

Documentul de Proiectare a Soluției Aplicației Software

(Software Design Document)

29 Octombrie, 2019

Jury Duty

Echipă: Achiriloaiei Ana – 341C2

Resul Ebru – 341C5

Monorean Ruxandra - 342C5

Stoica Ovidiu – 343C5

Facultatea de Automatică și Calculatoare, Universitatea Politehnică,
București

Cuprins

Cuprins.....	2
1. Scopul documentului.....	3
2. Conținutul documentului.....	3
3. Modelul datelor.....	3
3.1. Structuri de date globale.....	3
3.2. Structuri de date de legătură.....	3
3.3. Structuri de date temporare.....	3
3.4. Formatul fișierelor utilizate.....	3
3.5. Descrierea bazei de date.....	4
3.5.1. Descrierea tabelelor.....	4
4. Modelul architectural și modelul componentelor.....	6
4.1. Arhitectura sistemului.....	6
4.1.1. Șabloane arhitecturale folosite.....	6
4.1.2. Diagrama de arhitectură.....	6
4.1.3. Diagrama UML.....	7
4.2. Descrierea componentelor.....	7
4.3. Restricțiile de implementare.....	8
4.4. Interacțiunea dintre componente.....	8
5. Modelul interfeței cu utilizatorul.....	8
5.1. Ferestrele Aplicației.....	8
5.1.1. Fereastra Login.....	8
5.1.2. Fereastra Cocursuri diponibile.....	9
5.1.3. Fereastra Rundele unui concurs.....	9
5.1.4. Fereastra Participanților.....	10
5.1.5. Fereastra Votului.....	11
6. Elemente de testare.....	11

1. Scopul documentului:

Acest document a fost creat pentru a descrie într-o manieră detaliată și integrală modul de implementare a soluției software Jury Duty-aplicație de jurizat concursuri(dans, frumusețe, gătit, etc.). Documentul va fi folosit pe post de îndrumar pentru implementarea soluției care va fi utilă pentru echipa de dezvoltare a proiectului.

2. Conținutul documentului:

- ✓ Modelul datelor – prezintă principalele structuri de date folosite, precum și schema bazei de date.
- ✓ Modelul arhitectural și modelul componentelor – prezintă șabloanele arhitecturale folosite, arhitectura sistemului și descrie componentele arhitecturii.
- ✓ Modelul interfeței cu utilizatorul – prezintă interfața cu utilizatorul și succesiunea ferestrelor acesteia.
- ✓ Elemente de testare – prezintă componentele critice și alternative de proiectare a acestora.

3. Modelul datelor:

3.1. Structuri de date globale:

Structurile de date globale sunt reprezentări ale modelelor de date - Jurați, Concurs, Rundă, Serie, Participanți, etc. Aceste modele au o reprezentare comună atât în serverul de back-end cât și pe partea de front-end.

3.2. Structuri de date de legătură:

Pentru componenta client a aplicației, modulul Server comunică cu modulul Baze de date prin intermediul argumentelor de genul identificatori de către Jurați, Participanți, Concursuri, etc. Astfel, modulul Server comandă acțiuni de tipul CRUD modulului de Baze de date trimițându-i valori ale cheilor primare din schema bazei (identificatorii de cadre jurați, participanți, concursuri, runde, etc.). Modulul GUI(aplicație mobilă) afișează datele primite de la serverul ce interacționează cu Baza de date, într-un format ușor de înțeles.

3.3. Structuri de date temporare:

Nu se utilizează structuri de date temporare cu rol important sau presupunând un consum semnificativ de resurse de memorie.

3.4. Formatul fișierelor utilizate:

Înregistrările în Baza de date sunt păstrate într-o formă persistentă în fișiere sqlite3.

3.5. Descrierea bazei de date:

3.5.1. Descrierea tabelelor:

Schema bazei de date cuprinde următoarele tabele:

Juror – reține persoanele înscrise în concurs cu statutul de jurați. Are următoarele coloane:

- Contest – reprezintă concursul la care are voie să voteze juratul respectiv.
- Username – usernameul folosit la logare.
- Parolă – parola folosită la logare.
- Allow – valoare numerică care îl face pe jurat eligibil să voteze sau nu.

Contest – reține datele despre un anumit concurs. Are următoarele coloane:

- Name – numele concursului.
- Type – tipul concursului.
- Password - parola pentru concursul respectiv.
- Current_round – numărul aferent rundei în desfășurare.
- Round_nums – câte runde are concursul.
- Start_time – timpul când începe runda.
- End_start – timpul când se termina runda.

Round – reține datele referitoare rundei de joc. Are următoarele coloane:

- Contest – referință către identificatorul numeric al concursului (tabela Contest).
- Round_no – câte serii are runda respectivă.
- Start_time – timpul când începe runda.
- End_start – timpul când se termina runda.
- Series_no - în a câta serie a concursului e runda respectivă.
- Eliminate - câți concurenți sunt eliminați în runda respectivă.

Series – reține datele referitoare seriei de participanți. Are următoarele coloane:

- Round - referință către identificatorul numeric al rundei (tabela Round).
- Contest – referință către identificatorul numeric al concursului (tabela Contest).
- Jseries_no - a câta serie din concurs este seria respectivă.

- Start_time – timpul când intră pe scenă concurenții din serie.
- End_start – timpul când ies de pe scenă concurenții din serie.

Participant – reține datele aferente participanților. Are următoarele coloane:

- Round - referință către identificatorul numeric al runde (tabela *Round*).
- Serie – referință către identificatorul numeric al seriei (tabela *Series*).
- Contests - reprezintă concursurile la care un participant este înscris.
- First_name – reprezintă prenumele concurentului.
- Last_namei – reprezintă numele concurentului.
- Start_time – timpul când intră pe scenă concurentul.
- End_start – timpul când iese de pe scenă concurentul.
- Status – identifică dacă concurentul a fost sau nu eliminat din concurs.
- Notă – reprezintă nota finalului unei runde, de la toți jurații.
- Vote – reprezintă numărul persoanelor care l-au votat în runda respectivă.
- WhoVoted – reprezintă numele juraților care l-au votat pe participant.

Note – reține notele fiecărui participant. Are următoarele coloane:

- Contest_id – referință către identificatorul numeric al concursului (tabela *Contest*).
- Jurat - identificator al juratului.
- Participant_id - identificator numeric al participantului.
- Ritm - reprezintă nota pentru ritm.
- Coregrafie - reprezintă nota pentru coregrafie.
- Corectitudine - reprezintă nota pentru corectitudine.
- ComponentaArtistică -reprezintă nota pentru componenta artistică.

Grade – reține cât procent reprezintă fiecare notă pentru calcularea notei finale. Are următoarele coloane:

- Contest_id – identificator numeric al concursului.
- Ritm – reprezintă procentul din nota finală acordat pentru ritm.

- Coregrafie – reprezintă procentul din nota finală acordat pentru coregrafie.
- Corectitudine – reprezintă procentul din nota finală acordat pentru corectitudine.
- Componenta Artistică reprezintă procentul din nota finală acordat componentei artistice.

4. Modelul architectural și modelul componentelor:

4.1 Arhitectura sistemului:

4.1.1 Șabloane arhitecturale folosite:

Soluția software actuală a fost proiectată după modelul de arhitectură client-server. În același timp, interfața cu utilizatorul se bazează pe șablonul event-driven architecture.

Componenta server oferă servicii de interacțiune cu Baza de date, și procesarea stării unui concurs. Componenta server comunică pe rețea (prin intermediul socket-urilor) cu aplicațiile client. Astfel, clientul îi transmite serverului voturile juraților, iar serverul, în urma procesării acestora, va transmite clientului starea finală a unei runde sau a concursului.

Aplicația client oferă utilizatorului posibilitatea de a vizualiza concursurile la care acesta este jurat, runda curentă a unui concurs și concurenții pe care acesta trebuie să-i voteze.

4.1.2. Diagrama de arhitectură:

Diagrama de arhitectură expusă mai jos (Fig 1.1) descrie componentele arhitecturii aplicației și relațiile de interacțiune dintre acestea.

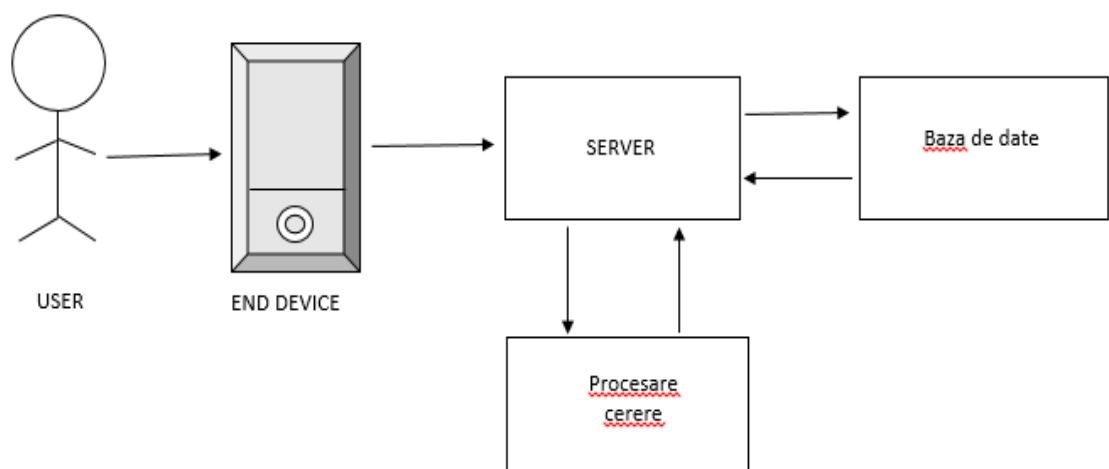


Fig 1.1

4.1.3. Diagrama de sistem:

Diagrama de sistem afișată mai jos (Fig 1.2) reprezintă schema funcționalităților oferite de sistemul software, în termeni de actori, cazuri de utilizare și relații între acestea

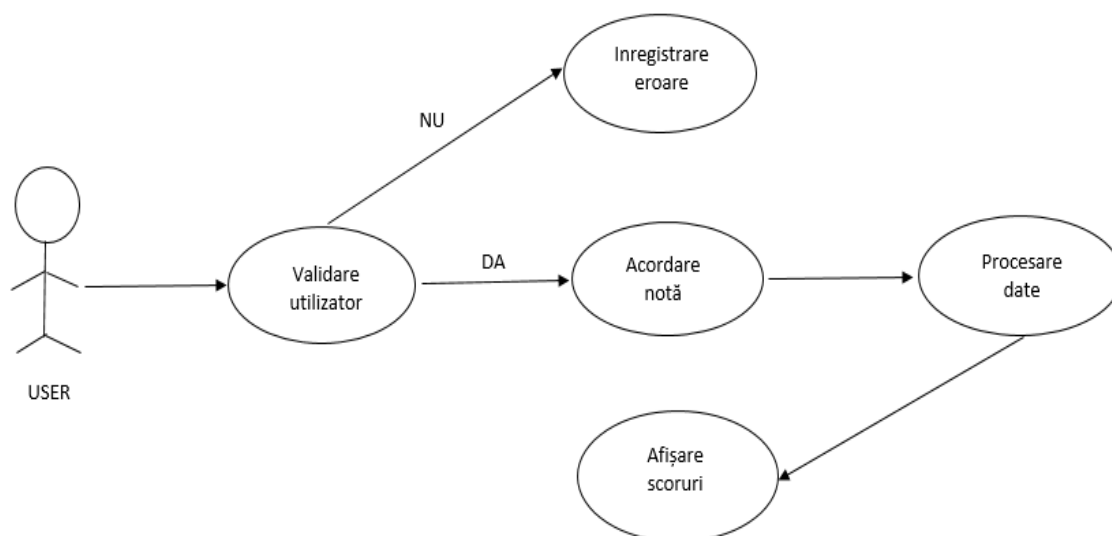


Fig 1.2

4.2. Descrierea componentelor:

Aplicația constă din următoarele module interconectate:

Modulul de Baze de date:

Este responsabil de interacțiunile cu serverul de baze de date, de realizare a conectării, a operațiilor de tip CRUD (Create, Update, Delete) și de rulare a procedurilor stocate.

Modulul GUI (Graphical User Interface):

Este responsabil cu desenarea și randarea optimă a interfețelor grafice ale aplicației.

4.3. Restricțiile de implementare:

Modulele aplicației trebuie să acopere următoarele restricții de implementare:

- Modulul de Baze de date va fi dezvoltat pentru serverul de baze de date sqlite3.
- Toate modulele aplicației vor fi dezvoltate utilizând limbajul de programare Python Edition Versiunea 3.
- Pe partea de client acesta va fi dezvoltat în Javascript folosind frameworkul React Native.

4.4. Interacțiunea dintre componente:

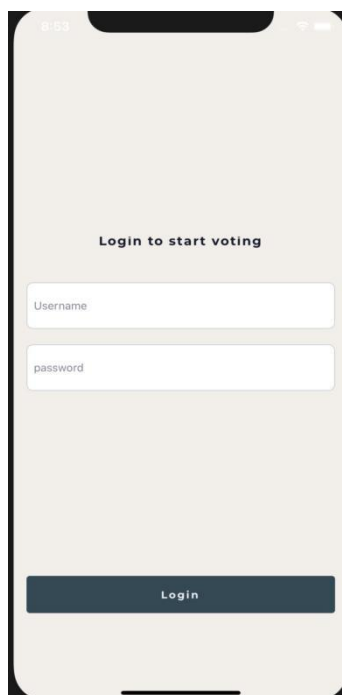
Prin serviciul de autentificare cu token, este asigurată securitatea aplicației, utilizatorul fiind nevoit să se autentifice pentru a putea vota pentru concurs. Toate interacțiunile ulterioare ale clientului cu serverul vor conține tokenul de sesiune pentru a stabili identitatea utilizatorului.

5. Modelul interfeței cu utilizatorul:

5.1. Ferestrele Aplicației:

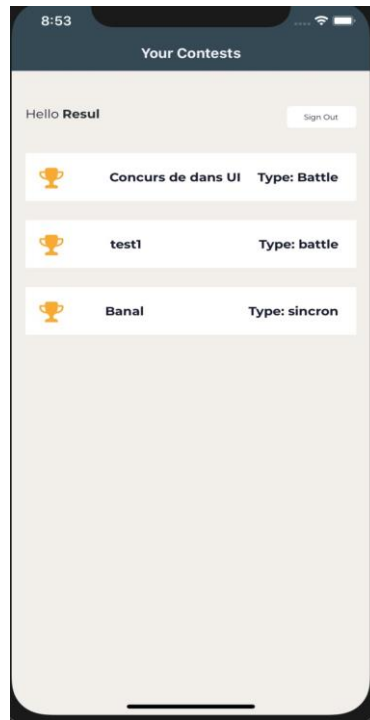
5.1.1. Fereastra Login:

Acestă fereastră este folosită de către jurați pentru logarea în aplicația de vot.



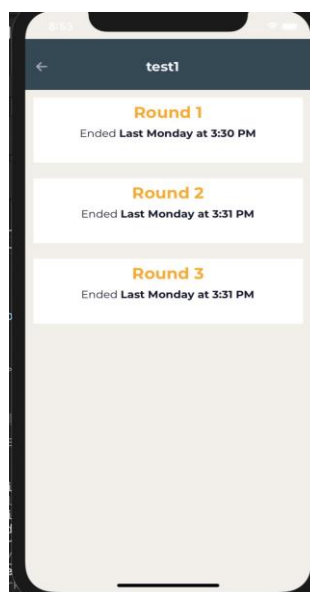
5.1.2. Fereastra Cocursuri disponibile:

Această fereastră conține lista concursurilor disponibile, la care un jurat are dreptul de a vota.



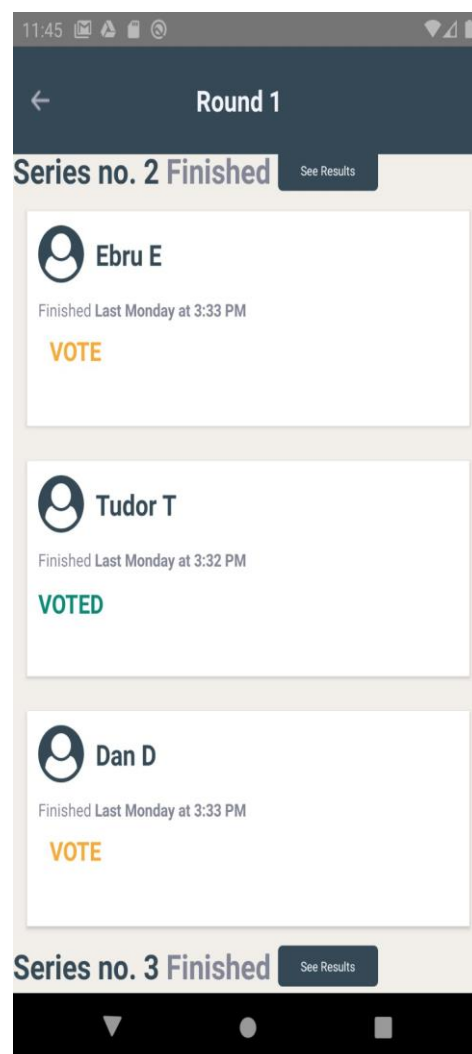
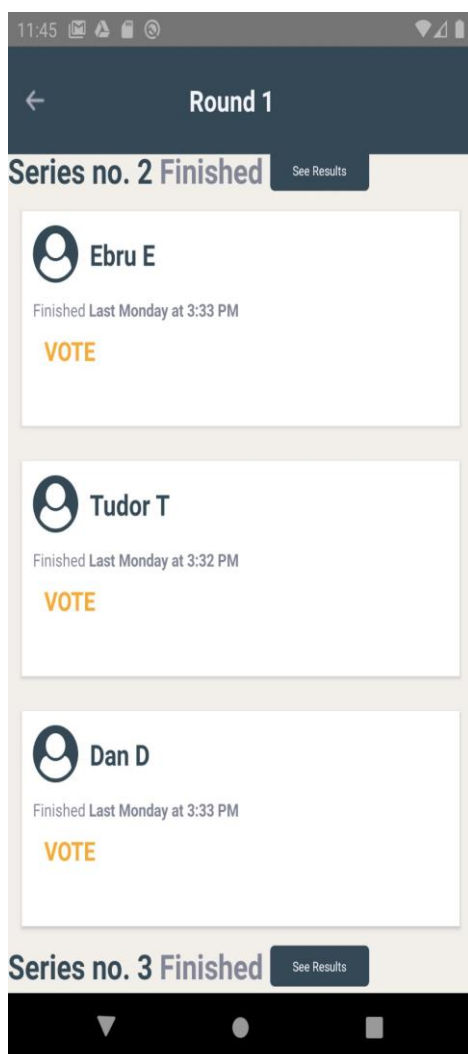
5.1.3. Fereastra Runduile unui concurs:

Această fereastră conține lista rundelor disponibile și timpul când acestea se termină.



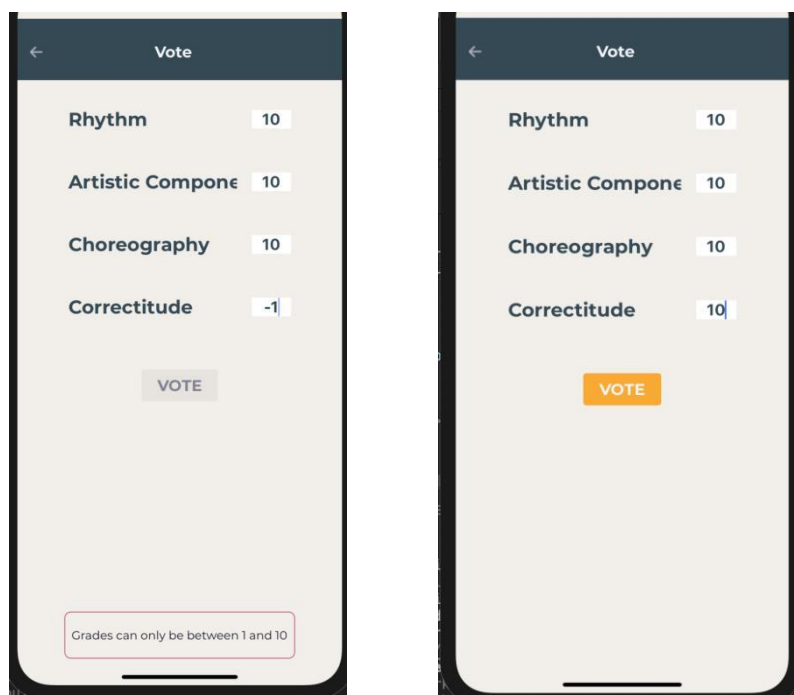
5.1.4. Fereastra Participanților:

Acestă fereastră conține lista concurenților din runda respectiva și seria. Aici se poate observa și statusul fiecărui participant (dacă a fost sau nu votat).



5.1.5. Fereastra Votului:

În fereastră se realizează votarea concurentului pentru fiecare item. Dacă juratul introduce note care nu sunt în intervalul 1-10, este avertizat că nota respectivă nu este o notă validă.



6. Elemente de testare:

Testarea componentelor front end folosind jest.

Testare manuală a tuturor părților ale aplicației și repararea problemelor prin anunțarea responsabilului de respectivul feature.

Evidența bug-urilor se află în desfășurătorul de bug-uri.