**Documentul de Descriere a Design-ului Software**

**Jury Duty**

Ștefănescu Andreea – 342C5

Mormocea Daniela – 342C5

Irimia Tudor – 341C5

Smaranda Alex – 342C5

Telehoi Alex – 342C5

Facultatea de Automatică şi Calculatoare, Universitatea Politehnica, Bucureşti

Cuprins

1. **Scopul documentului**......................................................................................................................3
2. **Obiective**.........................................................................................................................................3
3. **Continutul documentului**...............................................................................................................3
4. **Modelul datelor**..............................................................................................................................3
   1. **Structuri de date globale**..........................................................................................................3
   2. **Structuri de date de legatura**.....................................................................................................................................4
   3. **Structuri de date temporare**....................................................................................................4
   4. **Formatul fisierelor utilizate**......................................................................................................4
   5. **Descrierea bazei de date**..........................................................................................................5
      1. **Diagrama bazei de date**...............................................................................................5
      2. **Descrierea tabelelor**....................................................................................................6
5. **Modelul arhitectural si modelul componentelor**...........................................................................7
   1. **Arhitectura sistemului**..............................................................................................................7
      1. **Sabloane arhitecturale folosite**...................................................................................7
      2. **Diagrama de arhitectura**..............................................................................................7
   2. **Descrierea componentelor**.......................................................................................................9
   3. **Restrictiile de implementare**....................................................................................................9
   4. **Interactiunea dintre componente**............................................................................................9
6. **Modelul interfetei cu utilizatorul**.................................................................................................12
   1. **Succesiunea interfetelor**.........................................................................................................12
   2. **Ferestrele aplicatiei**................................................................................................................12
7. **Elemente de testare**......................................................................................................................12
   1. **Componente critice**................................................................................................................12
   2. **Alternative**..............................................................................................................................13
8. **Scopul documentului**

Acest document are rolul de a descrie detaliat si structurat componentele sistemului software Jury Duty. Mai exact, acesta ofera echipei de dezvoltare detaliile de implementare necesare satisfacerii cerintelor specificate in documentul Software Requirements Specification (SRS).

1. **Obiective**

Obiectivul acestui proiect este realizarea unei aplicatii web, pentru a facilita jurizarea in cadrul unui concurs de gatit. Aplicatia are 4 tipuri de conturi: admin, organizator, jurat si spectator.

1. **Continutul documentului**

Documentul este structurat in 4 sectiuni principale:

* Modelul datelor – prezinta principalele structuri de date folosite, precum si schema bazei de date
* Modelul arhitectural si modelul componentelor – prezinta sabloanele arhitecturale folosite, arhitectura sistemului si descrie componentele arhitecturii
* Modelul interfata cu utilizatorul – prezinta interfata cu utilizatorul si succesiunea ferestrelor acesteia
* Elemente de testare – prezinta componentele critice si alternative de proiectare a acestora

1. **Modelul datelor**
   1. **Structuri de date globale**

Modulele care sunt disponibile tuturor componentelor arhitecturii:

* ***auth –*** incorporeaza metodele pentru autentificare (sign-up si log-in) si metoda pentru oferire de permisiuni a unui utilizator
* ***forms –*** contine o clasa pentru a valida permisiunile
* ***models -*** contine clase pentru instantierea bazei de date pentru diferite entitati (User, Contestant, Jury si Contest)
* ***routes –*** contine metodele pentru implementarea dinamicii jocului
  1. **Structuri de date de legatura**

Pentru comunicatia client-server sunt folosite request-uri HTTP (GET si POST), iar pentru cea dintre server si baza de data este folosit un API.

Pentru facilitarea lucrului cu baza de date este folosit toolkit-ul SQLAlchemy.

* 1. **Structuri de date temporare**

Nu au fost folosite structuri de date temporare cu rol important sau care presupun un consum de resurse semnificativ.

* 1. **Formatul fisierelor utilizate**

Aplicatia foloseste .py ce implementeaza functionalitatile aplicatiei. Aceste fisiere contin cod Python si pot fi executate de catre un interpretor Python.

Un alt format este cel HTML, dat fiind faptul ca interfata cu utilizatorul a fost scris in limbajul cu acelasi nume.

Pentru implementarea mecansimului de votare a fost folosit Javascript, astfel ca fisirele in cauza au formatul .js, iar pentru configurarea aspectului regasim fisiere sub formatul CSS.

* 1. **Descrierea bazei de date**
     1. **Diagrama bazei de date**

O imagine care conține captură de ecran

Descriere generată automat

**4.5.2 Descrierea tabelelor**

* **User –** retine utilizatorii ce folosesc aplicatia. Are urmatoarele coloane:
* id – identificatorul numeric al utilizatorului
* email – adresa de email a utilizatorului
* password – parola utilizatorului
* name – numele utilizatorului
* type\_user – tipul utilizatorului (0 – admin, 1 – organizator,

2 – juriu, 3 – spectator)

* **Contestant –** retine participantii la concurs din cadrul aplicatiei. Are urmatoarele coloane:
* id – identificatorul numeric al participantului
* name – numele participantului
* age – varsta participantului
* description – descrierea participantului
* round\_no – runda din care face parte participantul
* series\_no – seria din care face parte participantul
* grade – nota oferita participantului
* **JuryVoted –** retine juratii care si-au exercitat votul, tabela fiind resetata la inceputul fiecarei runde. Are urmatoarele coloane:
* id – identificatorul numeric al juratului
* jury\_name – numele juratului
* contestant\_name – numele participantului evaluat de catre jurat
* **Contest –** retine competitia ce se desfasoara la un moment dat. Are urmatoarele coloane:
* id – identificatorul numeric al concursului
* name – numele concursului
* rounds – numarul de runde ale concursului
* category1 – prima categorie de jurizare
* category2 – a doua categorie de jurizare
* current\_rounds\_junior – numarul rundei curente pentru juniori
* current\_rounds\_senior – numarul rundei curente pentru seniori
* procent1 – ponderea primei categorii de jurizare
* procent2 – ponderea celei de-a doua categorii de jurizare

1. **Modelul arhitectural si modelul componentelor**
   1. **Arhitectura sistemului**

**5.1.1. Sabloane arhitecturale folosite**

Solutia software actuala a fost proiectata dupa modelulul de arhitectura client-server. In acelasi timp, interfata cu utilizatorul se bazeaza pe sablonul arhitectural bazat pe evenimente.

Componenta server ofera servicii de exploatare a continutului bazei de date, de asemenea inglobata in componenta server, fiind deci o platforma de procesare.

Componenta server comunica pe retea cu aplicatiile client. Astfel, clientul ii transmite serverului informatii necesare pentru desfasurarea competitiei, iar serverul, in urma procesarii acestora, va transmite feedback-ul sau.

Aplicatia client ofera utilizatorului, in functie de permisiunile acestuia, posibilitatea de a vizualiza parcursul concurentilor, de a adauga jurati sau concurenti, de a stabili categoriile de jurizare si ponderea acestora, de a descalifica concurenti.

**5.1.2 Diagrama de arhitectura**

Diagrama de arhitectura de mai jos descrie componentele arhitecturii aplicatiei si relatiile de interactiune dintre acestea.

O imagine care conține text

Descriere generată automat

* 1. **Descrierea componentelor**

Aplicatia consta in urmatoarele module interconectate:

* **Modulul GUI** (Graphical User Interface)

Este responsabil cu desenarea si randarea optima a interfetelor grafice ale aplicatiei.

* **Modulul de Logica**

Este responsabil cu coordonarea celorlalte module astfel incat rezultatele rularii lor sa reflecte logica aplicatiei.

* **Modulul de Baze de date**

Este responsabil de interactiunile cu baza de date, de realizare a conectarii a operatiilor de tip REST, mai exact GET si POST.

* **Modulul de Retea**

Este responsabil de asigurarea schimbului de informatii dintre client si server.

* **Modulul de Raportare**

Este responsabil cu exploatarea informatiilor stocate in baza de date in vederea generarii unui rezultat necesar utiliztorului.

* 1. **Restrictiile de implementare**

Modulele aplicatiei trebuie sa acopere urmatoarele restrictii de implementare:

* Toate modulele aplicatiei vor fi dezvoltate pe baza framework-ului pentru aplicatii web Flask, utilizand limbajul de programare Python 3
* Interfata grafica va fi imlpementata folosind HTML, Bootstrap, Javascript si Jinja 2
* Modulul de Baze de date va fi dezvoltat utilizand toolkit-ul SQLAlchemy specific implementarii SQL in Python
  1. **Interectiunea dintre componente**

Pentru componenta **Client,** atunci cand utilizatorul porneste aplicatia, se va lansa in executie modulul \_\_init\_\_ din cadrul modulului de Logica. Acesta va interactiona cu modulul Retea, care va face o cerere de tip GET catre modulul Baza de date pentru a obtine datele necesare initializarii interfetei grafice. Ulterior, va lansa fereastra de start a aplicatiei – Home din cadrul modulului GUI.

Pasii urmatori depind de tipul utilizatorului si de comanda aleasa de catre acesta din fereasta Home.

In cazul utilizatorului admin, acesta are o optiune prin care poate oferi permisiuni utilizatorilor, dat fiind faptul ca oricine se logheaza in aplicatie are statutul de spectator pana cand acesta primeste noi permisiuni daca este cazul.

Atat pentru admini cat si pentru organizatori este valabila pagina Game Options, unde:

* Adminul poate elimina jurati
* Organizatorul poate alege inceperea sau sfarsitul unei runde si poate elimina concurenti

In cazul utilizatorului de tip organizator, acesta poate accesa pagina Organize pentru a organiza un nou concurs. Acesta trebuie sa introduca numele concursului, nr. de runde al acestuia, categoriile de jurizare si ponderea acestora. Apoi acestuia ii va fi prezentata o pagina unde adauga concurenti, la final fiind redirectionat catre pagina Home. Din acest moment, organizatorul poate accesa pagina Game Options unde poate da start concursului, avand de ales intr doua serii (Juniori sau Seniori).

In momentul in care juratii au terminat de votat, organizatorul trebuie sa sfarseasca runda, fiind redirectionat catre pagina Winners in care vor fi prezente doua tabele (pentru Jr si Sr), fiecare dintre aceste tabele continand concurentii ce au trecut in runda urmatoare.

In cazul utilizatorului de tip jurat, acesta poate accesa doar pagina Home, in care poate vota un concurent doar daca runa acestuia se afla in desfasurare.

Utilizatorul de tip spectator poate accesa paginile Home in care poate vedea parcurusul competitiei si pagina de log-in, unde se poate autentifica.

Aceasta suita de operatii sunt declansate prin lansarea in executie, fie a modulului Retea, fie a modulului Raportare. La sfarsitul procesarilor cerute de utilizator, controlul se va intoarce catre modulul GUI si, ulterior, catre modulul de Logica.

Pentru componenta **Server,** modulul de Retea este cel responsabil de receptionarea informatiilor transmise de componentele client prin retea. Apoi, se va lansa in executie, modulul de Raportare care lucreaza sub directa coordonare a modulului de Logica. La sfarsitul procesarilor, se va transmite comanda modulului de Baze de date spre a fi realizata, fie interogarea bazei de date, fie stocarea datelor procesate in aceasta. La final, controlul se va intoarce catre modulul de Retea, care va transmite feedback aplicatiei Client.

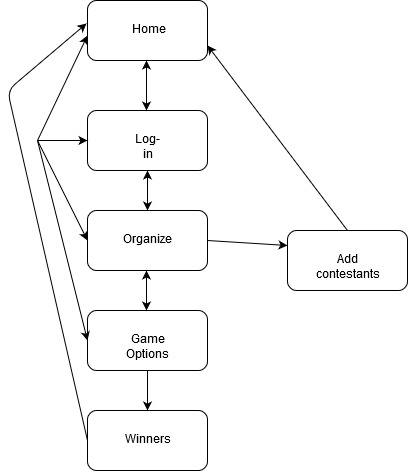
O imagine care conține text

Descriere generată automat

1. **Modelul interfetei cu utilizatorul**
   1. **Succesiunea interfetelor**

In cadrul modului GUI, ferestrele aplicatiei sunt afisare respectand un flux stabilit.

Acest flux este descris in figura de mai jos.

******

* 1. **Ferestrele aplicatiei**

**6.2.1. Fereastra Home**

O imagine care conține captură de ecran

Descriere generată automat

**6.2.2. Fereastra Sign-up/Log-in**

**O imagine care conține captură de ecran

Descriere generată automat**

**6.2.3 Fereastra Give Permisions**

**O imagine care conține captură de ecran

Descriere generată automat**

**6.2.4. Fereastra Organize**

**O imagine care conține captură de ecran

Descriere generată automat**

**6.2.5. Fereastra Game Options**

**O imagine care conține captură de ecran

Descriere generată automat**

**6.2.6. Fereastra Add Contestants**

**O imagine care conține captură de ecran

Descriere generată automat**

1. **Elemente de testare**
   1. **Componente critice**

Componenta critica a aplicatiei este intereactiunea cu server-ul. Fara ca server-ul sa fie functional, aplicatia este nefunctionala, deoarece stocarea si generarea informatiilor este facuta de catre server.

Deoarece functionalitatea aplicatiei depinde de abilitatea de a solicita informatii serever-ului, este necesar ca acest lucru sa functioneze conform asteptarilor.

Procedurile de testare vor valida corectitudinea datelor trimise de la server si primirea lor integral si fara erori.

* 1. **Alternative**

Una dintre alternativele folosirii server-ului ar fi renuntarea la arhitectura client- server in intregime, o solutie fiind cea a folosirii unei arhitecturi distribuite, de tip peer- to-peer, in care fiecare computer este deopotriva client cat si server, astfel ca in eventualitatea defectiunii unui server, datele vor fi inca disponibile pe celelalte calculatoare din retea.