



Erasmus+



AGENCIJA ZA
MOBILNOST I
PROGRAME EU

Co-funded by the
Creative Europe Programme
of the European Union



ACTIVE LEARNING THROUGH IMPROVED INTERACTIVITY

Project No. 2020-1-HR01-KA226-SCH-094735

AUDIT

Architecture and software design

Juraj Petrović, Ph.D.

University of Zagreb, Faculty of Electrical Engineering and
Computing

v2.0

January 2022

Contents

General.....	3
Architecture	4
Software design	7
AUDIT web application	7
AUDIT textual answers redirection application	9

General

This document describes the architecture and the design and structure of the AUDIT application. While the system architecture is general and should provide opportunities for different implementations, the design and structure describes the decisions and properties of the system implementation for the purpose of the Active learning through improved interactivity Erasmus+ project.

Ovaj dokument opisuje arhitekturu te dizajn i strukturu AUDIT aplikacije. Iako je arhitektura sustava općenita i trebala bi pružiti mogućnosti za različite implementacije, dizajn i struktura opisuju odluke i svojstva implementacije sustava u svrhu Aktivnog učenja kroz poboljšanu interaktivnost Erasmus+ projekta.

Architecture

The AUDIT web application will follow the three-tier architecture based on the classical client-server architecture pattern. As usual, the system architecture will be formed of

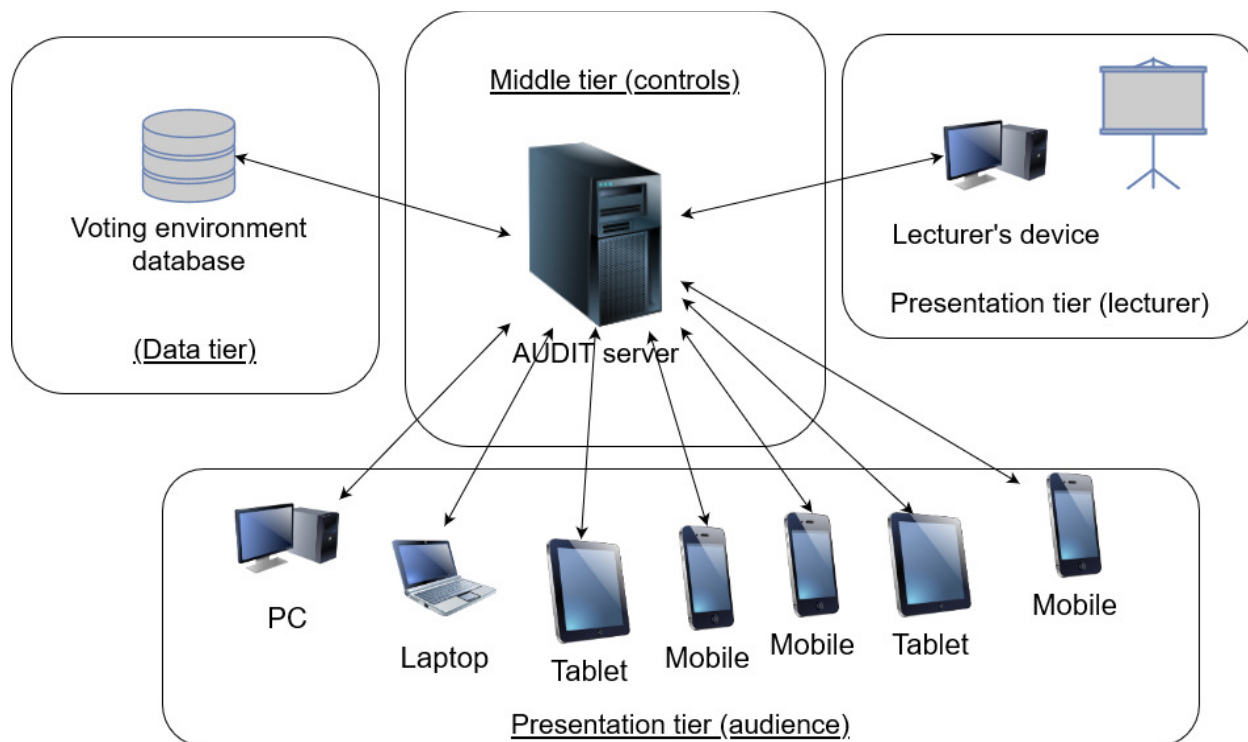
- Presentation tier (view, front end, rendered in the browser) presenting the lecturers with the data display and controls for manipulating it as well as presenting the audience with the voting interface
- Middle tier (controller), implementing controls for data sending, retrieving, and managing, as well as server maintenance
- Data tier (model, back-end database)

The system is graphically presented in the Figure 1.

Web aplikacija AUDIT slijedit će troslojnu arhitekturu temeljenu na klasičnom obrascu arhitekture klijent-poslužitelj. Kao i obično, arhitektura sustava će se formirati od

- Prezentacijski sloj (pregled, prednji kraj, renderiran u pregledniku) koji predavačima predstavlja prikaz podataka i kontrole za manipuliranje njime, kao i predstavljanje publici sučelja za glasanje
- Srednja razina (kontrolor), implementacija kontrola za slanje podataka, dohvaćanje i upravljanje, kao i održavanje poslužitelja
- Razina podataka (model, pozadinska baza podataka)

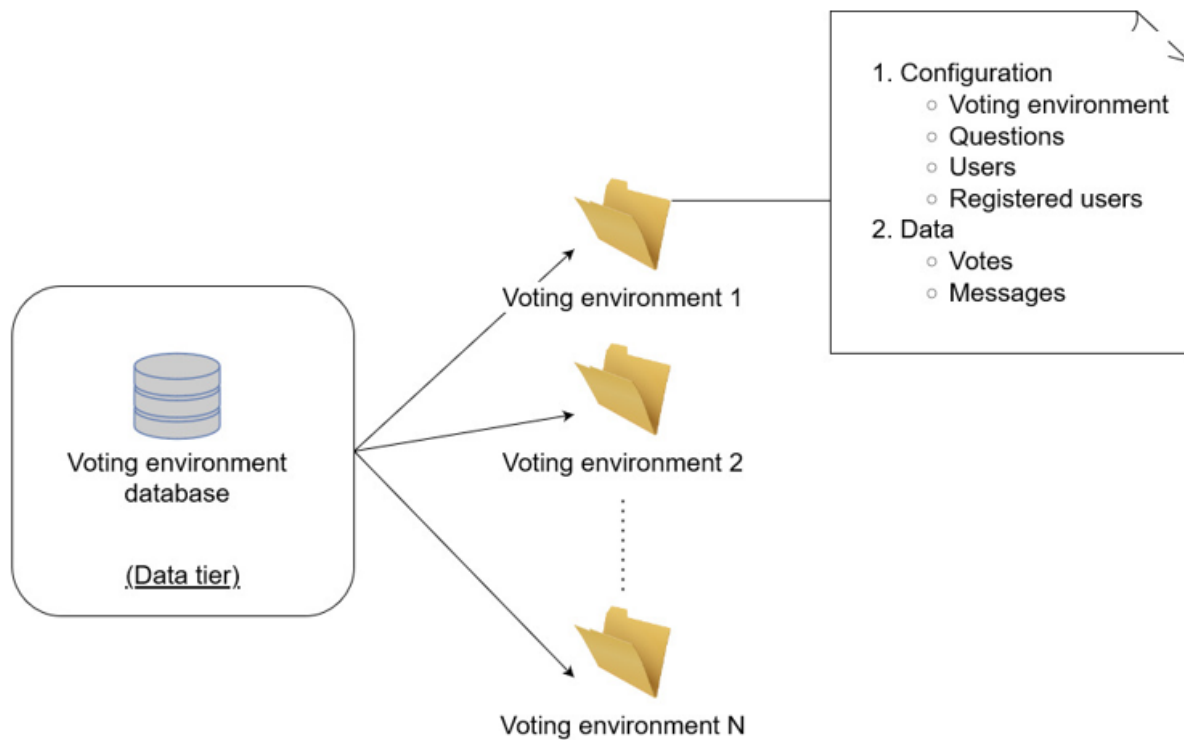
Sustav je grafički prikazan na slici 1.



Slika1. / Figure 1. System architecture

Presentation tier of the application enables the lecturer to view/display the acquired data and to configure display properties using links to the control tier functions. For the audience, the display tier enables them to use the voting interface. By accessing the tier controls through them, the system updates the data tier with the audiences' data. The data tier is presented in more detail in Figure 2.

Prezentacijski sloj aplikacije omogućuje predavaču pregled/prikaz dobivenih podataka i konfiguriranje svojstava prikaza korištenjem poveznica na funkcije kontrolne razine. Za publiku, razina prikaza omogućuje im korištenje sučelja za glasanje. Pristupajući kontrolama razine putem njih, sustav ažurira podatkovnu razinu s podacima o publici. Razina podataka detaljnije je prikazana na slici 2.



Slika2. /Figure 2. Data tier architecture

Software design

AUDIT web application

The AUDIT web application, following the recommendations in the software requirements specification documents will be implemented using HTML/PHP, JavaScript and CSS languages. To facilitate segmenting the AUDIT web application into layers in order to achieve the design as modular as possible, the main application modules and their functions will be the following:

- Lecturer's presentation tier will be implemented in the *room.php* file/script. This script which will take care of the data display and use the voting environment database API to get and process server data. Display options will be configured either on the client side using JavaScript (thereby avoiding unnecessary POST requests to server and page reloading) or by using the interface to middle layer functions which will enable data processing and acquisition through POST requests. Functions controlled through JavaScript are display show/hide and switching between bar and pie chart. Chart drawing will be handled using charts.js library. Chart data input will be provided by the *processAnswers.php* script. Functions which require POST requests to server include answers accept on/off (since that information has to be written into the question configuration file so audience clients can access it), refresh (since data has to be acquired from server), timer (since when expired the timer must disable voting), the number of offered answers (since that information has to be written into the question configuration file so audience clients can access it), as well as which answer from which user to show (first, last, all answers from registered, unregistered, all users). Opening next or going back to a previous question will be managed through POST requests of the *room.php* script. Additional options on this page will include an interface to upload a list of users, delete votes to the current question (POST), and download data associated with the voting environment. Following the proposed architectural design, the lecturer won't access the voting environment data directly, but through functions of the middle layer.
- Audience's presentation tier will be implemented in *index.php* page in */s/* subfolder of the AUDIT web application. Initially the script provides its users with a login interface where an audience member should write the ID of the voting environment they want to access. Once they have accessed it, the interface consists of a section with buttons and section with a text area to write a textual response. Answer history is displayed below on the interface. If the user has registered, their name or other ID provided by the lecture is visible on top of the interface. To maximize voting efficiency and minimize server workload per submitted vote, voting will technically be implemented as a POST HTTP request and subsequential appending a line to a csv answers file.

Other AUDIT application scripts will include:

- *index.php* – main AUDIT page for choosing between the lecturer and audience role or getting help or information about the tool/project

- `lecturer.php` – the page for setting up a voting environment for the lecturer. Lecturer's options include opening a new or re-opening an active room. A new room can be opened anonymously or using email/password credentials. Anonymous rooms can't be re-accessed later. Registered rooms can be accessed again using room number and email/password. A room remains active on the system for 4 weeks and is deleted if there is no activity (measured as file modification timestamps) for 4 weeks.
- `roomPurge.php` – middle layer room purge utility that will check modification timestamps for all rooms and be called whenever a lecturer tries to open or access a room
- `sessionManagement.php` – the control of a session. PHP session will be started the first time a user accesses the *lecturer.php* page and is maintained for 90 minutes since the last user's activity (page refresh or HTTP request from the system)
- `processAnswers.php` – the main middle layer function for processing answers acquired in the text/ and votes/ subfolders, to assess answer similarity or uniqueness, and to prepare them for display
- `downloadAnswers.php` – script for downloading or having the results emailed for later analysis

Web aplikacija AUDIT, slijedeći preporuke u dokumentima specifikacije softverskih zahtjeva bit će implementirana korištenjem HTML/PHP, JavaScript i CSS jezika. Kako bi se olakšalo segmentiranje web aplikacije AUDIT na slojeve kako bi se dizajn postigao što je više modularniji, glavni moduli aplikacije i njihove funkcije bit će sljedeće:

- Razina prezentacije predavača bit će implementirana u datoteci/skripti `room.php`. Ova skripta koja će se pobrinuti za prikaz podataka i koristiti API baze podataka glasačkog okruženja za dobivanje i obradu podataka poslužitelja. Opcije prikaza bit će konfigurirane ili na strani klijenta pomoću JavaScripta (čime se izbjegavaju nepotrebni POST zahtjevi poslužitelju i ponovno učitavanje stranice) ili korištenjem sučelja za funkcije srednjeg sloja koje će omogućiti obradu i akviziciju podataka putem POST zahtjeva. Funkcije kojima se upravlja putem JavaScripta su prikaz prikaza/sakrij i prebacivanje između trakastog i tortnog grafikona. Crtanje grafikona bit će obrađeno pomoću biblioteke `charts.js`. Unos podataka grafikona bit će osiguran skriptom `processAnswers.php`. Funkcije koje zahtijevaju POST zahtjeve poslužitelju uključuju uključivanje/isključivanje prihvatanja odgovora (budući da se te informacije moraju upisati u konfiguracijsku datoteku pitanja kako bi joj klijenti publike mogli pristupiti), osvježavanje (budući da se podaci moraju preuzeti s poslužitelja), mjerač vremena (od kada istekao, mjerač vremena mora onemogućiti glasanje), broj ponuđenih odgovora (budući da se te informacije moraju upisati u konfiguracijsku datoteku pitanja kako bi joj klijenti publike mogli pristupiti), kao i odgovor kojeg korisnika prikazati (prvi, zadnji, svi odgovori registriranih, neregistriranih, svih korisnika). Otvaranjem sljedećeg ili vraćanjem na prethodno pitanje upravljat će se putem POST zahtjeva skripte `room.php`. Dodatne opcije na ovoj stranici uključivat će sučelje za prijenos popisa korisnika, brisanje glasova za trenutno pitanje (POST) i preuzimanje podataka povezanih s okruženjem za glasanje. Slijedom predloženog arhitektonskog rješenja, predavač neće pristupiti podacima glasačkog okruženja izravno, već putem funkcija srednjeg sloja.
- Razina prezentacije publike bit će implementirana na stranici `index.php` u podmapi `/s/` web aplikacije AUDIT. U početku skripta svojim korisnicima pruža sučelje za prijavu gdje bi član publike trebao napisati ID glasačkog okruženja kojem želi pristupiti. Nakon što mu pristupe, sučelje se sastoji od odjeljka s

gumbima i odjeljka s tekstualnim područjem za pisanje tekstualnog odgovora. Povijest odgovora prikazana je ispod na sučelju. Ako se korisnik registrirao, na vrhu sučelja vidljivo je njegovo ime ili drugi ID koji je naveden u predavanju.

Kako bi se povećala učinkovitost glasovanja i smanjilo opterećenje poslužitelja po poslanom glasu, glasovanje će se tehnički implementirati kao POST HTTP zahtjev i naknadno dodavanje retka u csv datoteku odgovora.

Ostale skripte aplikacije AUDIT uključivat će:

- index.php – glavna stranica AUDIT za odabir između uloge predavača i publike ili dobivanje pomoći ili informacija o alatu/projektu
- lecturer.php – stranica za postavljanje okruženja za glasanje za predavača. Mogućnosti predavača uključuju otvaranje nove ili ponovno otvaranje aktivne sobe. Nova soba se može otvoriti anonimno ili pomoću vjerodajnica e-pošte/lozinke. Anonimnim sobama kasnije se ne može ponovno pristupiti. Registriranim sobama može se ponovno pristupiti pomoću broja sobe i e-pošte/lozinke. Soba ostaje aktivna na sustavu 4 tjedna i briše se ako nema aktivnosti (mjereno kao vremenske oznake izmjene datoteke) 4 tjedna.
- roomPurge.php – uslužni program za čišćenje sobe srednjeg sloja koji će provjeriti vremenske oznake modifikacije za sve sobe i biti pozvan kad god predavač pokuša otvoriti prostoriju ili pristupiti njoj
- sessionManagement.php – kontrola sesije. PHP sesija će se pokrenuti prvi put kada korisnik pristupi stranici lecturer.php i održava se 90 minuta od posljednje aktivnosti korisnika (osvježavanje stranice ili HTTP zahtjev iz sustava)
- processAnswers.php – glavna funkcija srednjeg sloja za obradu odgovora dobivenih u podmapama text/ i votes/, za procjenu sličnosti ili jedinstvenosti odgovora i pripremu za prikaz
- downloadAnswers.php – skripta za preuzimanje ili slanje rezultata e-poštom za kasniju analizu

AUDIT textual answers redirection application

The part of the AUDIT application that will redirect textual messages will be implemented as a desktop application in Java. The application will connect to a secret URL provided to the lecturer in the presentation tier. The lecturer will copy that URL into AUDIT redirection application and select a profile for supplementing textual messages from the audience with additional keyboard characters or key combinations need by the final destination application. Once the start key is pressed the application will

1. connect to the secret AUDIT URL
2. repeatedly read textual messages from it in plaintext/CSV format used by the web application

3. if any new messages have arrived, the application will re-write them using the Java Robot class as if they were typed from the keyboard from the lecturer's machine
4. the application will keep doing steps 2-3 until the stop key is pressed

Dio AUDIT aplikacije koji će preusmjeravati tekstualne poruke bit će implementiran kao desktop aplikacija u Javi. Aplikacija će se povezati s tajnim URL-om koji je dostavljen predavaču na razini prezentacije. Predavač će kopirati taj URL u aplikaciju za preusmjeravanje AUDIT i odabrati profil za dopunu tekstualnih poruka publike s dodatnim znakovima tipkovnice ili kombinacijama tipki koje su potrebne za krajnju odredišnu aplikaciju. Nakon što se pritisne tipka za početak, aplikacija će

1. povežite se s tajnim URL-om AUDIT
2. više puta čitati tekstualne poruke iz njega u običnom/CSV formatu koji koristi web aplikacija
3. ako su pristigle nove poruke, aplikacija će ih ponovno napisati koristeći klasu Java Robot kao da su upisane s tipkovnice s predavača
4. aplikacija će nastaviti raditi korake 2-3 sve dok se ne pritisne tipka za zaustavljanje