



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

General description, software specification, and
implementation recommendations

v2.0

January 2022

Contents

General description.....	3
Overview	3
Context and goals	4
Software specification.....	5
User functions	5
Other functions and requirements	8
Testing.....	8
Function and usability testing	8
Workload testing.....	10
Implementation recommendations.....	10
Implementation technologies.....	10
Design recommendations	12

General description

Overview

The general description and properties of the AUDIT system are defined within the project proposal of the *Active learning through improved interactivity* Erasmus+ project. AUDIT is foremost intended to be used in lectures to facilitate lecture interactivity by serving as an additional communication channel with functions which will help lecturer get feedback or answers from their audience. In the wider context of the project, the AUDIT application will be used in the methodological examples describing efficient usage of technology in the classroom.

AUDIT will be an audience response system supporting the interaction between the lecturer and their audience. It will enable the lecturer to acquire and display information obtained from their audience. The audience members will be able to send information to the lecturer either as votes (ABCDE answers, typically answers to lecturer's multiple-choice questions) or as free text responses (unconstrained textual answers to lecturer's questions or questions of audience members) using their mobile devices with Internet connection. Audience members will be able to answer lecturer's questions either as registered users (under their name or given identity) or anonymously. The lecturer will be able to display the acquired data as a chart or text and configure display properties including chart type, include first, last or all votes in the chart, cluster textual answers based on their similarity, and control the voting process (start, stop, delete votes, hide or show chart or set voting timer). The lecturer will also be able to redirect all textual messages from his audience to an application of their choice as if they were typed on the lecturer's computer to support real-time collaboration in applications like concept mapping tools.

AUDIT will be developed by University of Zagreb, Faculty of electrical engineering and computing under supervision of Aquilonis d.o.o. (consortium leader) and other project partners (Zavod Antona Martina Slomškega, Slovenia; Srednja Ekonomska Škola I Gimnazija Maribor, Slovenia; Tartu Kurtehariduskeskus, Estonia).

Opći opis i svojstva AUDIT sustava definirani su unutar projektnog prijedloga Erasmus+ projekta Aktivno učenje kroz poboljšanu interaktivnost. AUDIT je prvenstveno namijenjen upotrebi u predavanjima kako bi se olakšala interaktivnost predavanja služeći kao dodatni komunikacijski kanal s funkcijama koje će predavačima pomoći da dobiju povratne informacije ili odgovore od svoje publike. U širem kontekstu projekta, aplikacija AUDIT koristit će se u metodološkim primjerima koji opisuju učinkovito korištenje tehnologije u učionici.

AUDIT će biti sustav odgovora publike koji podržava interakciju između predavača i njihove publike. To će omogućiti predavaču da dobije i prikaže informacije dobivene od svoje publike. Članovi publike moći će predavaču slati informacije u obliku glasova (ABCDE odgovori, obično odgovori na pitanja s višestrukim izborom predavača) ili kao slobodni tekstualni odgovori (neograničeni tekstualni odgovori

na pitanja predavača ili pitanja članova publike) koristeći svoje mobilne uređaje s internetskom vezom. Članovi publike će na pitanja predavača moći odgovarati kao registrirani korisnici (pod svojim imenom ili pod datim identitetom) ili anonimno. Predavač će moći prikazati dobivene podatke kao grafikon ili tekst i konfigurirati svojstva prikaza uključujući vrstu grafikona, uključiti prve, posljednje ili sve glasove u grafikonu, grupirati tekstualne odgovore na temelju njihove sličnosti i kontrolirati proces glasanja (početak, zaustaviti, izbrisati glasove, sakriti ili pokazati grafikon ili postaviti mjerač vremena za glasanje). Predavač će također moći preusmjeriti sve tekstualne poruke od svoje publike u aplikaciju po svom izboru kao da su upisane na predavačevom računalu kako bi podržao suradnju u stvarnom vremenu u aplikacijama poput alata za mapiranje koncepta.

AUDIT će razviti Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva pod nadzorom Aquilonis d.o.o. (voditelj konzorcija) i drugi projektni partneri (Zavod Antona Martina Slomška, Slovenija; Srednja Ekonomska Škola I Gimnazija Maribor, Slovenija; Tartu Kurtehariduskeskus, Estonija).

Context and goals

The aim of this project is to design an audience response system that will facilitate interactivity while still being simple enough to use. Many audience response systems exist like Kahoot, Quizzit, Mentimeter, but their power comes at a cost of complexity and necessity of detailed preparation. AUDIT should support functions that make those audience response system powerful but keep things simple by separating the question posing environment from the virtual voting environment. This way the AUDIT system will support *ad-hoc* usage without the need to register and open an account in advance or upload questions or presentation to another tool. The lecturer will be able to prepare the questions where they find convenient (blackboard, slideshow etc.). The main characteristics of AUDIT have been elaborated in ALTI project workshop and here they are summarized together with related design and architecture decisions. AUDIT will be an open source application to enable further development and improvements.

Cilj ovog projekta je osmisliti sustav odgovora publike koji će olakšati interaktivnost, a pritom biti dovoljno jednostavan za korištenje. Postoje mnogi sustavi odgovora publike kao što su Kahoot, Quizzit, Mentimeter, ali njihova moć dolazi pod cijenu složenosti i potrebe detaljne pripreme. AUDIT bi trebao podržavati funkcije koje čine te sustave odgovora publike moćnim, ali čine stvari jednostavnim odvajanjem okruženja za postavljanje pitanja od virtualnog okruženja za glasanje. Na taj će način AUDIT sustav podržavati ad-hoc korištenje bez potrebe za prethodnom registracijom i otvaranjem računa ili učitavanjem pitanja ili prezentacije na drugi alat. Predavač će moći pripremiti pitanja gdje mu odgovara (ploča, dijaprojeksija itd.). Glavne karakteristike AUDIT-a razrađene su u ALTI projektnoj radionici i ovdje su sažete zajedno sa srodnim dizajnerskim i arhitektonskim odlukama. AUDIT će biti aplikacija otvorenog koda koja će omogućiti daljnji razvoj i poboljšanja.

Software specification

User functions

The AUDIT application will be developed as an online application and as such it will be usable on desktop or laptop computers, but also on any other web-capable device including tablets and smartphones.

From the user's perspective, the application will consist of two major components:

- **The lecturer's component**, where the lecturer will have control over the virtual voting environment (*room*) for collecting information from the audience. Lecturer will be able to:
 - choose interface language
 - open a new voting environment or re-access a previously opened one using their credentials
 - define a list of users for that room
 - to enable users to register and participate under their names
 - to enable display of information acquired from registered vs. unregistered users
 - configure display options
 - choose between *vote* and *text* question/answer type
 - *vote* can have bar graph or pie chart display
 - for both displays the lecturer can choose between first, last or all votes/messages from audience members and combine that with only displaying votes from registered, unregistered or all audience members
 - *text* can have text, cluster or word-cloud display
 - for text the lecture can choose between first, last or all votes/messages from audience members and combine that with only displaying votes from registered, unregistered or all audience members. They can choose to display username and/or user message for the selected parameters.
 - for cluster display, the lecturer can choose clustering method (algorithm for assessing answer similarity, the similarity coefficient, or the number of target cluster groups) and whether to display names together with answer groups
 - for word-cloud the user can use a clustering method with same properties as for the clustering display. They can also choose whether to display votes only from registered users and first/last/all options.
 - refresh the charts manually or automatically
 - display or hide charts
 - enable or disable voting
 - delete votes to a question

- open a new question, navigate between questions, and return to previous questions and their data
- configure a timer to automatically disable voting after some time
- enable their audience to access voting environment using a link, voting environment ID, and a QR code
- o redirect user's textual messages to another application as if they were all typed on their computer
 - this application must be implemented as a desktop, cross-platform solution
 - encapsulate entire messages, their text or their username with additional elements (combinations of keyboard controls or keys) that will help their proper arrangement in the target application
- o export and download or have all acquired data sent by email for later analysis

The lecturer's component will NOT provide an interface for posing questions – this will be done elsewhere (for example on a blackboard, lecturer's slideshow, verbally, ...)

- **The audience component**, where audience members will be able to:
 - o access a virtual voting environment based on the voting environment ID provided by the lecturer or by scanning a QR code
 - o choose interface language
 - o register in that voting environment (if the lecturer has uploaded a list of users)
 - o vote anonymously to any question at any time, even if they are registered
 - o submit a textual response or an A/B/C/D/E vote by clicking a button

Aplikacija AUDIT bit će razvijena kao online aplikacija i kao takva bit će upotrebljiva na stolnim ili prijenosnim računalima, ali i na bilo kojem drugom web sposobnom uređaju uključujući tablete i pametne telefone.

Iz perspektive korisnika, aplikacija će se sastojati od dvije glavne komponente:

Komponenta predavača, gdje će predavač imati kontrolu nad virtualnim glasačkim okruženjem (sobom) za prikupljanje informacija od publike. Predavač će moći:

- odaberati jezik sučelja
- otvoriti novo okruženje za glasanje ili ponovno pristupiti prethodno otvorenom koristeći njihove vjerodajnice
- definirati popis korisnika za tu sobu
- omogućiti korisnicima da se registriraju i sudjeluju pod svojim imenom
- omogućiti prikaz informacija dobivenih od registriranih naspram neregistriranih korisnika
- konfigurirajte opcije prikaza
- birati između glasanja i vrste tekstualnog pitanja/odgovora
- glasanje može imati trakasti ili tortni grafikon

- za oba prikaza predavač može birati između prvih, zadnjih ili svih glasova/poruka članova publike i kombinirati to s prikazom samo glasova registriranih, neregistriranih ili svih članova publike
- tekst može imati tekstualni, cluster ili prikaz u oblaku riječi
- za tekst predavanje može birati između prvih, zadnjih ili svih glasova/poruka članova publike i kombinirati to s prikazom samo glasova registriranih, neregistriranih ili svih članova publike. Oni mogu odabrati prikaz korisničkog imena i/ili korisničke poruke za odabrane parametre.
- za prikaz klastera, predavač može odabrati metodu grupiranja (algoritam za procjenu sličnosti odgovora, koeficijenta sličnosti ili broja ciljnih skupina klastera) i hoće li prikazati imena zajedno s grupama odgovora
- za word-cloud korisnik može koristiti metodu grupiranja s istim svojstvima kao i za prikaz grupiranja. Također mogu odabrati hoće li prikazati glasove samo registriranih korisnika i prve/zadnje/sve opcije.
- osvježite karte ručno ili automatski
- prikazati ili sakriti grafikone
- omogućiti ili onemogućiti glasovanje
- brisanje glasova za pitanje
- otvorite novo pitanje, navigirajte između pitanja i vratite se na prethodna pitanja i njihove podatke
- konfigurirajte mjerač vremena za automatsko onemogućavanje glasovanja nakon nekog vremena
- omogućiti svojoj publici pristup okruženju za glasanje pomoću veze, ID-a glasačkog okruženja i QR koda
- preusmjerite tekstualne poruke korisnika na drugu aplikaciju kao da su sve upisane na računalu
- ova aplikacija mora biti implementirana kao stolno, višepatformsko rješenje
- enkapsulirati cijele poruke, njihov tekst ili njihovo korisničko ime s dodatnim elementima (kombinacijama kontrola na tipkovnici ili tipkama) koji će pomoći njihovom pravilnom rasporedu u ciljnoj aplikaciji
- izvoz i preuzimanje ili slanje svih prikupljenih podataka putem e-pošte za kasniju analizu

Predavačeva komponenta NEĆE pružiti sučelje za postavljanje pitanja – to će se učiniti negdje drugdje (na primjer na ploči, prezentaciji predavača, usmeno,...)

Komponenta publike, gdje će publika moći:

- pristupiti virtualnom okruženju za glasanje na temelju ID-a glasačkog okruženja koji je dao predavač ili skeniranjem QR koda
- odaberati jezik sučelja
- registrirati se u tom glasačkom okruženju (ako je predavač prenio popis korisnika)
- anonimno glasati za bilo koje pitanje u bilo koje vrijeme, čak i ako su registrirani
- podnesti tekstualni odgovor ili A/B/C/D/E glas klikom na gumb

Other functions and requirements

The system must support a large number of concurrent users (minimally 500 registered users and real-time processing of 500 submitted votes/messages in 100 seconds).

The system must automatically archive and delete data of voting environments older than 30 days.

The system must prevent unauthorized access to virtual voting environments.

The system must be easy to setup and maintain with as few additional install packages as possible.

Display chart must be responsive to window size so that it can be maximized and be clearly visible to audience members if displayed on a digital image projector.

Sustav mora podržavati veliki broj istodobnih korisnika (minimalno 500 registriranih korisnika i obrada 500 poslanih glasova/poruka u stvarnom vremenu u 100 sekundi).

Sustav mora automatski arhivirati i brisati podatke glasačkih okruženja starijih od 30 dana.

Sustav mora spriječiti neovlašteni pristup virtualnim okruženjima za glasanje.

Sustav mora biti jednostavan za postavljanje i održavanje sa što manje dodatnih instalacijskih paketa.

Prikaz grafikona mora odgovarati veličini prozora kako bi se mogao maksimalno povećati i biti jasno vidljiv članovima publike ako se prikazuje na projektoru digitalnih slika.

Testing

Function and usability testing

Function and usability testing will be performed through several typical usage scenarios:

Regular registered class

1. The lecturer will open a new room and register it with their email address
2. The lecturer will upload a list of users and invite his audience to register in the room
3. The lecturer will invite their audience to answer a multiple-choice question, hide the results display, disable the voting after 60 seconds and display the results as a bar graph and pie chart
4. The lecturer will try to affect the audience members' opinions and invite them to vote again.
The lecturer will display the results of the first and then the last vote

5. The lecturer will move on to a new question and invite the audience to answer a question in free text. They will display the answers in plain text, only answers for registered users, only answers for registered users but without username displayed.
 6. The lecturer will move to a new question and invite the audience to answer a question with an anonymous short text answer. They will display answers clustered into similar groups for different values of similarity coefficient.
 7. The lecturer will download the acquired answers to all questions.
- Audience members will access the room, register, and submit the required format of answer
 - Audience members will check their submitted answers in the user interface

Ad-hoc unregistered class

1. The lecturer will open a new, unregistered room and invite their audience to answer a question
2. The audience will access the room and vote without registration

Usability testing will be performed using the “think-aloud method” with the described scenarios with at least 5 users in accordance with good practices.

Testiranje funkcionalnosti i upotrebljivosti

Testiranje funkcije i upotrebljivosti provodit će se kroz nekoliko tipičnih scenarija upotrebe:

Redovni upisani razred

1. Predavač će otvoriti novu prostoriju i registrirati je svojom e-mail adresom
2. Predavač će učitati popis korisnika i pozvati svoju publiku da se prijavi u sobu
3. Predavač će pozvati svoju publiku da odgovori na pitanje s višestrukim izborom, sakri prikaz rezultata, onemogućiti glasanje nakon 60 sekundi i prikaže rezultate kao trakasti i tortni grafikon
4. Predavač će pokušati utjecati na mišljenje članova publike i pozvati ih da ponovno glasaju. Predavač će prikazati rezultate prvog, a zatim posljednjeg glasovanja
5. Predavač će prijeći na novo pitanje i pozvati publiku da odgovori na pitanje u slobodnom tekstu. Prikazat će odgovore u običnom tekstu, samo odgovore za registrirane korisnike, samo odgovore za registrirane korisnike, ali bez prikazanog korisničkog imena.
6. Predavač će prijeći na novo pitanje i pozvati publiku da odgovori na pitanje anonimnim kratkim tekstualnim odgovorom. Prikazat će odgovore grupirane u slične skupine za različite vrijednosti koeficijenta sličnosti.
7. Predavač će preuzeti stečene odgovore na sva pitanja.

- Članovi publike će pristupiti prostoriji, registrirati se i dostaviti traženi format odgovora

- Članovi publike će provjeriti svoje poslane odgovore u korisničkom sučelju

Ad-hoc neregistrirani razred

1. Predavač će otvoriti novu, neregistriranu sobu i pozvati svoju publiku da odgovori na pitanje
2. Publika će pristupiti prostoriji i glasovati bez registracije

Testiranje upotrebljivosti provodit će se “metodom razmišljanja naglas” s opisanim scenarijima s najmanje 5 korisnika u skladu s dobrim praksama.

Workload testing

The system must be able to support accepting a minimum of 500 submitted votes or textual messages (max size of 100 characters per message) in 100 seconds. This will be tested using the <https://loader.io/> service. This will be achieved by optimizing code and reducing server workload (especially regarding users' requests) but also, if needed, by managing web server configuration. Instructions on optimal server configuration and installation will be provided in the documentation.

Sustav mora podržavati prihvaćanje najmanje 500 poslanih glasova ili tekstualnih poruka (maksimalna veličina od 100 znakova po poruci) u 100 sekundi. Ovo će se testirati pomoću usluge <https://loader.io/>. To će se postići optimizacijom koda i smanjenjem radnog opterećenja poslužitelja (osobito u pogledu zahtjeva korisnika), ali i, ako je potrebno, upravljanjem konfiguracijom web poslužitelja. Upute za optimalnu konfiguraciju i instalaciju poslužitelja bit će navedene u dokumentaciji.

Implementation recommendations

What follows are the recommendations for implementation of the software specification.

Implementation technologies

- PHP 7.4+
Due to recent significant improvements in PHP version 7 upwards, PHP is again a very attractive language for generating HTML output with significant improvements in library availability, speed, and community of developers. It is recommended, therefore, that it is the main back-end

development technology for this project. Alternatives, if necessary, include React JS, Django/Python, and Ruby.

- JavaScript

The most widely supported choice for web application front-end managing. Pure JavaScript is recommended or minimizing JavaScript libraries not to bloat the users with unnecessary code. Charts should be implemented in JavaScript using Chart.js well maintained library.

- CSS

For styling web pages. No additional CSS frameworks will probably be needed.

- Plaintext files for voting environment data storage (instead of SQL)

With a typical classroom consisting of <30 users or large enrolment classes with <250 users per class, advantages of SQL in speed and data manipulation will not exceed its downsides in terms of maintenance. For this specific application storing data in CSV files should be simple and efficient as well as practical for error handling or data rescue

- Java

For implementing the separate part of the application for redirecting acquired textual answers to a different application as if they were typed on the lecturer's computer. Java enables cross-platform applications. Target operating systems are Windows, Linux and MacOS.

Preporuke za implementaciju softverske specifikacije.

Tehnologije implementacije

- PHP 7,4+

Zbog nedavnih značajnih poboljšanja u PHP verziji 7 navise, PHP je opet vrlo atraktivan jezik za generiranje HTML izlaza sa značajnim poboljšanjima u dostupnosti knjižnice, brzini i zajednici programera. Stoga se preporučuje da to bude glavna pozadinska razvojna tehnologija za ovaj projekt. Alternative, ako je potrebno, uključuju React JS, Django/Python i Ruby.

- JavaScript

Najšire podržan izbor za upravljanje front-edom web aplikacija. Preporuča se čisti JavaScript ili minimiziranje JavaScript knjižnica kako se korisnici ne bi nadimali nepotrebnim kodom. Grafikoni bi se trebali implementirati u JavaScriptu koristeći dobro održavanu knjižnicu Chart.js.

- CSS

Za stiliziranje web stranica. Vjerojatno neće biti potrebni nikakvi dodatni CSS okviri.

- Datoteke otvorenog teksta za pohranu podataka okoline glasanja (umjesto SQL)

Uz tipičnu učionicu koja se sastoji od <30 korisnika ili velikih razreda s <250 korisnika po razredu, prednosti SQL-a u brzini i manipulaciji podacima neće premašiti njegove nedostatke u smislu održavanja. Za ovu specifičnu aplikaciju pohrana podataka u CSV datotekama trebala bi biti jednostavna i učinkovita te praktična za rukovanje pogreškama ili spašavanje podataka

- Java

Za implementaciju posebnog dijela aplikacije za preusmjeravanje stečenih tekstualnih odgovora u drugu aplikaciju kao da su upisani na računalu predavača. Java omogućuje višepatformske aplikacije. Ciljani operativni sustavi su Windows, Linux i MacOS.

Design recommendations

- The system architecture should follow the MVC (Model, View, Controller) architecture
- Consequently, voting environment data should be separated from the application logic and consist of a folder containing a voting environment data subfolders
- Each of the subfolders should contain
 - Voting environment general configuration file with data including lecturer's credentials, last access timestamp, interface language, active question number
 - Question configuration file with data including display settings and configured options for each of the questions the lecturer has opened so far based on the configuration of a question before they navigated to the next/previous question
 - Subfolders for textual and for votes data with CSV files for the corresponding data belonging to each question number. This will minimize question data processing workload
 - A file containing a list of all users that can register (if uploaded by the lecturer) and a separate file containing a list of currently registered users
- PHP sessions should be used to control user sessions, specifically:
 - For the lecturer, session variables should include:
 - A variable with the unique identifier of the voting environment that they are currently using
 - A variable with last activity timestamp for session expiry control
 - For the audience, session should include
 - A variable with the unique identifier of the voting environment that they are currently using
 - A variable with last activity timestamp for session expiry control
 - A variable with a unique user identifier (4 digits + 4 letters)
 - A variable with the chosen language setting
 - A variable with the display name of the registered user in the system
- PHP sessions require GDPR compliance. GDPR compliance notification should be implemented before cookie storing
- User session should be prolonged for 90 minutes on every user's interaction with the system
- Voting environments with no new activity in 4 weeks should be automatically be deleted and that function should be activated by opening a new voting environment
- Depending on the system responsiveness and imposed workload on the server
 - data retrieval and processing that is performed by the lecturer's component should be realized either through long polling (by sending an AJAX request for data every 2-5 seconds)

or server-sent events (should AJAX prove to be unpractical because of HTTP request overhead or additional mechanisms that would be needed to prevent display refresh when no new data is available)

- o sending textual messages or A/B/C/D/E votes should be implemented either through simple POST HTTP requests or, alternatively, through AJAX (to reduce server workload by eliminating the need for client page refresh) or web-sockets (to reduce server workload even further by eliminating HTTP request overhead)

- Arhitektura sustava treba slijediti arhitekturu MVC (Model, View, Controller).

- Slijedom toga, podaci o okolini glasanja trebali bi biti odvojeni od logike aplikacije i sastojati se od mape koja sadrži podmape podataka o okolini glasanja

- Svaka od podmapa treba sadržavati

- o Opća konfiguracijska datoteka okruženja za glasanje s podacima uključujući vjerodajnice predavača, vremensku oznaku posljednjeg pristupa, jezik sučelja, broj aktivnog pitanja

- o Konfiguracijska datoteka pitanja s podacima uključujući postavke prikaza i konfigurirane opcije za svako od pitanja koje je predavač do sada otvorio na temelju konfiguracije pitanja prije nego što je prešao na sljedeće/prethodno pitanje

- o Podmape za tekstualne podatke i podatke o glasovima s CSV datotekama za odgovarajuće podatke koji pripadaju svakom broju pitanja. To će minimizirati radno opterećenje obrade podataka pitanja

- o Datoteka koja sadrži popis svih korisnika koji se mogu registrirati (ako ju je učitao predavač) i zasebna datoteka koja sadrži popis trenutno registriranih korisnika

- PHP sesije bi se trebale koristiti za kontrolu korisničkih sesija, posebno:

- o Za predavača, varijable sesije trebaju uključivati:

- ☐ Varijabla s jedinstvenim identifikatorom glasačkog okruženja

- koje trenutno koriste

- ☐ Varijabla s vremenskom oznakom zadnje aktivnosti za kontrolu isteka sesije

- o Za publiku, sesija bi trebala uključivati

- ☐ Varijabla s jedinstvenim identifikatorom glasačkog okruženja

- koje trenutno koriste

- ☐ Varijabla s vremenskom oznakom zadnje aktivnosti za kontrolu isteka sesije

- ☐ Varijabla s jedinstvenim identifikatorom korisnika (4 znamenke + 4 slova)

- ☐ Varijabla s odabranom postavkom jezika

☐ Varijabla s prikaznim imenom registriranog korisnika u sustavu

- PHP sesije zahtijevaju usklađenost s GDPR-om. Obavijest o usklađenosti s GDPR-om trebala bi se implementirati prije pohranjivanja kolačića
- Korisničku sesiju treba produžiti za 90 minuta pri svakoj interakciji korisnika sa sustavom
- Glasačka okruženja bez novih aktivnosti u 4 tjedna trebala bi se automatski izbrisati i tu funkciju treba aktivirati otvaranjem novog okruženja za glasanje
- Ovisno o odzivu sustava i nametnutom radnom opterećenju poslužitelja

o dohvaćanje i obradu podataka koju izvodi komponenta predavača treba ostvariti ili kroz dugo ispitivanje (slanjem AJAX zahtjeva za podacima svakih 2-5 sekundi) ili događaje poslane poslužiteljem (ako se AJAX pokaže nepraktičnim zbog prevelikog opterećenja HTTP zahtjeva ili dodatni mehanizmi koji bi bili potrebni za sprječavanje osvježavanja zaslona kada novi podaci nisu dostupni)

o slanje tekstualnih poruka ili A/B/C/D/E glasova treba biti implementirano ili putem jednostavnih POST HTTP zahtjeva ili, alternativno, putem AJAX-a (kako bi se smanjilo opterećenje poslužitelja eliminacijom potrebe za osvježavanjem klijentske stranice) ili web-utičnica (za dodatno smanjite radno opterećenje poslužitelja eliminacijom pretjeranog opterećenja HTTP zahtjeva)