ima dostop do slik vseh svojih pacientov. Tako je zdravniku ob vpisu ponujen seznam pacientov, medtem ko je pacient takoj usmerjen na zavihek študije.

Uporabnik izbere ime pacienta (študije, serije, instance) in izbiro potrdi s klikom na gumb (slika 3). S strežnika Orthanc[4] se prenesejo ustrezni podatki, nato pa se odpre nov zavihek naslednje ravni. Za večino uporabnikov je najbolj zanimiv zavihek serija, ki omogoča tekoč pregled vseh instanc posamezne študije. S klikom na "Poglej preiskavo v novem zavihku" se odpre spletni prikazovalnik Orthanc, v katerem lahko s kolesčkom na miški drsimo skozi instance. Če miško s pritisnjenim levim gumbom vodimo levo ali desno po sliki, spreminjamo kontrast in svetlost posamezne instance.

Vsako instanco, serijo ali študijo (vse serije) lahko prenesemo v obliki arhiva zip. Prenešene datoteke so datoteke dcm (format DICOM), za katere potrebujemo ustrezno programsko opremo (npr. pregledovalnik medicinskih slik Weasis [5]). Prenešene datoteke dcm so nestisnjena oblika medicinske slike, ki jo ustvaril posamezen slikovni aparat. Posamezna datoteka je zato lahko velika do nekaj deset MB. Za paciente večinoma ne prinaša dodatne vrednosti, zdravniku pa omogoča naknadno prilaganje več parametrov posnetka.

Zavihek "Instanca" prikaže posamezno instanco v formatu png. Na levi strani je ves čas prisotna orientacijska razpredelnica, ki uporabnika seznanja s trenutno izbranimi ravni hierarhije DICOM (slika 4).

2.3 Platforma R Shiny

Aplikacija za oddaljen dostop do digitalnih izvidov po standardu DICOM je napisana v statističnem programskem jeziku R, razširjenim s knjižnico Shiny [6]. Sle-

Slika 2: Zgoraj: Prijavno okno. Ker je aplikacija DEMO različica, so objavljeni tudi uporabniški podatki. Spodaj: Izbira pacienta. Seznam pacientov je v obliki padajočega seznama.

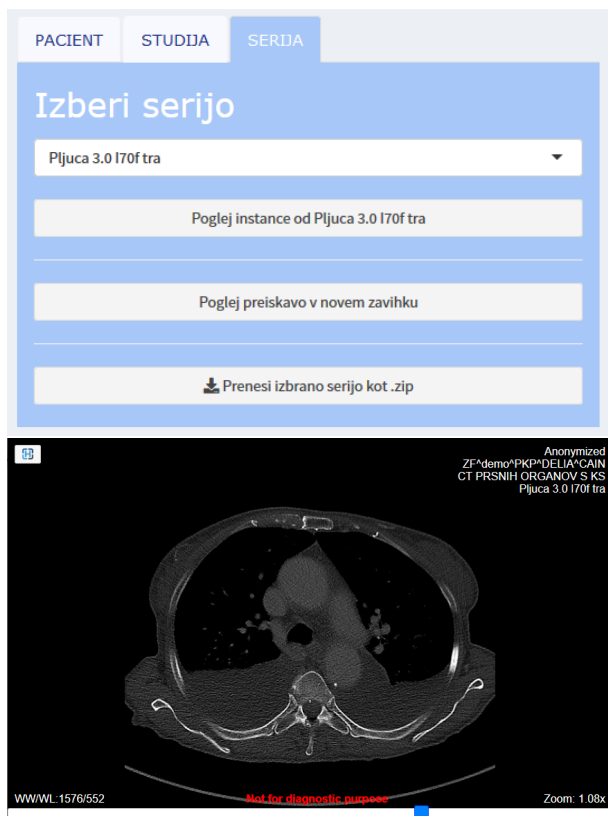
dnja omogoča postavitve spletne aplikacije brez poglobljenega znanja programiranja spletnih strani in vsebuje uporabniški vmesnik (*angl.* front-end) in strežniški del kode (*angl.* back-end).

Osnovna struktura aplikacije Shiny R:

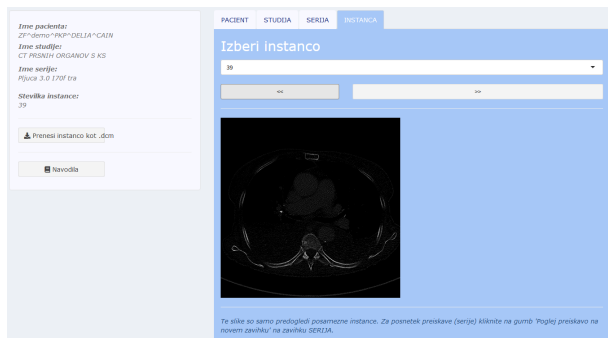
```
library(shiny)
# Define UI for application
ui <- fluidPage()
# Define server logic
server <- function(input, output) {}
# Run the application
shinyApp(ui = ui, server = server)
```

Spletna aplikacija v R (Shiny) je razdeljena na dva dela – na strukturi server in ui, ki se kot argumenta podata funkciji shinyApp(). Slednja zažene strežnik, ki nudi dostop do aplikacije. Na osebem računalniku v času razvoja za to poskrbi kar razvojno okolje RStudio, na javnem strežniku pa potrebujemo RStudio Server.

V ui je koda za uporabniški vmesnik. Določiti moramo osnovno postavitve strani, morebitno razdelitev na zavihke in položaje modulov s katerimi uporabnik upravlja (gumbi, meniji, lokacije slik,...). Vsak modul se



Slika 3: Zgoraj: Zavihek serija. Spodaj: spletni prikazovalnik Orthanc.



Slika 4: Trenutno izbrane DICOM ravni na zavihku instance.

določi v obliki funkcije, npr. `tabPanel(args)`, `actionButton(args)`, `selectInput(args)`. Funkcija `ui` se izvede le enkrat, ko se stran naloži, nato pa se ne izvaja več. To pomeni, da se moduli kasneje ne spreminjajo glede na interakcije z uporabnikom, temveč je njihov izgled (slika), funkcija (gumb) ali vsebina (meni) določena ob naložitvi aplikacije. To omogoča izdelavo strani z le izrazito deterministično strukturo in podatki, ki so na voljo že ob zagonu aplikacije ter se nato ne spreminjajo več. Za prikaz dinamičnih podatkov je rešitev uporaba funkcije `uiOutput()`, ki osrednji strukturi `ui` pove, da bo na določenem mestu nek element, ki se bo spreminjal glede na uporabnikove vnose. Tako se ob naložitvi strani za določen element le rezervira geometrijski

prostor v aplikaciji, vsebina pa se dinamično oblikuje in spreminja med uporabe aplikacije.

Uporaba funkcije `uiOutput()`:

```
tabPanel(
  title = div("PACIENT",
    style="font-family:verdana;"),
  value = "1",
  column(
    width = 12,
    uiOutput("selectPatient"),
    uiOutput("toStudyButton"),
    hr(),
    uiOutput(
      "launchPatientHweasisButton"),
  ),
  fluidRow(),
  br()
)
```

V naši aplikaciji smo se odločili še za dinamično dodajanje oz. odstranjevanje zavihkov glede na napredovanje uporabnika po hierarhiji standarda DICOM. Funkcije, ki jih proži uporabnik s klikom na gumb, kličejo funkcijo `addPanel()`, ki znotraj aplikacije naloži dodaten zavihek, o katerem pred klikom ni bilo nič znanega. Tako smo iz strukture `ui` praktično vso kodo, ki določa izgled, preselili v strukturo `server`, saj je stran izjemno dinamična in nedeterministična.

V strukturi `server` se poleg funkcij za izgradnjo omenjenih dinamičnih elementov nahaja vsa logika aplikacije. Ta vključuje funkcije, ki se odzivajo na klike gumbov in izvajajo klice funkcij, ki iz Orthanc strežnika prenesejo podatke o pacientu, študiji, seriji oz. instanci in jih nato implementirajo v padajočih menijih in gumbih oz. prikažejo kot slike, ter dodajo ali odstranijo zavihke. Struktura `server` tako predstavlja 95% vse programske kode.

Aplikacija v okviru razširitve `shinydashboard` vsebuje tudi vpisno okence, ki je pravzaprav neodvisna pred-aplikacija, ki ob uspešni avtentikaciji naloži osrednjo aplikacijo. Ob vpisu se glede na uporabniško ime določi tudi stopnja dostopa v obliki spremenljivke `permission`, ki se uporablja kot indikator v osrednji aplikaciji za omejitve modulov, ki se naložijo. Uporabnikovi prijavnji podatki so v našem primeru definirani kar v programski kodi, vendar omenjena razširitev omogoča tudi preverjanje iz podatkovne baze.

Programiranje z vgrajenimi funkcijami je sicer precej omejeno, saj je razširitev `Shiny` namenjena predvsem predstavitvi statističnih podatkov in ne toliko splošnih podatkov, prenesenih iz drugih virov. Omogoča tudi dodajanje kode CSS za oblikovanje in kode JavaScript za dodajanje manjših funkcionalnosti.

Uporaba HTML `style` značke za dodajanje kode CSS:

```
div(paste(
  reactVals$instances$instanceNumber
  [instIdx(input, reactVals)]),
  style="font-family:verdana;
```

```
font-style: italic;" ),
hr(),
downloadButton(
  outputId = "downloadInstance",
  label = "Prenesi instanco
kot .dcm",
  style = "vertical-align: middle;
font-family: verdana;
width: 50%;" )
```

Baza podatkov za aplikacijo je strežnik Orthanc. Orthanc ima pripravljen aplikacijski vmesnik REST (iz angl. “Representational state transfer”), tako imenovan REST API, ki omogoča komuniciranje s strežnikom. Za uporabo zahtev REST moramo v skripto R jezika dodati knjižnico `httr`. Naša aplikacija podatke le prenaša, tako da uporablja le GET zahtevo v obliki funkcije `GET()`. Funkcija vrne odgovor iz strežnika, ki je v primeru Orthanca zapakiran v formatu JSON (skrajšano od angl. “JavaScript Object Notation”). V tem primeru zna R zapisati vsebino odgovora JSON v spremenljivko, do vsebine posameznih značk znotraj odgovora JSON pa se enostavno dokopljemo tako, da zaporedoma pišemo značke. Alternativa je, da uporabimo funkcijo, ki nam želena značko izlušči iz odgovora, npr. s `status_code(response)`.

GET zahteva za prenos podatkov o pacientu:

```
response = GET(
  paste("https://pacs.zf.uni-lj.si/
orthanc/patients/",
  patientIDs[i], sep=""),
  authenticate(username, password)
)
patientName =
  content(response)
  $MainDicomTags$PatientName
patID =
  content(response)
  $MainDicomTags$PatientID
status =
  status_code(response)
```

Funkcije iz `httr` knjižnice namenjene komuniciranju sprejmejo tudi opcijski argument `authenticate()`, kjer podamo podatke za dostop do zaščitenega strežnika. Prenos datotek se opravi z uporabo funkcije jezika R `download()`.

Prenos instance v obliki slike v formatu png:

```
outfile <- tempfile(fileext = '.png')
src=paste("https://",
  username, ":",
  password,
  "@pacs.zf.uni-lj.si
/orthanc/instances/",
  instanceID, "/preview",
  sep = "" )
download.file(src, outfile,
  mode = 'wb')
```

3 Razprava in zaključek

Tekom razvoja aplikacije smo naleteli na nekaj težav, saj je R Shiny zasnovan predvsem za prijazen prikaz podatkov generiranih v jeziku R in ne toliko za oblikovanje splošnih spletnih aplikacij. Tudi dokumentacija za knjižnico Shiny je precej skopa. So pa toliko bolj bogati spletni forumi, kar daje vedeti, da je Shiny pogosto uporabljano okolje tudi za izdelavo spletnih aplikacij. Hkrati omogoča oblikovanje s kaskadnimi stilskimi podlogami (CSS) in dodajanje JavaScript kode. Zaradi strukture jezika R (Shiny) daljša koda sicer hitro postane nepregledna, hkrati pa deljenje aplikacije na več datotek ni preveč razvijalcu prijazno. Povzeli bi lahko, da je R Shiny primeren za oblikovanje enostavnejših aplikacij za prikaz večje količine podatkov in manj za specifične aplikacije z veliko interakcije z uporabnikom. V našem primeru je sicer R Shiny več kot zadostoval in na koncu je nastala konceptualna aplikacija, ki izpolnjuje vse na začetku zastavljene cilje.

Razvili smo preprosto odprtokodno spletno aplikacijo za dostop do medicinskih slik v formatu DICOM, ki je lahko tudi osnova za razvoj cenovno dostopne storitve, ki morda najde svoje mesto predvsem v manjših medicinskih ustanovah s slikovno diagnostiko.

4 Zahvala

Pričujoče delo je v določeni meri del projekta *Odprtokodni sistem za zajem in obdelavo medicinskih slik iz ultrazvočnih diagnostičnih naprav po DICOM standardu*, ki smo ga med marcem in junijem 2020 izvedli v okviru programa *Po kreativni poti do znanja*. Zato se avtorji zahvaljujemo *Javnemu štipendijskemu, razvojnemu, invalidskemu in preživitvenemu skladu RS, Ministrstvu za izobraževanje, znanost in šport RS, Evropskemu socialnemu skladu EU, Zdravstveni fakulteti UL ter Univerzi v Ljubljani* za finančno in logistično podporo. In nenazadnje seveda tudi vsem ostalim sodelujočim na projektu: podjetju *ECHOSON d.o.o.* in študentom Manci Pišek, Pii Pikelj, Sari Kidrič, Ninu Kleindienstu in Nejcju Klanjščku.

Literatura

- [1] A. Gerečnik, *Implementacija odprtokodnih sistemov PACS IN RIS*. Ljubljana: Zdravstvena fakulteta, 2020.
- [2] G. Smith, “Introduction to RIS and PACS,” in *PACS* (K. Dreyer, J. Thrall, D. Hirschorn, and A. Mehta, eds.), pp. 9–25, Springer, 2006.
- [3] Wikipedia, “DICOM — Wikipedia, the free encyclopedia.” <http://en.wikipedia.org/w/index.php?title=DICOM&oldid=955326347>, 2020. [Spletna različica; dostop 19. julija 2020].
- [4] S. Jodogne and Orthanc community, “Orthanc.” <https://www.orthanc-server.com/>, 2012–now. Odprtokodna programska oprema. [Spletna različica; dostop 19. julija 2020].
- [5] University Hospital of Geneva, “Weasis medical viewer.” <https://nroduit.github.io/en/>. Odprtokodna programska oprema. [Spletna različica; dostop 19. julija 2020].

[6] R Studio, "Shiny from R Studio." <https://shiny.rstudio.com/>. Odprtokodna programska oprema. [Spletna različica; dostop 19. julija 2020].