

Gamification

Evaluare Tehnica

Internship 2023 @Accesa & RaRo

Vilcelean Dalia

dalia.vilcelean@gmail.com

Student

Cuprins:

1. Introducere
2. Scopul si Obiectivele Aplicatiei
3. Prezentare Aspecte Generale
4. Prezentare Aspecte Tehnice
5. Prezentare Detaliata a Functionalitatilor
6. Manual de Instalare si Utilizare
7. Idei de Interfata Grafica
8. Contributii ulterioare
9. Concluzii

1. Introducere

Așa cum faimosul Bob Dylan cânta în anii ‘60, “The Times They Are a-Changin’”, vremurile se schimbă și pentru noi, programatorii, în 2023. Tool-urile pe care le folosim devin tot mai bune, iar skill-urile de care noi avem nevoie pentru a avea succes, tot mai diverse.

Din acest motiv, am creat o aplicație Web pentru compania Blue care are nevoie de o soluție software prin care să crească engagement-ul angajaților, în așa fel încât aceștia să se implice în mai multe activități, în urma cărora să aibă un sentiment mai puternic de satisfacție. Studiile arată că un concept de gamification s-ar plia foarte bine pentru a rezolva problema într-un mod interactiv, oferind utilizatorilor o experiență faină.

Această documentație oferă o prezentare generală a proiectului, descrie arhitectura și modul de funcționare a aplicației, precum și cum să o instalați și să o utilizați.

Gaminator este o aplicație web de gamificare care are ca scop motivarea utilizatorilor să își îmbunătățească abilitățile și performanțele prin îndeplinirea de sarcini (quest-uri) și obținerea de recompense (badges și tokens). Aceasta poate fi utilizată atât în mediul corporatist, pentru a motiva angajații și a îmbunătăți performanțele companiei, cât și în scop personal, pentru a îmbunătăți abilitățile și competențele individuale.

2. Scopul si Obiectivele Aplicatiei

Scopul aplicației este de a încuraja și motiva utilizatorii să își dezvolte abilitățile și să își atingă obiectivele personale prin participarea la quest-uri și câștigarea de recompense. Prin intermediul aplicației, utilizatorii pot accesa o varietate de quest-uri legate de job-ul de corporatist, care vizează dezvoltarea abilităților specifice acestui domeniu. Utilizatorii pot utiliza token-urile câștigate pentru a achiziționa beneficii precum cărți, cursuri și acces la conferințe de specialitate, ceea ce le poate ajuta să-și dezvolte și mai mult abilitățile profesionale.

Obiectivele aplicației includ creșterea performanței și productivității utilizatorilor în muncă, îmbunătățirea abilităților profesionale și oferirea unei experiențe de utilizare intuitivă și atractive. Acestea pot fi:

1. Motivarea utilizatorilor să își dezvolte abilitățile prin participarea la quest-uri și câștigarea de recompense.
2. Încurajarea utilizatorilor să își fixeze obiective personale și să își monitorizeze progresul în atingerea acestora.
3. Îmbunătățirea abilităților specifice job-ului de corporatist și creșterea performanței acestora în cadrul companiei.
4. Creșterea gradului de implicare și loialitate a angajaților față de companie prin intermediul oferirii de recompense.
5. Ofertarea unei platforme interactive și atractive pentru dezvoltarea abilităților și îmbunătățirea performanței profesionale.

3. Prezentare Aspecte Generale

Prin intermediul aplicației web, utilizatorii pot accesa o serie de misiuni și provocări, pe care le pot alege în funcție de interesele lor și de obiectivele pe care doresc să le atingă. Fiecare misiune și provocare are o descriere detaliată, împreună cu instrucțiuni clare cu privire la modul în care trebuie îndeplinită.

Utilizatorii pot încărca dovezi de îndeplinire a misiunilor și provocărilor, iar acestea pot fi verificate de administratorii aplicației. În funcție de performanțele obținute de utilizatori, aceștia pot primi recompense sub forma de token-uri sau de badge-uri, care le pot fi utile în dezvoltarea și creșterea profesională.

În plus, aplicația oferă și funcții de monitorizare a performanțelor și de urmărire a progresului angajaților. Acestea pot fi utilizate de către manageri pentru a evalua progresul angajaților și pentru a identifica posibile probleme sau deficiențe în performanța acestora.

Aplicația este structurată în module, fiecare modul fiind responsabil pentru anumite funcționalități. Aceste module includ gestiunea utilizatorilor, gestionarea quest-urilor și recompenselor, gestionarea badge-urilor și a token-urilor, precum și gestionarea rapoartelor și statisticii.

În ceea ce privește arhitectura bazei de date, aceasta a fost proiectată pentru a putea gestiona toate datele legate de utilizatori, quest-uri, recompense, badge-uri și

token-uri. Aceasta a fost construită folosind Microsoft SQL Server și include tabele, relații și chei străine pentru a asigura integritatea referențială și consistența datelor.

4. Prezentare Aspecte Tehnice

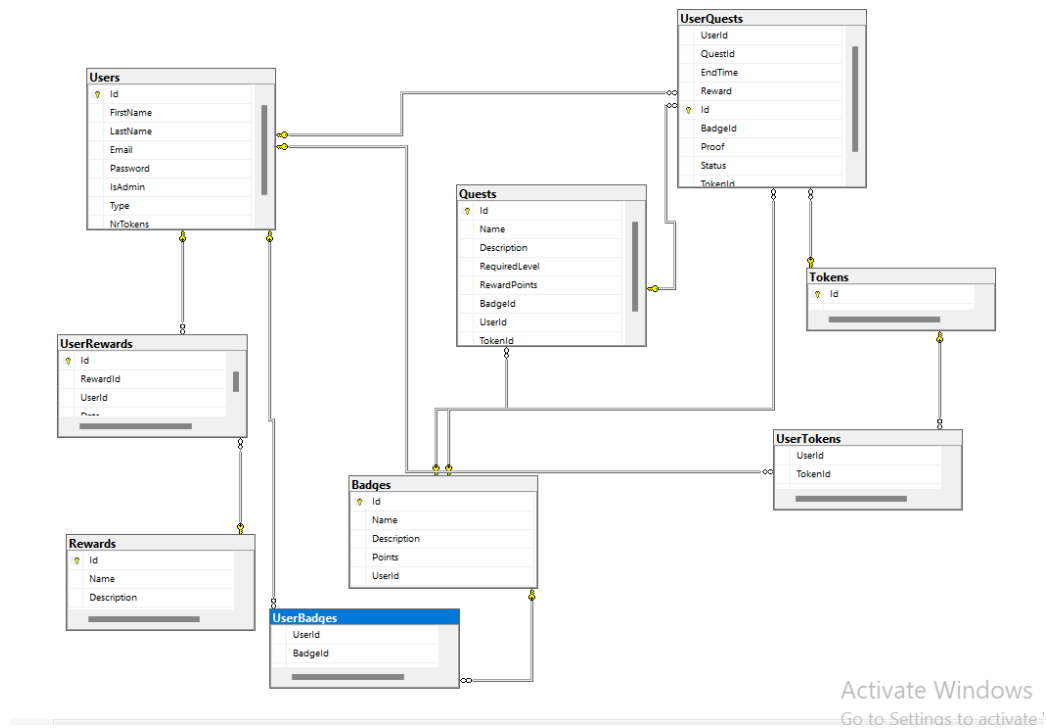
Aspectele tehnice ale aplicației includ utilizarea platformei .NET 6.0, care este o platformă de dezvoltare software open-source dezvoltată de Microsoft. Platforma .NET 6.0 oferă o gamă largă de instrumente și tehnologii care permit dezvoltatorilor să creeze aplicații de calitate într-un mod rapid și eficient.

În ceea ce privește arhitectura aplicației, aceasta a fost concepută pe baza arhitecturii Model-View-Controller (MVC) care separă logica de prezentare de logica de afaceri și de interacțiunea cu baza de date. Utilizarea acestei arhitecturi permite o dezvoltare modulară și scalabilă a aplicației, permițând dezvoltatorilor să adauge noi funcționalități sau să îmbunătățească funcționalitățile existente fără a afecta alte părți ale aplicației.

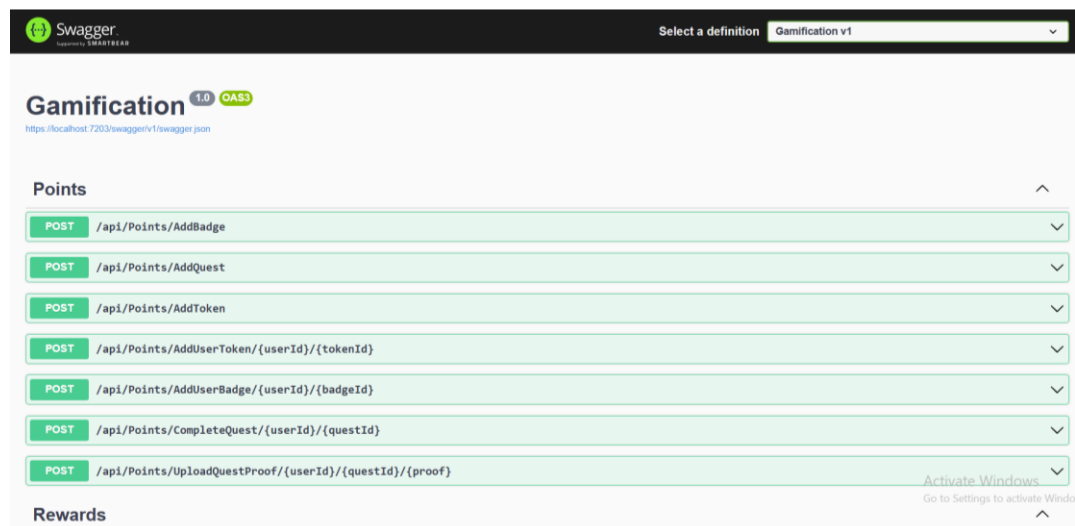
Datele vor fi luate din baza de date printr-o clasă GameDbContext, unde voi crea toate entitățile necesare și voi face legături între tabele. Pentru implementarea acestora am folosit framework-ul Entity Framework, care a permis crearea și actualizarea bazei de date folosind migrări. De asemenea, am folosit abordarea Code-First pentru a defini modelele bazei de date și a le mapea cu entitățile din codul sursă. Folosind aceste tehnologii, am putut asigura un management eficient al bazei de date și o adaptare ușoară la schimbările aduse de cerințele proiectului.

Pentru baza de date a aplicației, am utilizat SQL Server, care este un sistem de gestionare a bazelor de date relaționale dezvoltat de Microsoft. Acesta a fost ales datorită fiabilității și scalabilității sale, dar și datorită faptului că există multe instrumente și resurse disponibile pentru dezvoltatori pentru a-l utiliza.

Diagrama bazei de date:



Pentru gestionarea dependențelor, am utilizat pachetul NuGet, care este un manager de pachete pentru platforma .NET. Acesta permite dezvoltatorilor să instaleze și să gestioneze pachetele și bibliotecile de cod utilizate în aplicație.



Pentru gestionarea API-urilor am folosit interfata Swagger, permițând dezvoltatorilor terți să folosească cu ușurință funcționalitățile aplicației prin intermediul unei documentații clare și ușor de înțeles.

Aplicația noastră este structurată într-un mod modular și utilizăm mai multe componente pentru a asigura o funcționare corectă. Aceste componente includ:

1. DbContext - este clasa principală a framework-ului de bază pentru interacțiunea cu baza de date. Folosim acesta pentru a crea și executa migrări, pentru a face interogări și pentru a obține date din baza de date.
2. Entities - sunt modelele de date care sunt mapate la tabelele din baza de date. Acestea sunt utilizate pentru a accesa și modifica datele din baza de date.
3. DTOs - sunt obiecte transferate între diferite straturi ale aplicației, cum ar fi între API Controller și Service. Acestea au un format simplificat pentru a reduce traficul și a îmbunătăți performanța.
4. Repository - este un strat intermediar între DbContext și Service și este responsabil pentru accesul și manipularea datelor din baza de date.
5. Service - este stratul de business logic care gestionează operațiile de creare, citire, actualizare și ștergere a datelor. Acestea se bazează pe repository pentru a obține și modifica datele din baza de date.
6. API Controller - este stratul de interfață utilizator care primește cereri HTTP și returnează răspunsuri HTTP. Acestea utilizează serviciile pentru a efectua operațiile de bază și returnează DTO-uri către client.

5. Prezentare Detaliata a Functionalitatilor

1. **Autentificare și înregistrare utilizator:** utilizatorii se pot autentifica în aplicație utilizând adresa lor de e-mail și parola. Exista, deocamdata, 2 tipuri de utilizatori: user si admin. Userul va putea intra in contul propriu, sa-si vada datele personale, sa completeze Quest-uri si sa incarce dovada aferenta rezolvarii Quest-ului. Admin-ul va avea un rol foarte important, acesta poate realiza operatii CRUD pe useri, Quest-uri, Token-uri, Badges-uri si Reward-uri. Userii vor fi adaugati de catre Admin, acestia neputand sa-si creeze singuri un cont, fiind o aplicatie a corporatiei. Acesta va mai putea vedea si un clasament a tuturor Userilor in functie de Tokens si Badges, ordonati descrescator si sa accepte sau sa refuze o completare a unui Quest trimis de catre User.

2. **Crearea de Quest-uri:** utilizatorii pot crea propriile lor Quest-uri care să le ajute să-și atingă obiectivele personale. Acestea pot fi misiuni zilnice, săptămânale sau lunare și pot fi ajustate în funcție de preferințele utilizatorului.
3. **Urmărirea progresului:** utilizatorii pot vizualiza progresul lor pentru fiecare Quest terminat. Acest lucru îi ajută să vadă cum se descurcă și să se concentreze asupra obiectivelor lor personale.
4. **Acordarea de recompense:** utilizatorii pot obține recompense sub formă de badge-uri și puncte pentru atingerea obiectivelor lor. Aceste recompense sunt afișate pe profilul lor și le permit să urmărească realizările lor și să se simtă motivați să continue să-și atingă obiectivele. Pe langa asta, la acumularea unui anumit număr de puncte în aplicație, acesta își va putea cumpara un Reward.
5. **Tokens-urile și Badges-urile** vor fi legate de fiecare Quest la adăugarea acestuia în baza de date, iar la completarea acelui Quest, User-ul va primi recompensele promise.

6. Manual de Instalare și Utilizare

Pentru a instala și utiliza aplicația, urmați acești pași:

1. Descărcați și instalați .NET 6 SDK de pe site-ul oficial al Microsoft:
<https://dotnet.microsoft.com/download/dotnet/6.0>
2. Descărcați sau clonați codul sursă al aplicației din repository-ul GitHub.
3. Deschideți proiectul într-un editor de cod (cum ar fi Visual Studio Code) sau în Visual Studio.
4. Accesați fișierul appsettings.json și completați informațiile necesare pentru a vă conecta la baza de date. De asemenea, asigurați-vă că baza de date este creată în prealabil.

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=LAPTOP-DALIA;Database=Gaminator1;Trusted_Connection=SSPI;Encrypt=False;TrustServerCertificate=true;"  
},
```

5. Deschideți Consola de Gestiune a Pachetelor (Package Manager Console) și rulați comanda `dotnet ef database update` pentru a aplica migrările la baza de date.
6. În aplicație, puteți utiliza endpoint-urile API-ului pentru a accesa diversele funcționalități.

Pentru a utiliza aplicația, urmați pașii specifici pentru fiecare funcționalitate:

Pentru a accesa funcționalitatea de autentificare, accesați link-ul <https://localhost:7203/api/User/LogIn> și trimiteți un request POST cu un obiect JSON care conține numele de utilizator și parola. Aplicația va returna un DTO pe care îl puteți utiliza pentru a accesa alte funcționalități care necesită autentificare.

Pentru a accesa funcționalitatea de creare de utilizator, accesați link-ul <https://localhost:7203/api/User/SignUp> și trimiteți un request POST cu un obiect JSON care conține informațiile despre noul utilizator.

Pentru a accesa funcționalitatea de creare de quest-uri, accesați link-ul <https://localhost:7203/api/Points/AddQuest> și trimiteți un request POST cu un obiect JSON care conține informațiile despre noul quest.

Pentru a accesa funcționalitatea de completare a unui quest, accesați link-ul <https://localhost:7203/api/Points/CompleteQuest/{userId}/{questId}> și trimiteți un request POST cu un obiect JSON care conține ID-ul utilizatorului și ID-ul quest-ului.

Pentru a accesa funcționalitatea de verificare a unui quest, accesați link-ul <https://localhost:7203/api/Points/UploadQuestProof/{userId}/{questId}/{proof}> și trimiteți un request POST cu un obiect JSON care conține ID-ul utilizatorului, ID-ul quest-ului și un URL către proba quest-ului. Este necesar să fiți autentificat ca administrator pentru a putea efectua această acțiune.

Pentru a accesa funcționalitatea de cumparare a unui Reward, accesați link-ul <https://localhost:7203/api/Rewards/BuyReward/{userId}/{itemId}>. Aceasta funcție permite unui utilizator să cumpere un reward folosind punctele lui de recompensă. Request-ul trebuie să conțină ID-ul utilizatorului care dorește să cumpere reward-ul și ID-ul reward-ului pe care îl dorește. Dacă utilizatorul are suficiente puncte de recompensă și reward-ul este în stoc, atunci tranzacția este procesată și punctele de recompensă sunt scăzute din contul utilizatorului.

Pentru a adăuga un Reward în baza de date, accesați link-ul <https://localhost:7203/api/Rewards/AddReward>. Aceasta funcție permite unui utilizator cu rol de administrator să adăuge un nou reward în baza de date. Request-ul trebuie să conțină un obiect JSON care să descrie detaliile noului reward.

Pentru a accesa detaliile unui anumit utilizator, accesați link-ul <https://localhost:7203/api/User/GetUserDetails/{userId}>. Aceasta funcție returnează detaliile utilizatorului cu ID-ul specificat în request. Detaliile includ numele utilizatorului, adresa de email și punctele de recompensă deținute.

Pentru a accesa o lista cu toti userii in ordinea activitatii lor, accesati link-ul <https://localhost:7203/api/User/GetAllUsers>: Aceasta functie returneaza o lista cu toti utilizatorii din baza de date impreuna cu detaliile fiecaruia.

Pentru a utiliza aceste functionalitati, utilizatorii trebuie sa fie autentificati si sa detina rolurile corespunzatoare. De asemenea, trebuie sa se asigure ca baza de date a aplicatiei este configurata si functioneaza corespunzator.

7. Idei de Interfata Grafica

Avand la dispozitie un timp foarte limitat de dezvoltare a acestei aplicatii, nu am reusit sa implementez si o interfata grafica, dar voi propune in documentatie idei pentru aceasta, tehnologii si cum voi folosi API-urile pentru a o face functionala pana la capat.

Interfața grafică (UI) a aplicației va fi dezvoltată folosind tehnologii web, mai exact React.js pentru componentele UI, JavaScript pentru logica interactivității și CSS pentru stilizarea și aspectul vizual. De asemenea, se va folosi un design responsiv pentru a asigura o experiență consistentă utilizatorilor de pe dispozitivele mobile și desktop.

Pentru a gestiona starea componentelor și a interacțiunilor utilizatorului cu aplicația, vom utiliza librăria Redux, care permite o gestionare eficientă și predictibilă a stării aplicației. De asemenea, pentru a crea interfețe de utilizator responsive și performante vom folosi librăria React Bootstrap, care ne oferă un set de componente UI predefinite, stilizate și compatibile cu design-ul bootstrap.

Pentru a adăuga o notă vizuală plăcută și profesională la interfața aplicației, vom folosi o paletă de culori corespunzătoare temei aplicației și vom asigura o utilizare coerentă a fonturilor și a stilurilor de design. De asemenea, vom asigura că interfața aplicației este ușor de utilizat și intuitivă prin poziționarea strategică a elementelor UI și utilizarea de animații și alte efecte vizuale.

Idei pentru Interfata Grafica:

1. O pagină de autentificare simplă, care va redirectiona user-ul pe pagina aferenta rolului sau.
2. Pagina principală a aplicației care ar putea afișa informații despre utilizator și diverse opțiuni pentru navigarea în cadrul aplicației.

3. O pagină de profil pentru fiecare utilizator, care ar putea afișa numele, poza de profil, informații despre nivelul de experiență și punctele acumulate.
4. Pagina de Rewards, care afișează lista de recompense disponibile pentru utilizatori și informații despre modul de achiziționare.
5. O pagină de istoric a recompenselor achiziționate, unde utilizatorul poate vedea toate recompensele pe care le-a obținut și detaliile lor.
6. O pagină de clasament, care afișează topul utilizatorilor în funcție de punctele acumulate. Aceasta pagina va fi doar pentru Admin.
7. O pagină de administrare pentru administratorii sistemului, care ar putea oferi opțiuni pentru gestionarea utilizatorilor, recompenselor și a altor aspecte ale aplicației.

8. Contributii Ulterioare

O posibila contributie ulterioara pentru aplicatie ar fi implementarea unui WebSocket pentru a permite unui administrator sa primeasca notificari in timp real atunci cand un utilizator completeaza un quest. Acest lucru ar putea fi realizat prin intermediul unei conexiuni WebSocket intre server si client, care ar putea fi configurata pentru a notifica automat administratorul atunci cand un quest este marcat ca fiind completat de catre un utilizator.

Odata ce notificarea este primita, aceasta ar putea fi afisata pe pagina admin-ului a si ar putea oferi optiunea de a accepta sau respinge completarea quest-ului. Aceasta functionalitate ar putea fi deosebit de utila pentru a imbunatati procesul de verificare si validare a completarii quest-urilor de catre utilizatori, in special in cazul in care utilizatorii pot castiga recompense pentru completarea lor.

Alte idei de dezvoltari ulterioare:

1. Crearea unui sistem de ranguri pentru utilizatori bazat pe activitatile lor in cadrul aplicatiei.
2. Dezvoltarea unei functionalitati de chat sau mesagerie pentru a permite utilizatorilor sa comunice intre ei si sa formeze echipe sau grupuri de interes comun.
3. Adaugarea unui sistem de notificari push pentru a alerta utilizatorii cu privire la evenimente sau activitati relevante.

9. Concluzii

În concluzie, această aplicație este utilă pentru utilizatorii care doresc să își gestioneze activitățile și să primească recompense pentru îndeplinirea lor. Ea oferă o interfață intuitivă și ușor de utilizat, cu o gamă variată de funcționalități, cum ar fi gestionarea quest-urilor și a recompenselor, precum și crearea de noi quest-uri și recompense.

Tehnologiile folosite, cum ar fi .NET 6 și SQL Server, asigură performanțe ridicate și o securitate sporită, iar acestea pot fi utilizate și pentru a dezvolta alte aplicații web cu funcționalități similare.

În plus, posibilitățile de extindere și dezvoltare ulterioară a aplicației oferă oportunități noi și interesante pentru viitoarele îmbunătățiri.