**Agario Game**

**Echipa:**

**Băbuț Rareș Ovidiu**

**Ilisei Elena-Denisa**

**Grupa 30232**

Cuprins

[1 Introducere si obiective 3](#_Toc60579015)

[2 Mini Proiect 3](#_Toc60579016)

[3 Descrierea functionalitatilor 4](#_Toc60579017)

[4 Limbaje si tehnologii 5](#_Toc60579018)

[4.1 Node.js 5](#_Toc60579019)

[4.2 JavaScript 5](#_Toc60579020)

[4.3 Socket.IO 5](#_Toc60579021)

[4.4 Bcrypt si Passport 6](#_Toc60579022)

[4.5 Express 6](#_Toc60579023)

[4.6 P5.js library 6](#_Toc60579024)

[5 Proiectare si implementare 7](#_Toc60579025)

[5.1 Diagrama cazurilor de utilizare 7](#_Toc60579026)

[5.2 Diagrama de stare 8](#_Toc60579027)

[5.3 Diagrama de secventa 9](#_Toc60579028)

[5.4 Design Patterns 10](#_Toc60579029)

[5.4.1 State Pattern 10](#_Toc60579030)

[5.4.2 Decorator Pattern 10](#_Toc60579031)

[6 User Manual 10](#_Toc60579032)

[Bibliografie 11](#_Toc60579033)

# Introducere si obiective

Agario este un joc de actiune multiplayer ce captiveaza atentia utilizatorilor de la prima incercare de a invinge un oponent. Obiectivul jocului consta in a aduna cat mai multe puncte pentru a-si mari dimensiunea propriei celule si a ajunge in ipostaza de a manca un alt player si a avansa, astfel, in tabela de scor.

Proiectul dat a fost un prilej de familiarizare cu noi tehnologii si limbaje de programare ce au stat la baza crearii paginilor web ale jocului. Un alt obiectiv atins a fost aprofundarea modului de functionare al arhitecturii client-server ce presupune efectuarea operatiilor de distribuire a sarcinilor intre furnizori (server) si cei care solicita aceste resurse (clienti). Comunicarea a fost realizata si utilizand libraria Socket.io pentru a transmite bidirectional informatii intre client si server.

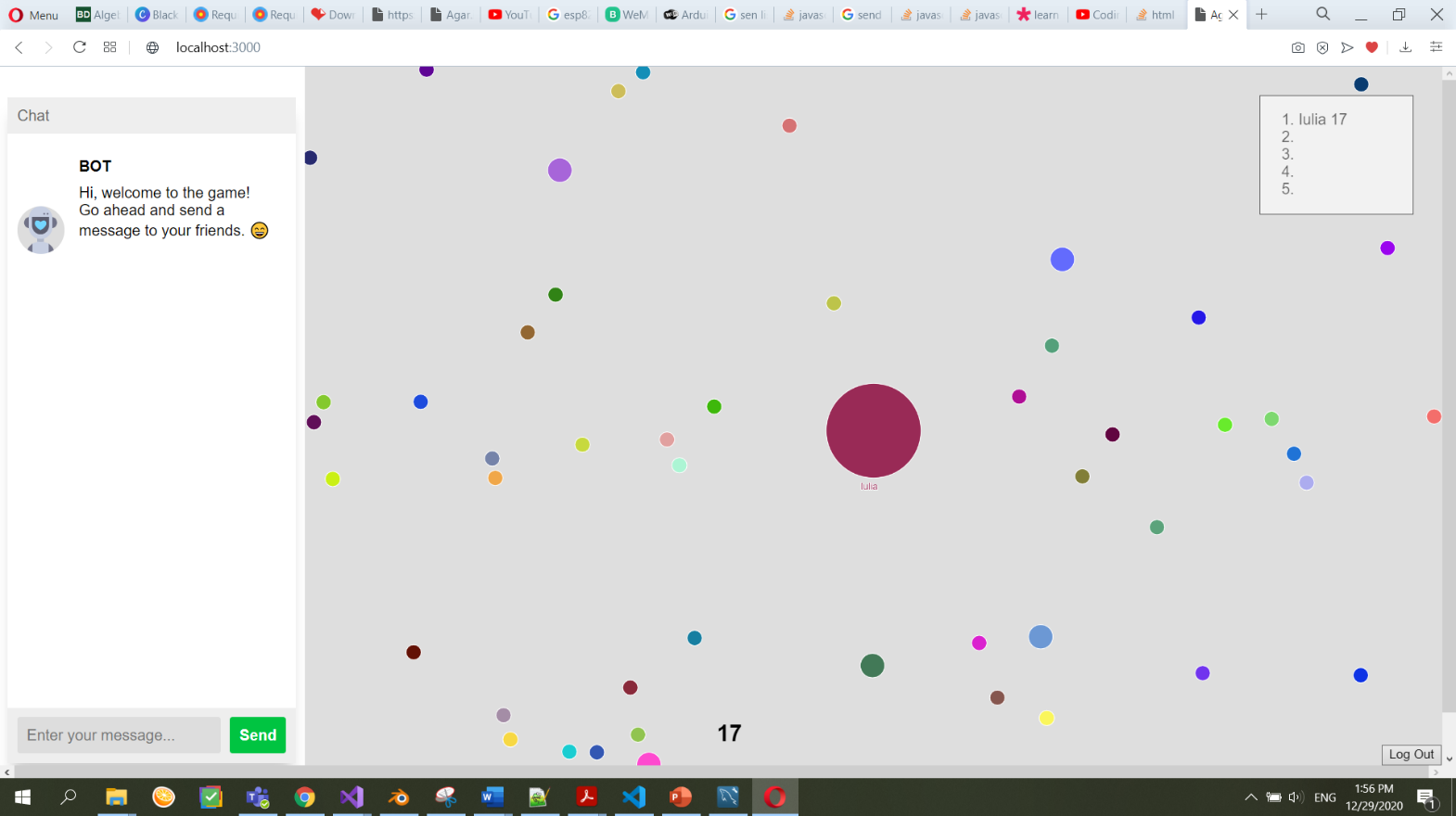
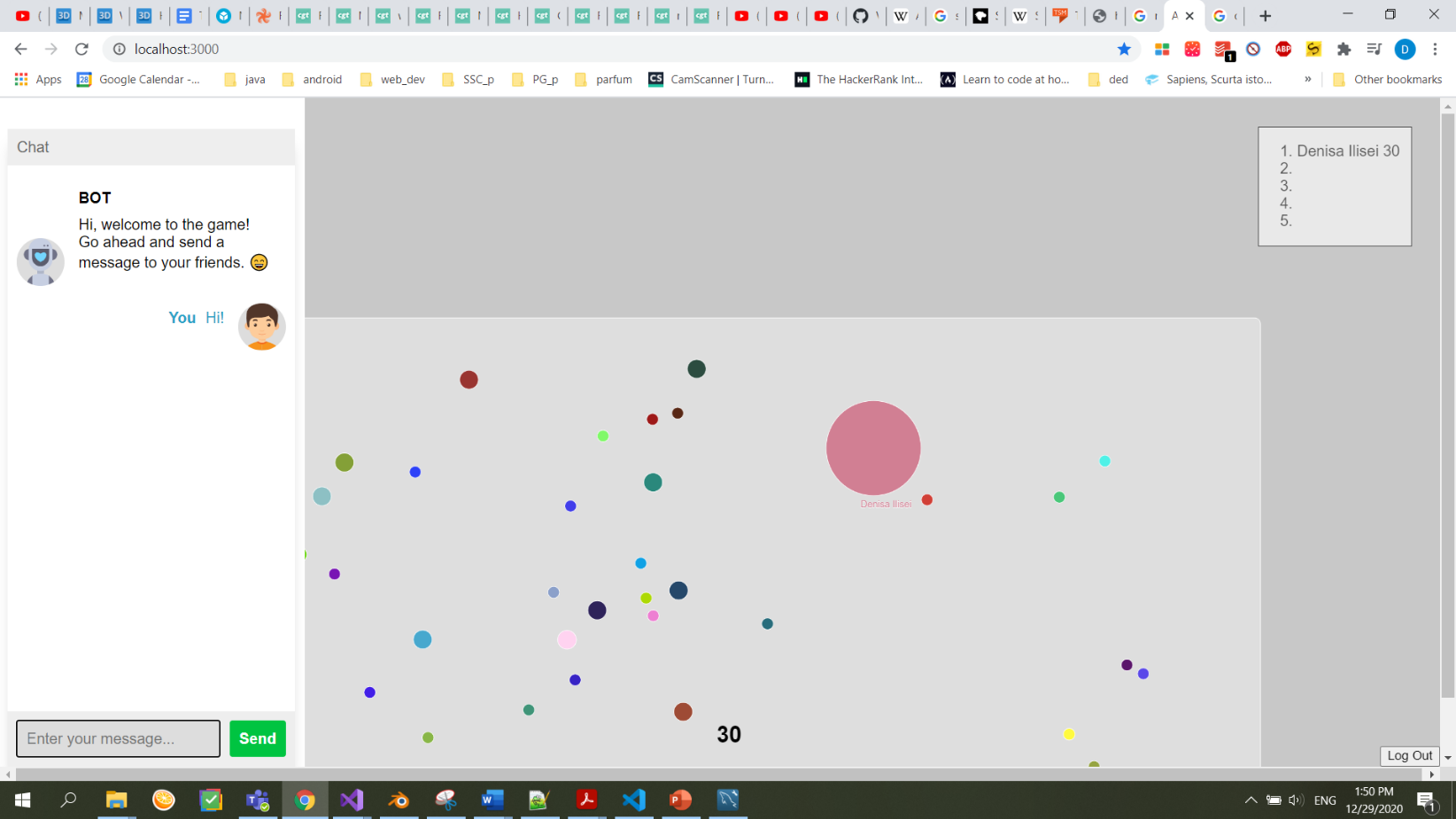
# Mini Proiect

Partea de mini proiect a constat in realizarea unei pagini de logare si inregistrare a utilizatorilor. Pentru aceasta, primul pas a fost crearea unei baze de date pentru sistem ce va cuprinde informatii despre user: username, email, parola. Pentru securizarea operatiilor, parola a fost salvata in baza de date criptata, acest lucru fiind posibil utilizand libraria bcrypt. Operatiile de creare, stergere, actualizare a bazei de date s-au realizat in serverul scris in limbajul node.js , un mediu de executie JavaScript ce permite rularea codului JavaScript in afara unui browser web.

In cazul mini proiectului, la pornirea server-ului, utilizatorul are posibilitatea de a accesa pagina de login. Daca acesta are un cont creat anterior, va introduce adresa de email si parola si va apasa butonul Login. In cazul in care datele se regasesc in baza de date, utilizatorul are permisiunea de a ajunge pe pagina web a jocului, iar in caz contrar se va afisa un mesaj de informare. Daca acesta nu a fost inregitrat anterior, poate accesa pagina de Register, unde va introduce datele necesare si se va memora un nou cont. Dupa inregistrare, user-ul este redirectionat la pagina de login si se va putea loga pentru a incepe jocul. Stilizarea paginilor de login si register a fost realizata utilizand libraria Bootstrap.

# Descrierea functionalitatilor

Utilizatorul are acces la joc dupa crearea unui cont de utilizator si logare, moment in care este redirectionat la pagina jocului. Jucatorul este reprezentat pe canvas printr-o celula ce are atasat in dreptul sau numele care a fost inregistrat. Aceasta celula este controlata cu ajutorul mouse-ului. Directia de deplasare a sa urmeaza cursorul mouse-ului. Jucatorul se va deplasa pe harta incercand sa creasca dimensiunea celulei prin acumularea mai multor bilute de mancare ce se afla imprastiate pe toata suprafata hartii. O biluta de mancare reprezinta un punct in plus la scorul afisat in tabela. In momentul in care mai multi utilizatori se conecteaza la joc, cel ce are o dimensiune mai mare (scor mai mare) poate manca un alt oponent, cu o dimensiune mai mica daca acestia se intalnesc. Odata cu marirea scorului, va scadea viteza unui player, astfel crescand sansele unuia mai mic de a se apara de cel mai mare. Jucatorii au ca scop si avansarea in tabela de scor. In momentul prinderii unui oponent, cel castigator va lua punctele acestuia, iar cel ce pierde va reveni la dimensiunea initiala.

Jocul dispune si de o sectiune de socializare, ce le permite jucatorilor sa trimita mesaje celorlalti prin chat-ul din partea stanga. In momentul in care un utilizator foloseste tastele, miscarea celulei sale se va opri. Pentru a reporni jocul, acesta va da click pe harta si miscarea va reincepe. Aceasta functionalitate de chat permite si crearea aliantelor si intalnirea cu prietenii in cadrul jocului.

# Limbaje si tehnologii

## Node.js

Node.js este un mediu de executie al codului JavaScript in afara unui browser web ce permite realizarea unui server utilizand JavaScript si organizarea in module. Motivul pentru care a fost aleasa aceasta metoda de a scrie serverul este ca Node.js pune la dispozitie programarea bazata pe evenimente, permitand dezvoltarea de servere web rapide. Date fiind schimbarile rapide care se petrec pe partea de client in timp real, a fost necesara gasirea unei modalitati de a pastra viteza de raspuns la cereri. In mod traditional, fiecare request ar fi dat nastere la un thread, dar in cazul Node.js exista un singur thread in care se utilizeaza apeluri non-blocking de intrare-iesire, permitand astfel mai multe conexiuni concurente rapide.

## JavaScript

JavaScript este un limbaj de programare utilizat atat pe partea de client cat si pe partea de sever pentru realizarea paginilor web interactive. O parte importanta a limbajului este reprezentata de evenimente. Acestea sunt structuri de cod ce urmaresc activitatea ce apare in browser si pot efectua o actiune corespunzatoare. Cateva exemple care folosesc acest concept folosite in proiect sunt: tratarea apasarii unei taste sau a butonului mouse-ului, miscarea mouse-ului pagina web, etc.

## Socket.IO

Socket.IO este o librarie JavaScript utilizata pentru crearea aplicatiilor web realtime, pentru evenimentele ce au loc instant. Tehnologia de socket este des pusa in practica in cazul aplicatiilor cu chat deoarece permite comunicarea bidirectionala intre server si client. Astfel, serverul poate transmite mesaje clientilor, iar mesajele transmise de clienti pot fi preluate de server si transmise tuturor celorlalti. Aceasta comunicare este realizata prin functii de transmisie a mesajelor (spre exemplu emit) si de preluare a lor (on). Fiecare client conectat poate fi identificat printr-un ID. In cazul in care un client se deconecteaza, aceasta actiune poate fi surprinsa de server si tratata intr-un anumit mod.

Prin intermediul functiilor puse la dispozitie de acest mediu, a fost posibila monitorizarea schimbarilor care apar in view-ul paginii fie in momentul atacarii unui oponent, fie in cazul transmiterii unor mesaje, fie in cazul conectarii unui nou player.

## Bcrypt si Passport

Bcrypt este instrumentul folosit pentru securizarea parolelor stocate in baza de date. Functia hash are ca argumente parola ce trebuie criptata si numarul de iteratii ale algoritmului de hashing. Salt-ul este un string random ce este adaugat la parola pentru securizare. De la aceeasi parola se genereaza un nou salt e fiecare data pentru a permite decodificarea sa.

Passport este un middleware utilizat pentru autentificarea utilizatorilor. Cu ajutorul functiilor authenticate si checkAuthenticated se monitorizeaza starea utilizatorilor, daca acestia sunt logati sau nu.

## Express

Express este un web framework pentru Node.js care furnizeaza mecanisme pentru tratarea cererilor date de diferite URL-uri si redirectionarea catre parti diferite ale aplicatiei.

Express-session module este utilizat pentru mentinerea user-ului activ pe diferite pagini iar express-flash faciliteaza transmiterea unor mesaje (spre exemplu in cazul introducerii unui nume existent in baza de date in cadrul inregistrarii).

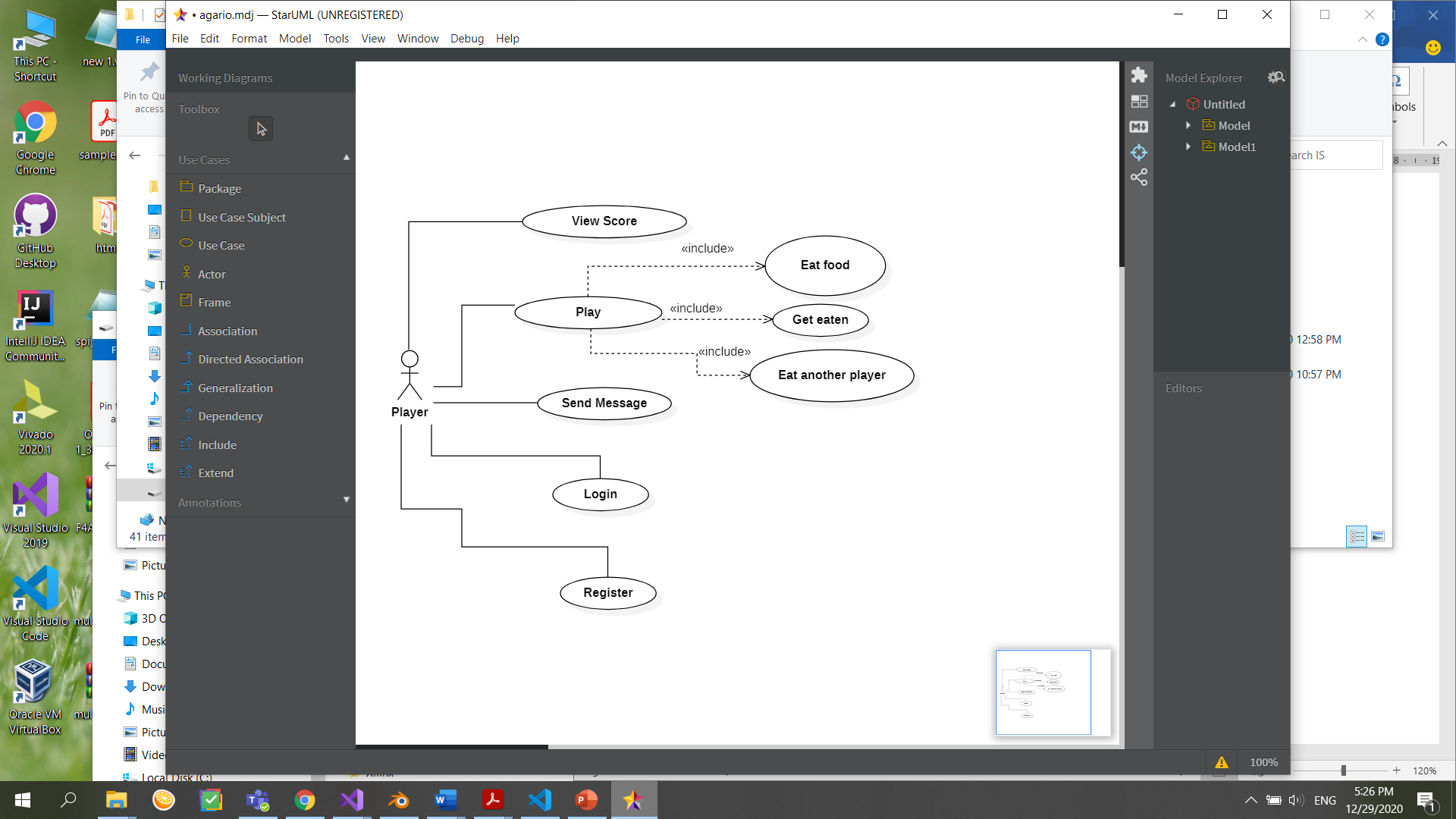
## P5.js library

Libraria P5 este instrumentul ce a facilitat desenarea formelor pe canvas, avand doua functii principale: draw ce se va executa continuu pana la specificarea in cod a opririi si functia setup ce se executa o singura data la inceputul programului. Aceasta librarie vine cu diferite functionalitati de desenare sau de urmarire a evenimentelor de pe canvas, facand posibila interactivitatea elementelor desenate.

# Proiectare si implementare

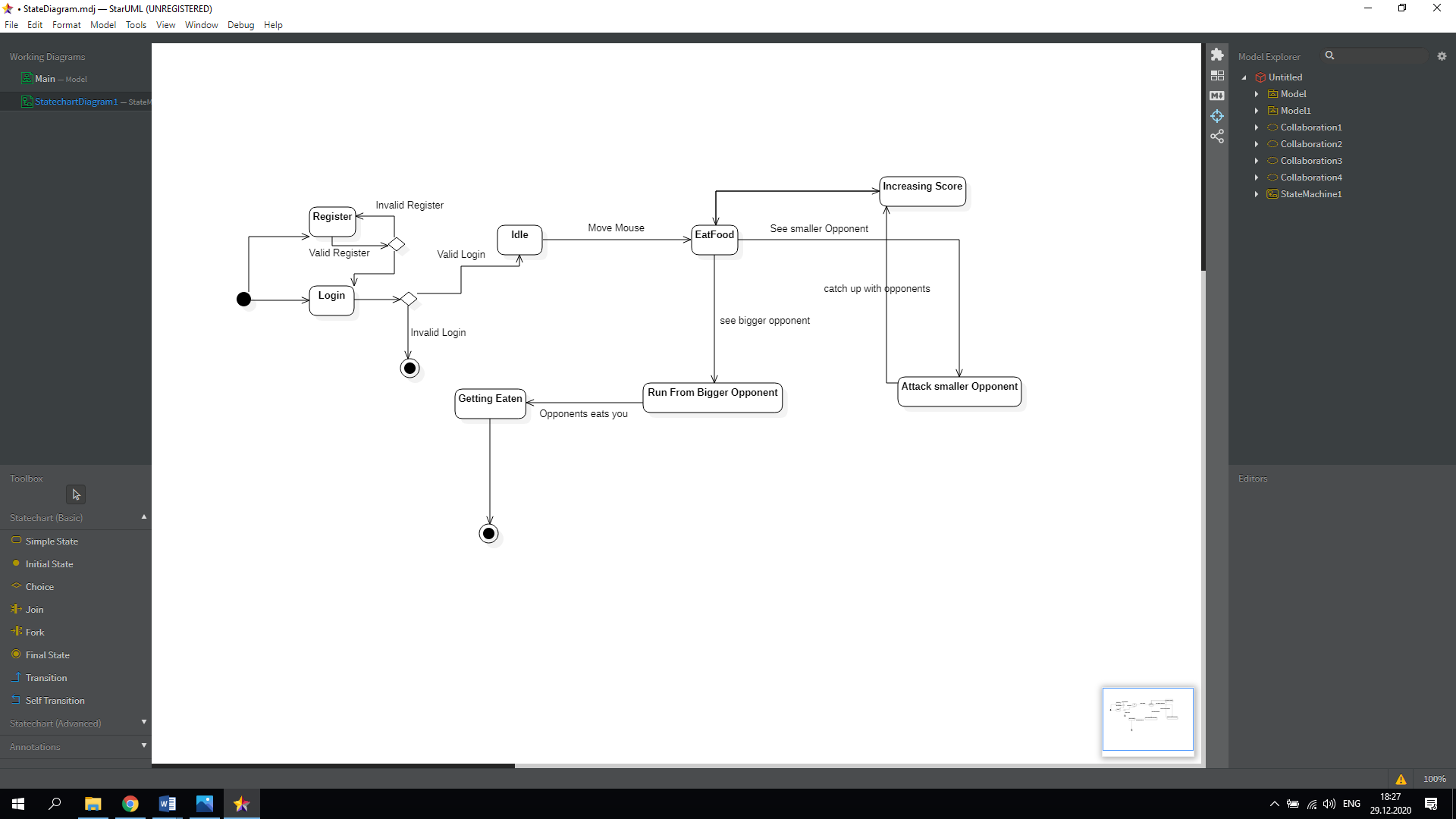
## Diagrama cazurilor de utilizare

Diagrama cazurilor de utilizare pune in evidenta actorii participanti ai sistemului si modurile in care sistemul poate fi utilizat. Utilizatorul care interactioneaza cu sistemul are denumirea de player. Acesta are la dispozitie mai multe functionalitati. La prima interactiune, el se va inregistra cu numele si email si va putea accesa, dupa logare, pagina jocului. In momentul aflarii pe pagina jocului, este evidentiata functionalitatea de Play. Aceasta implica mancarea celulelor mai mici, mancarea unui alt oponent sau pierderea jocului data de evenimentul prinderii propriei celule. Player-ul mai are posibilitatea de a vedea scorul in tabela de scor sau in partea de jos a paginii sau de a trimite un mesaj pe chat.



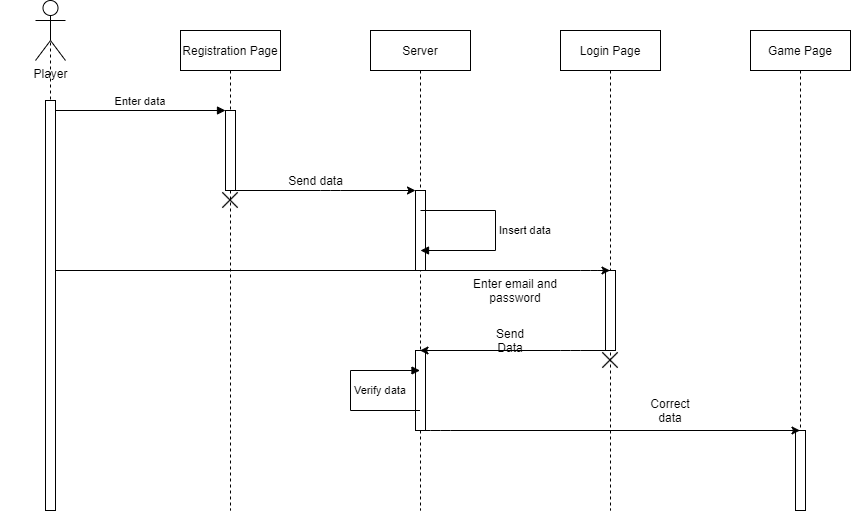
## Diagrama de stare

Diagrama de stare face posibila reprezentarea evenimentelor ce pot avea loc din sistem, a starilor caracterizate de un anumite comportament si tranzitiile care au loc. Din punctul de start al sistemului, user-ul are posibilitatea de a se loga, daca dispune de un cont creat anterior sau de a se inregistra. Daca datele introduse sunt valide (spre exemplu email-ul are forma cunoscuta a unui email) se poate trece la terminarea inregistrarii si logare, in caz contrar sunt solicitate alte date. Din starea de Login, daca datele sunt valide, utilizatorul va fi introdus in joc si va incepe sa manance celule mici pentru a aduna puncte daca va misca cursorul mouse-ului. Daca utilizatorul observa pe harta un player cu un scor mai mic, va urmari sa il prinda pentru a prelua punctele. In cazul in care exista coliziune intre cei doi, cel mai mare il va scoate din joc pe cel mai mic. Daca player-ul este mai mic decat alt oponent si vede intentia acestuia de a-l prinde, va incerca sa se apere indepartandu-se. In cazul in care exista coliziune, jocul se termina pentru oponentul mai mic.



## Diagrama de secventa

Diagrama de secventa evidentiaza interactiunile dintre entitatile sistemului intr-un anumit scenariu. Diagrama prezentata mai jos expune activitatea de inregistrare si logare a utilizatorului. In cazul inregistrarii, utilizatorul va introduce in campul corespunzator numele, email-ul si parola pe care le doreste. Dupa apasarea butonului de register, datele sunt transmise serverului, care le stocheaza in baza de date si se face redirectionarea la pagina de login. De aici, utilizatorul va trebui sa introduca email-ul si parola, iar dupa actionarea butonului Login, datele vor fi transmise la server pentru a fi verificate. Daca acestea sunt corecte, utilizatorul are posibilitatatea de a vizualiza pagina jocului si de a incepe colectarea celulelor de mancare.



## Design Patterns

### State Pattern

State pattern este un model behavioral ce permite unui obiect sa-si modifice comportamentul atunci cand starea sa interna se schimba, fiind un model asemanator automatelor cu stare finita. Cu ajutorul acestuia se poate implementa schimbarea comportamentului unui obiect in functie de starea interna. Starile sunt independente, fiind posibila adaugarea unor noi stari fara a le afecta pe cele anterioare. Programul poate face trecerea de la o stare la alta instantaneu, iar decizia de a depasi starea curenta este luata tinand seama de anumite conditii.

In jocul Agario, folosirea state pattern a facilitat atribuirea nivelului de experinta utilizatorilor. Fiecare utilizator va putea avea, pe rand, titulatura de Junior, Medium sau Expert. Aceste stagii sunt semnalizate prin medalii in dreptul scorului, in tabela de scor. Utilizatorul Junior va avea medalie de bronz, Medium - de argint, iar Expert – de aur. Fiecare dintre cele 3 stari este descrisa intr-o clasa separata. Decizia tranzitiei de la o stare la alta se ia, de asemenea in fiecare dintre clasele ce reprezinta titulatura. La toate acestea este adaugat contextul, clasa Experience ce cuprinde obiectul 'starea curenta' si permite trecerea la o stare urmatoare.

### Decorator Pattern

Decorator pattern este un model structural ce permite adaugarea unor noi comportamente unui anumit obiect, in mod dinamic, fara a afecta comportamentul obiectelor initiale. Decorator poate fi utilizat pentru extinderea functionalitatilor prin crearea unei noi clase ce mentine caracteristicile clasei anterioare dar vine cu noi atribute sau functii, facand posibila reutilizarea codului. Caracteristicile adaugate nu sunt considerate esentiale functionalitatii de baza a clasei. Utilizarea acestora elimina necesitatea de a modifica codul existent.

Utilizarea Decorator a facut posibila crearea unor bile ce au proprietati de "otravire". Implementarea acestei functionalitati s-a realizat prin crearea unei noi clase, ce are ca atribute un obiect 'blob' ce cuprinde caracteristicile obisnuite si atributul poisonColor ce reprezinta o culoare considerata otravitoare – negru sau albastru. In functie de aceasta culoare, daca un player va manca bila otravitoare va pierde cateva puncte din scor.

# User Manual

Pentru configurarea proiectului si pornirea server-ului pe un nou calculator sunt necesare cateva instalari. Baza de date necesita folosirea server-ului MySql. De asemenea, pentru rularea serverului va fi necesara descarcarea si instalarea Node.js (incluzand npm). Dupa acesti pasi, in linia de comanda, in directorul proiectului se introduc comenzile:

npm install express

npm install mysql2

npm install socket.io

npm install passport

npm install bcrypt

Dupa acesti pasi, serverul poate fi pornit prin comanda **node server.js**, iar in browser se introduce **<http://localhost:3000/>**

# Bibliografie

|  |  |
| --- | --- |
| [1] | "NodeJs Guides," [Online]. Available: https://nodejs.org/en/docs/guides/. |
| [2] | "MDN web docs," mozilla, [Online]. Available: https://developer.mozilla.org/en-US/. |
| [3] | "Socket.io," [Online]. Available: https://socket.io/. |
| [4] | "Express," Nodejs, [Online]. Available: https://expressjs.com/. |