



Μηχανική Λογισμικού I
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα HMMY
Α.Π.Θ

8^ο Εξάμηνο
Άνοιξη 2023



Backlogged

For All Your Gaming Needs

Απαιτήσεις Χρηστών και
Προδιαγραφές Συστήματος

07/05/2023



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Λίστα Σχημάτων	3
1 Απαιτήσεις Συστήματος	4
1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)	4
1.2 Χρήστες και εξωτερικά συστήματα	7
1.3 Σημαντικές Μη Λειτουργικές Απαιτήσεις	7
1.10 Λεξικογραφικοί Προσδιορισμοί	8
2 Σενάρια Χρήσης	10
2.1 Διάγραμμα σεναρίων χρήσης	10
3 Επιδεικτικά γραφικά παράθυρα διεπαφής	22
4 Στατική Μοντελοποίηση	26
4.1 <Game Package>	27
4.2 <Search Package>	44
4.3 <Review Package>	48
4.4 <Authentication Package>	52
4.5 <Home Package>	55
4.6 <User Package>	57
Παράρτημα I – Γλωσσάριο	63
Παράρτημα II – Ανοιχτά Θέματα	64



Λίστα Σχημάτων

- Figure 1: Splash Screen
- Figure 2: Login Page
- Figure 3: Register Page: Terms of Service Accepted
- Figure 4: Register Page: Passwords do not match
- Figure 5: Register Page: Email already taken
- Figure 6: Register Page: Terms of Service not Accepted
- Figure 7: Game Page: Top
- Figure 8: Game Page: Description
- Figure 9: Game Page: Scrolled to Bottom
- Figure 10: Game Page: Platform List
- Figure 11: Game Page: Developers and Publishers List
- Figure 12: Game Page: Add positive Review
- Figure 13: Game Page: Review expanded
- Figure 14: Game Page: Review scrolled
- Figure 15: Profile Page: Top
- Figure 16: Profile Page: Scrolled
- Figure 17: Edit Profile Page
- Figure 18: Home Page: Top
- Figure 19: Home Page: Scrolled
- Figure 20: Search Page
- Figure 21: Search Page: Search Results
- Figure 22: Filter Page



1 Απαιτήσεις Συστήματος

1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)

<ΛΑ- 1>

Ο χρήστης πρέπει να μπορεί να εγγραφεί στην εφαρμογή.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να δημιουργήσει έναν λογαριασμό στην εφαρμογή, είτε μέσω email, username και password, είτε αυτομάτως μέσω του ήδη υπάρχοντα λογαριασμού του στο Discord (το αρχικό username του θα είναι το ίδιο με αυτό του λογαριασμού στο Discord).

User Priority: **4/5** Είναι απαραίτητο για τον χρήστη να έχει προσωπικό λογαριασμό, ώστε να μπορεί να διαχειριστεί την προσωπική του λίστα backlog.

Technical Priority: **5/5** Είναι αδύνατο να υλοποιηθούν κάποιες λειτουργικότητες του συστήματος χωρίς τους χρήστες.

<ΛΑ- 2>

Ο χρήστης πρέπει να μπορεί να συνδεθεί στην εφαρμογή.

Περιγραφή: Ο χρήστης πρέπει να έχει την δυνατότητα να συνδεθεί στην εφαρμογή με τα προσωπικά του στοιχεία (email, password) ή μέσω του Auth API του Discord.

User Priority: **4/5** Χωρίς την δυνατότητα να συνδεθεί στην εφαρμογή, ο χρήστης δεν μπορεί να έχει πρόσβαση σε αυτή.

Technical Priority: **5/5** Το σύστημα δεν μπορεί να λειτουργήσει αν δεν το χειρίζεται κάποιος χρήστης.

<ΛΑ- 3>

Ο χρήστης πρέπει να μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία

Περιγραφή: Ο χρήστης πρέπει να μπορεί να επεξεργαστεί το email, username, password και προαιρετικά την φωτογραφία προφίλ του και το ονοματεπώνυμό του

User Priority: **3/5** Είναι σημαντικό για το χρήστη να έχει την δυνατότητα να **διαχειρίζεται** τον λογαριασμό του, σε περίπτωση που επιθυμεί να αλλάξει είτε την εμφάνιση του προφίλ του, είτε το email και το password του

Technical Priority: **2/5** Το σύστημα μπορεί να λειτουργήσει χωρίς την δυνατότητα επεξεργασίας των στοιχείων των χρηστών.

<ΛΑ- 3>

Ο χρήστης πρέπει να μπορεί να **προσθέτει παιχνίδια** στην προσωπική του λίστα backlog.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να **προσθέσει** τα παιχνίδια που τον ενδιαφέρουν να παίξει στο μέλλον στην προσωπική του λίστα backlog.

User Priority: **5/5** Η διαχείριση των παιχνιδιών του χρήστη είναι η βασική λειτουργικότητα που το σύστημα παρέχει στο χρήστη.

Technical Priority: **5/5** Το σύστημα βασίζεται στην ιδέα ότι ο χρήστης το χρησιμοποιεί κυρίως για να **διαχειρίζεται** τα παιχνίδια που θέλει να παίξει στο μέλλον ή έχει παίξει στο παρελθόν.



<ΛΑ- 4>

Ο χρήστης πρέπει να μπορεί να αφαιρεί παιχνίδια από την προσωπική του λίστα backlog.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να αφαιρεί τα παιχνίδια που είτε τελείωσε, είτε παιχνίδια για τα οποία δεν ενδιαφέρεται πλέον από την προσωπική του λίστα backlog.

User Priority: **5/5** Η διαχείριση των παιχνιδιών του χρήστη είναι η βασική λειτουργικότητα που το σύστημα παρέχει στο χρήστη.

Technical Priority: **5/5** Το σύστημα βασίζεται στην ιδέα ότι ο χρήστης το χρησιμοποιεί κυρίως για να διαχειρίζεται τα παιχνίδια που θέλει να παίξει στο μέλλον ή έχει παίξει στο παρελθόν.

<ΛΑ- 5>

Ο χρήστης πρέπει να μπορεί να συντάξει μία κριτική για ένα παιχνίδι.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να συντάξει μια κριτική για ένα παιχνίδι που έχει παίξει, γράφοντας την γνώμη του και αναρτώντας την στην σελίδα του παιχνιδιού, ώστε να την δουν και άλλοι χρήστες.

User Priority: **3/5** Οι κριτικές δίνουν ένα κοινωνικό στοιχείο στην εμπειρία του χρήστη, καθώς την μοιράζεται με τους άλλους χρήστες του συστήματος, αλλά και τον βοηθούν στο να επιλέξει καλύτερα ποια παιχνίδια θέλει να παίξει.

Technical Priority: **4/5** Η αλληλεπίδραση μεταξύ χρηστών μέσω των κριτικών τους παροτρύνει περισσότερους χρήστες στο να χρησιμοποιήσουν το σύστημα για να μοιραστούν τις απόψεις τους για τα παιχνίδια που τους αρέσουν, με αποτέλεσμα το σύστημα να μεγαλώνει σε σκέλος.

<ΛΑ- 6>

Ο χρήστης πρέπει να μπορεί να βαθμολογήσει ένα παιχνίδι μέσα από την κριτική του.

Περιγραφή: Ο χρήστης πρέπει να μπορεί καθώς συντάσσει μια κριτική για ένα παιχνίδι να βαθμολογήσει το παιχνίδι με έναν βαθμό μεταξύ του 1 και των 5 αστεριών. Η αξιολόγηση αυτή θα εμφανίζεται στην κριτική του και θα επηρεάζει τον γενικό βαθμό του παιχνιδιού.

User Priority: **4/5** Η βαθμολόγηση των παιχνιδιών δίνει την δυνατότητα στον χρήστη να βρει γρηγορότερα και ευκολότερα παιχνίδια υψηλής ποιότητας.

Technical Priority: **4/5** Η βαθμολόγηση των παιχνιδιών κάνει ευκολότερη την πρόταση και την πρόταση παιχνιδιών που θα ταίριαζαν στη γεύση του χρήστη.

<ΛΑ- 7>

Ο χρήστης πρέπει να μπορεί να ανανεώσει την κριτική του για ένα παιχνίδι.

Περιγραφή: Ο χρήστης πρέπει να μπορεί αφού αναρτήσει την κριτική του για ένα παιχνίδι να την ανανεώσει, είτε προσθέτοντας ή ανανεώνοντας το βαθμό της κριτικής του, είτε ανανέωντας το κείμενο της κριτικής.

User Priority: **3/5** Είναι σημαντικό για τον χρήστη να έχει την δυνατότητα να ανανεώσει την κριτική του για ένα παιχνίδι σε περίπτωση που αλλάζει η γνώμη του για αυτό.

Technical Priority: **2/5** Το σύστημα μπορεί να λειτουργήσει χωρίς την δυνατότητα ανανέωσης μιας κριτικής.

<ΛΑ- 8>

Ο χρήστης πρέπει να μπορεί να αναζητήσει για ένα παιχνίδι με το τίτλο του.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να αναζητήσει, μέσω μιας φόρμας, ένα παιχνίδι με το τίτλο του και το σύστημα να επιστρέψει τα παιχνίδια που έχουν αυτό το τίτλο.



User Priority: 4/5 Η αναζήτηση παιχνιδιών είναι απαραίτητη για τον χρήστη, γιατί χωρίς αυτή δεν έχει κάποιο μέσω άμεσης εύρεσης παιχνιδιών.

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα της αναζήτησης

<ΛΑ- 9>

Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση το στούντιο ανάπτυξής τους.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση ποιο στούντιο ανάπτυξης ήταν υπεύθυνο για την δημιουργία του.

User Priority: 4/5 Το φιλτράρισμα μεταξύ παιχνιδιών αποτελεί πολύ χρήσιμο εργαλείο εύρεσης παιχνιδιών για τον χρήστη

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα του φιλτραρίσματος.

<ΛΑ- 10>

Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση τον εκδότη τους.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση ποιος εκδότης ήταν υπεύθυνος για την κυκλοφορία του.

User Priority: 4/5 Το φιλτράρισμα μεταξύ παιχνιδιών αποτελεί πολύ χρήσιμο εργαλείο εύρεσης παιχνιδιών για τον χρήστη

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα του φιλτραρίσματος.

<ΛΑ- 11>

Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση το είδος τους.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με σε τι είδος παιχνιδιού ανήκουν.

User Priority: 4/5 Το φιλτράρισμα μεταξύ παιχνιδιών αποτελεί πολύ χρήσιμο εργαλείο εύρεσης παιχνιδιών για τον χρήστη

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα του φιλτραρίσματος.

<ΛΑ- 12>

Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση την πλατφόρμα τους.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση τις πλατφόρμες στις οποίες έχουν κυκλοφορήσει.

User Priority: 4/5 Το φιλτράρισμα μεταξύ παιχνιδιών αποτελεί πολύ χρήσιμο εργαλείο εύρεσης παιχνιδιών για τον χρήστη

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα του φιλτραρίσματος.

<ΛΑ- 13>

Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση την σειρά τους.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να φιλτράρει παιχνίδια με βάση σε ποιας σειράς είναι μέλη.

User Priority: 4/5 Το φιλτράρισμα μεταξύ παιχνιδιών αποτελεί πολύ χρήσιμο εργαλείο εύρεσης παιχνιδιών για τον χρήστη

Technical Priority: 2/5 Το σύστημα δεν επωφελείται από την λειτουργικότητα του φιλτραρίσματος.



1.2 Χρήστες και εξωτερικά συστήματα

1.2.1 <Εγγεγραμμένος Χρήστης>

Ο βασικός χρήστης του συστήματος. Μπορεί να γράψει κριτικές, να βαθμολογήσει παιχνίδια και να διαχειριστεί την λίστα backlog του. Δεν έχει την δυνατότητα να εισάγει νέα παιχνίδια στη βάση δεδομένων ή να διαγράψει κριτικές άλλων χρηστών.

1.2.2 <Επισκέπτης>

Ο επισκέπτης έχει δυνατότητα να εγγραφεί στην εφαρμογή. Ο λόγος για τον οποίο ο επισκέπτης δεν έχει το δικαίωμα εύρεσης παιχνιδιών είναι για την επέκταση της εφαρμογής και του user base της.

1.2.3 <Κεντρική βάση δεδομένων BackloggedDB>

Μια σχεσιακή βάση δεδομένων που περιέχει τους πίνακες για τις αντίστοιχες οντότητες του συστήματος. Δεν θεωρήθηκαν ως μεμονωμένα εξωτερικά συστήματα οι πίνακες τις βάσεις.

1.2.3 <Discord OAuth2 API>

Το παρεχόμενο API του Discord για login μέσω του λογαριασμού discord. Μέσω αυτού ο χρήστης μπορεί να δημιουργήσει λογαριασμό και να συνδεθεί στο σύστημα.

1.3 Σημαντικές Μη Λειτουργικές Απαιτήσεις

<ΜΛΑ- 1>

Το σύστημα θα πρέπει να είναι σύμφωνο με τους ισχύοντες κανονισμούς προστασίας δεδομένων που τίθενται από το GDPR.

Περιγραφή: Το σύστημα θα πρέπει να προστατεύει τα προσωπικά δεδομένα των χρηστών.

User Priority: **4/5** Είναι σημαντικό για το χρήστη τα δεδομένα του να είναι ασφαλή.

Technical Priority: **5/5** Η μη τήρηση των κανονισμών προστασίας δεδομένων μπορεί θα αναγκάσει την εφαρμογή να τεθεί εκτός λειτουργίας.

<ΜΛΑ- 2>

Το σύστημα θα πρέπει να έχει native application για κινητές συσκευές και να υποστηρίζεται από μοντέρνους internet browsers.

Περιγραφή: Το σύστημα θα πρέπει να έχει εφαρμογές σχεδιασμένες για κινητές συσκευές Android με έκδοση Android 9 και άνω, καθώς και για συσκευές iPhone με έκδοση iOS 11 και άνω. Επίσης, θα πρέπει να υποστηρίζεται από Chromium based browsers με έκδοση 70 και άνω και Mozilla Firefox 60 και άνω.

User Priority: **3/5** Ο χρήστης θέλει να έχει πρόσβαση στο σύστημα ακόμα και από παλαιώτερες συσκευές.

Technical Priority: **3/5** Το σύστημα πρέπει να έχει την δυνατότητα να λειτουργεί σε διάφορες πλατφόρμες.



<ΜΛΑ- 3>

Το σύστημα θα πρέπει να έχει επιλογές προσβασιμότητας για άτομα με προβλήματα όρασης

Περιγραφή: Το σύστημα θα πρέπει να δίνει τη δυνατότητα στο χρήστη να αλλάζει το μέγεθος του κειμένου και την αντίθεση του χρώματος του κειμένου, ώστε να είναι πιο εύχρηστο για άτομα με προβλήματα όρασης. Επίσης, θα πρέπει να είναι προσβάσιμο από συσκευές screen readers.

User Priority: 4/5 Είναι σημαντικό το σύστημα να είναι όσο πιο εύχρηστο γίνεται.

Technical Priority: 1/5 Το σύστημα δεν επωφελείται από περισσότερες επιλογές προσβασιμότητας.

<ΜΛΑ- 4>

Το σύστημα θα πρέπει να συντηρήσιμο και επεκτάσιμο

Περιγραφή: Το σύστημα πρέπει να σχεδιαστεί έχοντας υπόψη τη συντηρησιμότητα και την επεκτασιμότητα, με μια αρχιτεκτονική που ακολουθεί **SOLID αρχές** και κώδικα που μπορεί εύκολα να αναβαθμιστεί, προκειμένου να διευκολυνθούν οι συνεχείς ενημερώσεις και αναπτύξεις.

User Priority: 1/5 Η εμπειρία του χρήστη δεν αλλάζει ριζικά σύμφωνα με την αρχιτεκτονική του συστήματος

Technical Priority: 4/5 Η εύκολη συντήρηση του συστήματος οδηγεί στην εξοικονόμηση πόρων και σε γρηγορότερη επίλυση τυχόν προβλημάτων που προκύψουν στο σύστημα.

<ΜΛΑ- 5>

Το σύστημα θα πρέπει να έχει χρόνο απόκρισης μικρότερο των 400ms.

Περιγραφή: Το σύστημα πρέπει να είναι γρήγορο κατά την χρήση του.

User Priority: 5/5 Η εμπειρία του χρήστη είναι άμεσα επηρεαζόμενη από το πόσο γρήγορη είναι η περιήγησή του.

Technical Priority: 2/5 Το σύστημα λειτουργεί κανονικά ακόμα και αν υπάρχει κάποια καθυστέρηση.

<ΜΛΑ- 6>

Το σύστημα θα πρέπει να υποστηρίζει πάνω από 10000 ταυτόχρονους χρήστες.

Περιγραφή: Το σύστημα πρέπει να είναι αξιόπιστο και να ανταποκρίνεται σε μεγάλο φόρτο λόγω user traffic.

User Priority: 5/5 Ο χρήστης πρέπει να έχει πρόσβαση στην εφαρμογή.

Technical Priority: 4/5 Σε περίπτωση που το σύστημα αποτύχει να υποστηρίξει τους χρήστες του, θα τεθεί προσωρινά εκτός λειτουργίας.

1.10 Λεξικογραφικοί Προσδιορισμοί

Χρήστης: Οποιοσδήποτε χρησιμοποιεί το σύστημα.

Σύστημα: Όλα τα συστήματα που συνεργάζονται για την λειτουργία της εφαρμογής.

Προσωπικά στοιχεία: Τα στοιχεία που δίνει ο χρήστης προκειμένου να έχει πρόσβαση στην εφαρμογή.

Παιχνίδι: Ένα βιντεοπαιχνίδι.

Στούντιο ανάπτυξης: Ένα στούντιο προγραμματιστών και καλλιτεχνών που έφτιαξε το παιχνίδι.

Εκδότης: Μια εταιρεία που είναι υπεύθυνη για τη διανομή του παιχνιδιού.

Σειρά: Μια σειρά βιντεοπαιχνιδιών.

Πλατφόρμα: Ένα μέσο με το οποίο μπορεί κάποιος να παίξει το παιχνίδι.



Κριτική: Μια αξιολόγηση ενός παιχνιδιού από το χρήστη.

Βαθμολογία: Μια βαθμολογία που μπορεί να δώσει προαιρετικά ο χρήστης, καθώς συντάσει την κριτική.

Χρόνος απόκρισης: Ο χρόνος που απαιτείται ώστε το σύστημα να επιστρέψει την ζητούμενη πληροφορία στο χρήστη.

Ταυτόχρονοι χρήστες: Ο συνολικός αριθμός των χρηστών που χρησιμοποιούν την εφαρμογή παράλληλα, μια συγκεκριμένη χρονική στιγμή.

Φίλτρο: Κριτήριο με το οποίο ο χρήστης ορίζει λεπτομερώς την αναζήτησή του.

Αποτέλεσμα αναζήτησης: Το σύνολο των παιχνιδιών που συμφωνούν με τα κριτήρια που έχει ορίσει ο χρήστης κατά την αναζήτησή του.

Επεκτασιμότητα: Η δυνατότητα για την εφαρμογή να μεγαλώσει σε σκέλος και απαιτήσεις.

Συντηρισμότητα: Η δυνατότητα για την ομάδα ανάπτυξης της εφαρμογής να προβλέπει λάθη στον κώδικα και να διορθώνει υπάρχοντα λάθη.



2 Σενάρια Χρήσης

Τα model files, καθώς και όλα τα διαγράμματα και gherkin features είναι διαθέσιμα και στο [GitHub](#).

2.1 Διάγραμμα σεναρίων χρήσης

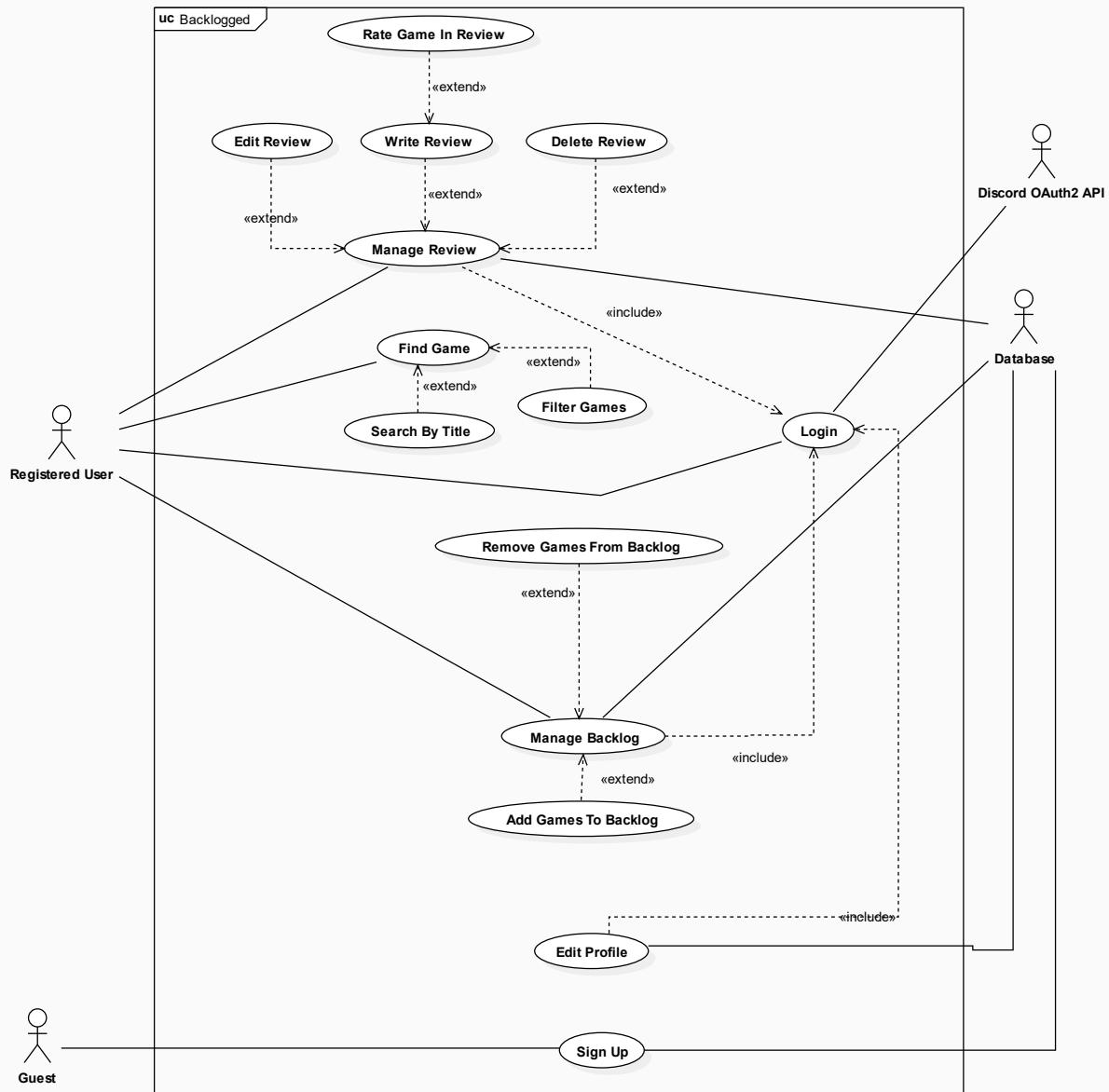


Diagram 1: Backlogged Use Case Diagram

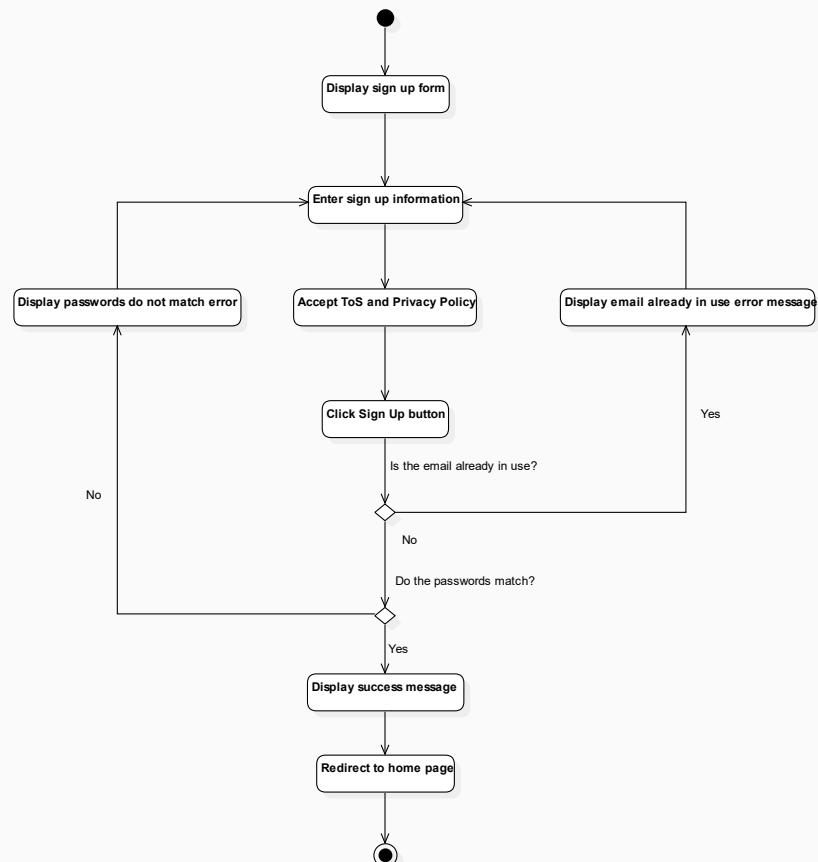


2.2 Feature 1: Sign Up

```

1  Feature: Sign up functionality
2    As a user
3      I want to create a new account
4      So that I can access the application
5
6  Scenario: Succesful sign up
7    Given I am on the sign up page
8    When I enter valid sign up information
9    And I check the accept the Terms of Service and Privacy Policy radio button
10   And I click the sign up button
11   Then I should be notified my account was created
12   And I should be redirected to my home page
13
14  Scenario: Sign up with existing email
15    Given I am on the sign up page
16    When I enter an email address that is already in use
17    And I click on the sign up button
18    Then I should see an error message that says "The email address is already in use"
19
20  Scenario: Passwords do not match
21    Given I am on the sign up page
22    When I enter a password
23    And I enter a different sign up password in the confirm password field
24    And I click the sign up button
25    Then I should see an error message that says "The passwords do not match"
26
27  Scenario: Haven't accepted the Terms of Service
28    Given I am on the sign up page
29    When I haven't checked the accept the Terms of Service and Privacy Policy radio button
30    Then the sign up button should be disabled
31

```



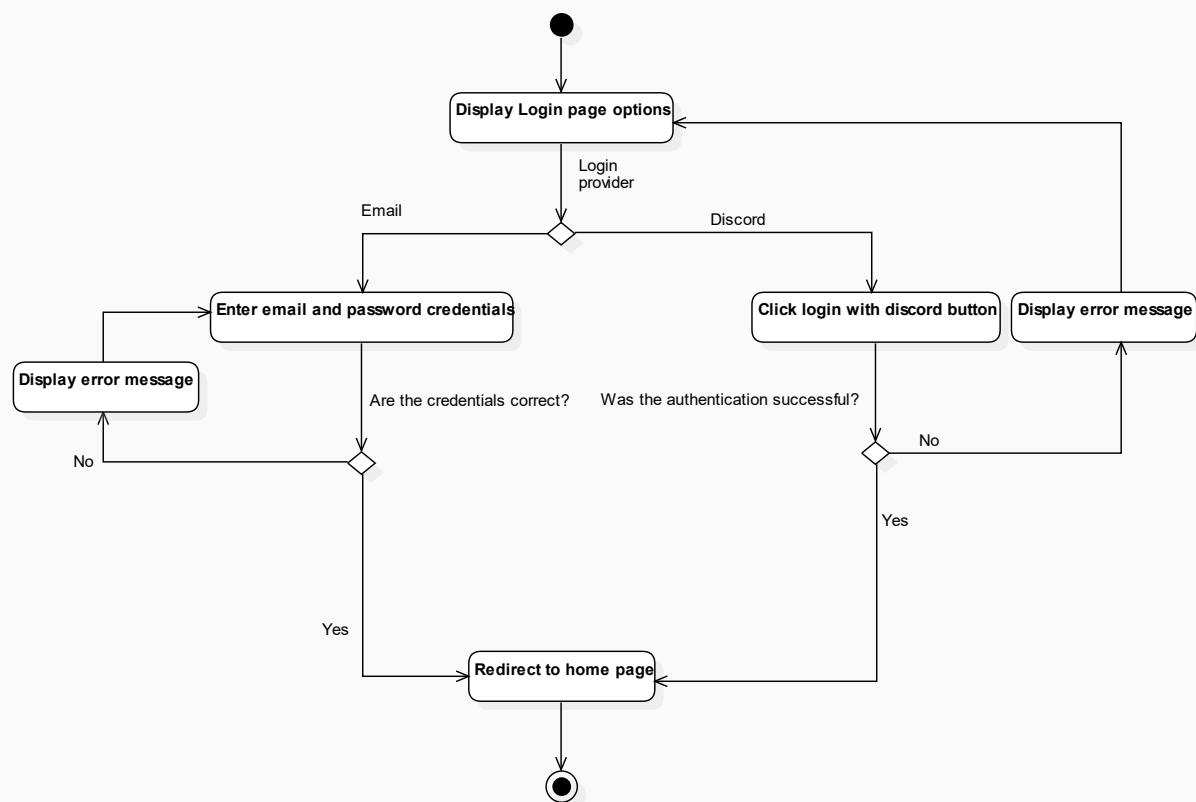


2.3 Feature 2: Login

```

1 Feature: Login functionality
2   As a user
3   I want to log in to my account
4   So that I can access my home page
5
6 Scenario: Succesful login with email and password
7   Given I am on the login page
8   When I enter valid email and password credentials
9   And I click the login button
10  Then I should be redirected to my home page
11
12 Scenario: Succesful login with Discord
13   Given I am on the login page
14   When I click on the login with Discord button
15   And I authorize the application to use my Discord credentials
16   Then I should be redirected to my home page
17
18 Scenario: Invalid email and password credentials
19   Given I am on the login page
20   When I enter invalid email and password credentials
21   And I click the login button
22   Then I should see an error message that says "The email/password combination is not correct"
23
24 Scenario: Unsuccessful Discord login
25   Given I am on the login page
26   When I click on the login with Discord button
27   And the authorization process of Discord fails
28   Then I should see an error message that says "The Discord login was unsuccesful, please try again or choose another form of login"

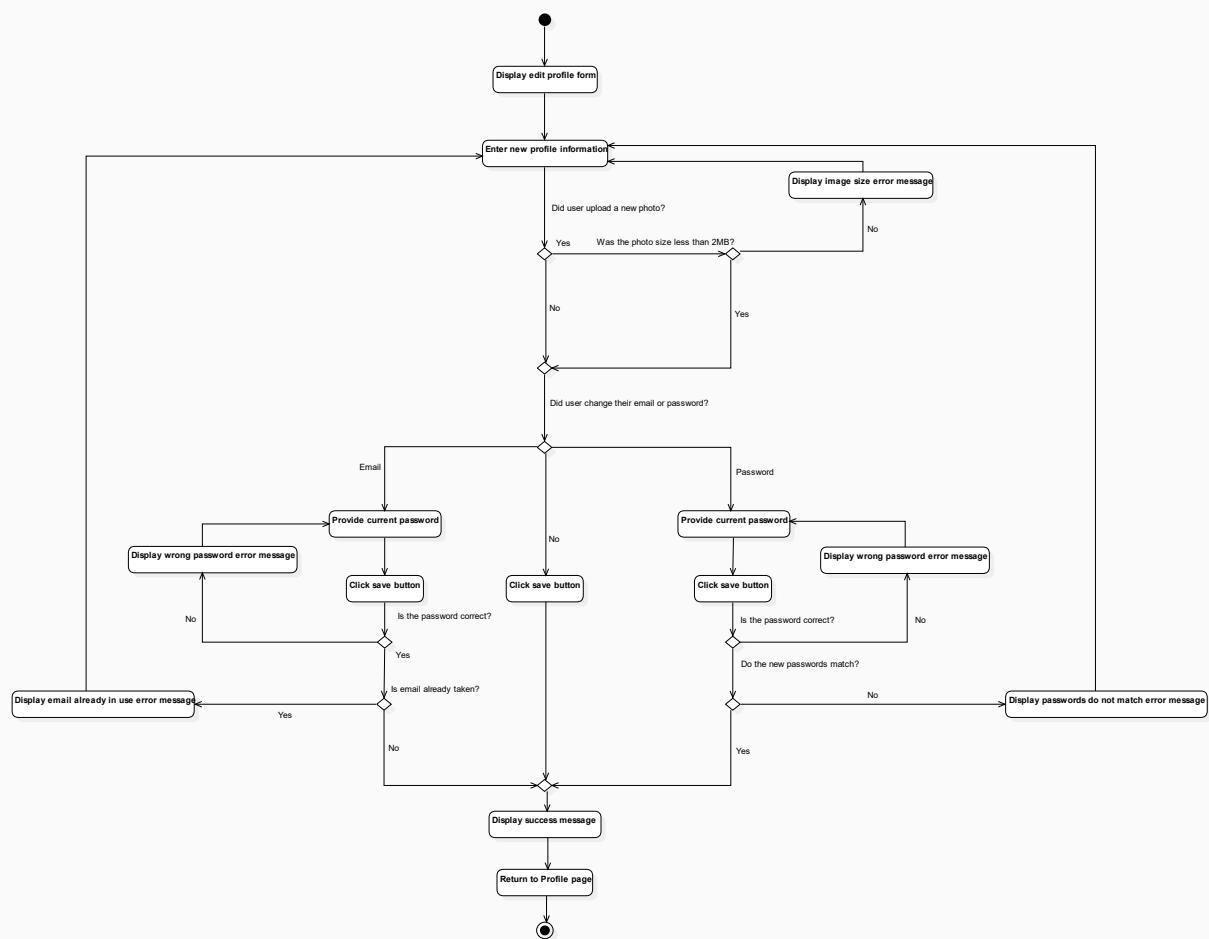
```





2.4 Feature 3: Edit Profile

```
1 Feature: Edit profile functionality
2   As a user
3     I want to edit my profile
4     So that I can keep my information up-to-date
5
6   Scenario: Succesful profile edit with no password or email change
7     Given I am on the edit profile page
8     When I enter valid updated profile information with no changes to password or email
9     And I click the save button
10    Then I should see a success message that says "Your profile has been updated"
11    And I should be redirected to my new profile page
12
13  Scenario: Succesful profile edit with password or email change
14    Given I am on the edit profile page
15    When I enter valid updated profile information with changes to password or email
16    And I enter the correct current password
17    And I click the save button
18    Then I should see a success message that says "Your profile has been updated"
19    And I should be redirected to my new profile page
20
21  Scenario: Edit profile with invalid image size
22    Given I am on the edit profile page
23    When I upload a profile image that is larger than 2MB
24    And I click the save button
25    Then I should see an error message that says "The image is too large"
26
27  Scenario: Passwords do not match
28    Given I am on the edit profile page
29    When I enter a new password
30    And I enter a different password in the confirm password field
31    And I provide the correct current password
32    And I click the save button
33    Then I should see an error message that says "The passwords do not match"
34
35  Scenario: Email is already taken
36    Given I am on the edit profile page
37    When I enter an email that is already taken
38    And I provide the correct current password
39    And I click the save button
40    Then I should see an error message that says "The email has already been taken"
41
42  Scenario: Current password is wrong
43    Given I am on the edit profile page
44    When I enter valid updated profile information with changes to password or email
45    And I provide the wrong current password
46    And I click the save button
47    Then I should see an error message that says "The current password is incorrect"
```



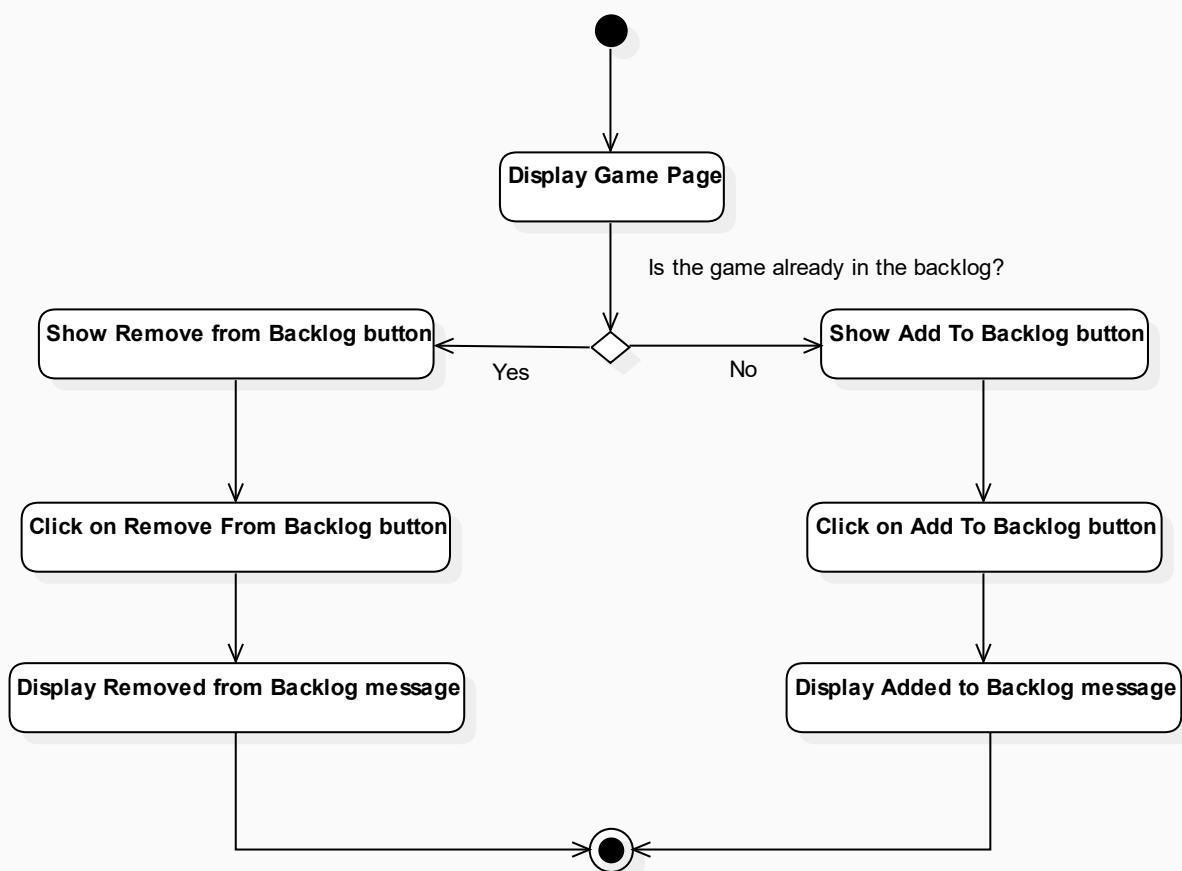


2.5 Feature 4: Manage Backlog

```

1 Feature: Manage backlog
2   As a user
3     I want to manage my backlog
4     So that I can keep track of my games
5
6 Scenario: Add game to backlog
7   Given I am on the game page
8   When I click on the "Add to backlog" button
9   Then I should see a confirmation message
10  And I should see the game in my backlog when I go to my backlog
11
12 Scenario: Remove game from backlog
13  Given I am on the game page
14  And the game is already in my backlog
15  When I click on the "Remove from backlog" button
16  Then I should see a confirmation message
17  And I should not see the game in my backlog when I go to my backlog

```



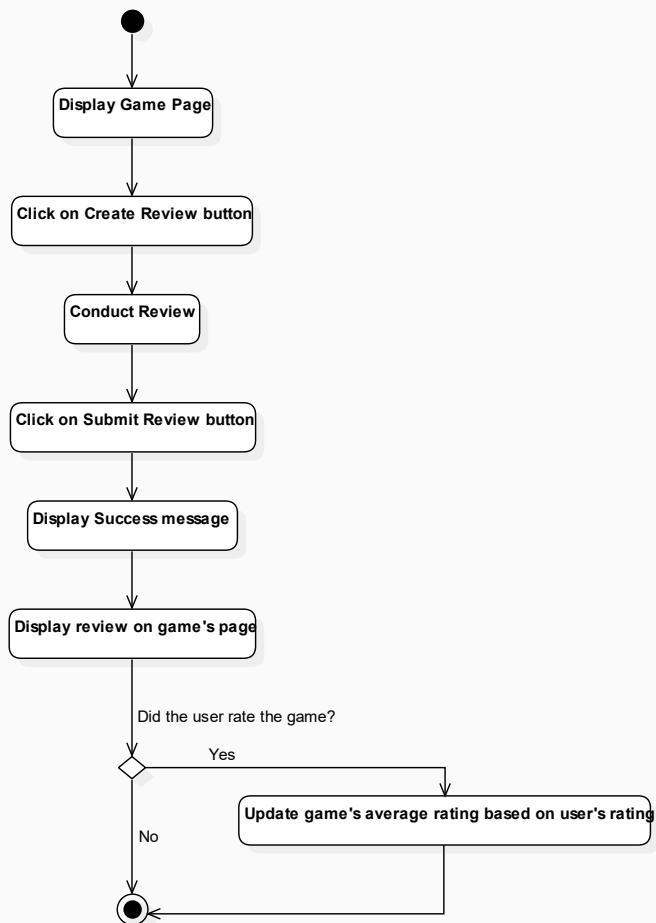


2.6 Feature 5: Create Review

```

1 Feature: Review games and rate them
2 As a user
3 I want to review games and optionally give them a rating
4 So that I can share my thoughts and opinions with others
5
6 Scenario: Review a game without rating
7   Given I am on a game's page
8   When I write a review for the game
9   And I submit the review
10  Then I should see a success message
11  And my review should be visible on the game's page
12  And the review should not have a rating yet
13
14 Scenario: Review a game with rating
15   Given I am on a game's page
16   When I write a review for the game
17   And I give the game a rating of 4 stars
18   And I submit the review
19   Then I should see a success message
20   And my review should be visible on the game's page
21   And the review's rating should be updated to 4 stars
22   And the game's average rating should be updated based on my rating

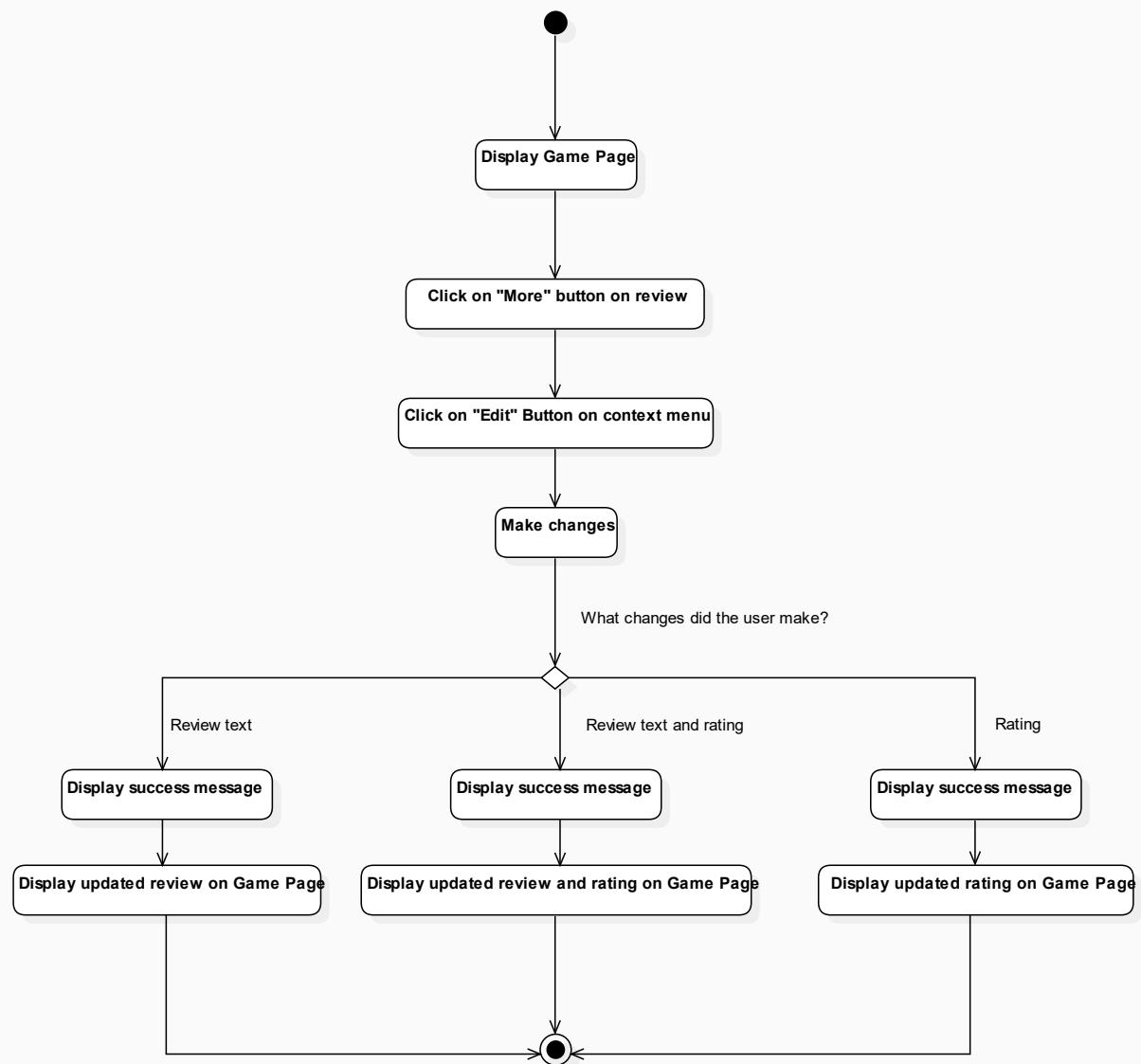
```





2.7 Feature 6: Edit Review

```
1  Feature: Edit review or rating for a game
2    As a user
3      I want to edit my review or rating for a game
4      So that I can rewrite my review or change my rating
5
6    Scenario: Edit review text and rating
7      Given I am on a game's page
8      And I have already written a review for a game
9      When I click on the "More" button on my review
10     And I click on the "Edit" button
11     And I change my review's rating and text
12     And I click on the "Save" button
13     Then I should see a success message
14     And I should see the updated review and rating on the game's page
15
16  Scenario: Edit rating
17    Given I am on a game's page
18    And I have already written a review for a game
19    When I click on the "More" button on my review
20    And I click on the "Edit" button
21    And I change my review's rating
22    And I click on the "Save" button
23    Then I should see a success message
24    And I should see the updated rating on the game's page
25
26  Scenario: Edit review text
27    Given I am on a game's page
28    And I have already written a review for a game
29    When I click on the "More" button on my review
30    And I click on the "Edit" button on the context menu
31    And I update the review text
32    And I click on the "Save" button
33    Then I should see a success message
34    And I should see the updated review on the game's page
```



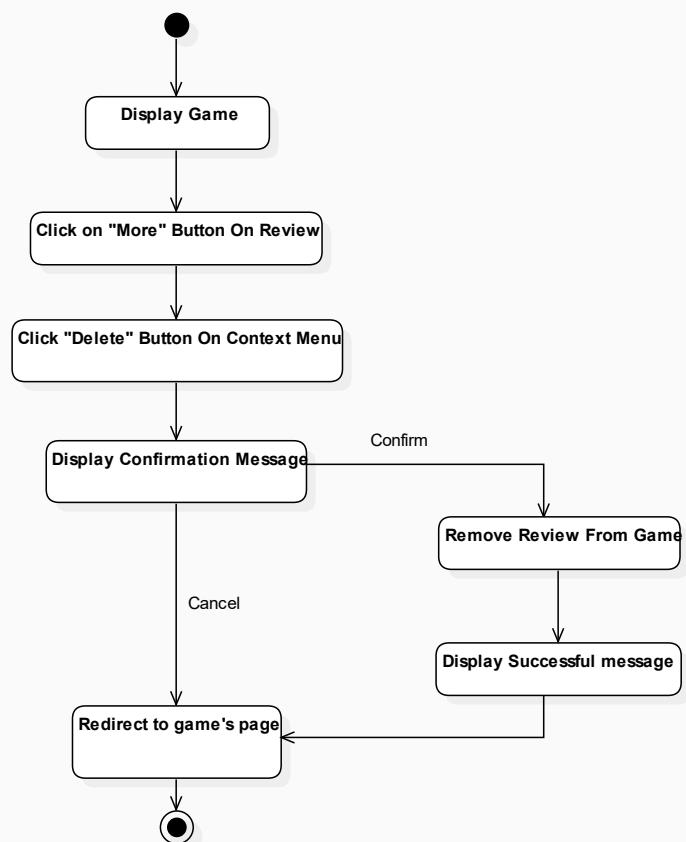


2.8 Feature 7: Delete Review

```

1 Feature: Delete a review for a game
2 As a user
3 I want to delete my review for a game
4 So that I can remove my review from the game page.
5
6 Scenario: Delete review
7 Given I am on game's page
8 And I have already written a review for a game
9 When I click on the "More" button on the review
10 And I click "Delete" button on the context menu
11 And I click the "Delete" button on the confirmation modal
12 Then I should see a success message
13 And the review should be removed from the game page
14
15 Scenario: Cancel Deletion
16 Given I am on game's page
17 And I have already written a review for a game
18 When I click on the "More" button on the review
19 And I click "Delete" button on the context menu
20 And I click the "Cancel" button on the confirmation modal
21 Then the review should still be visible on the game page

```



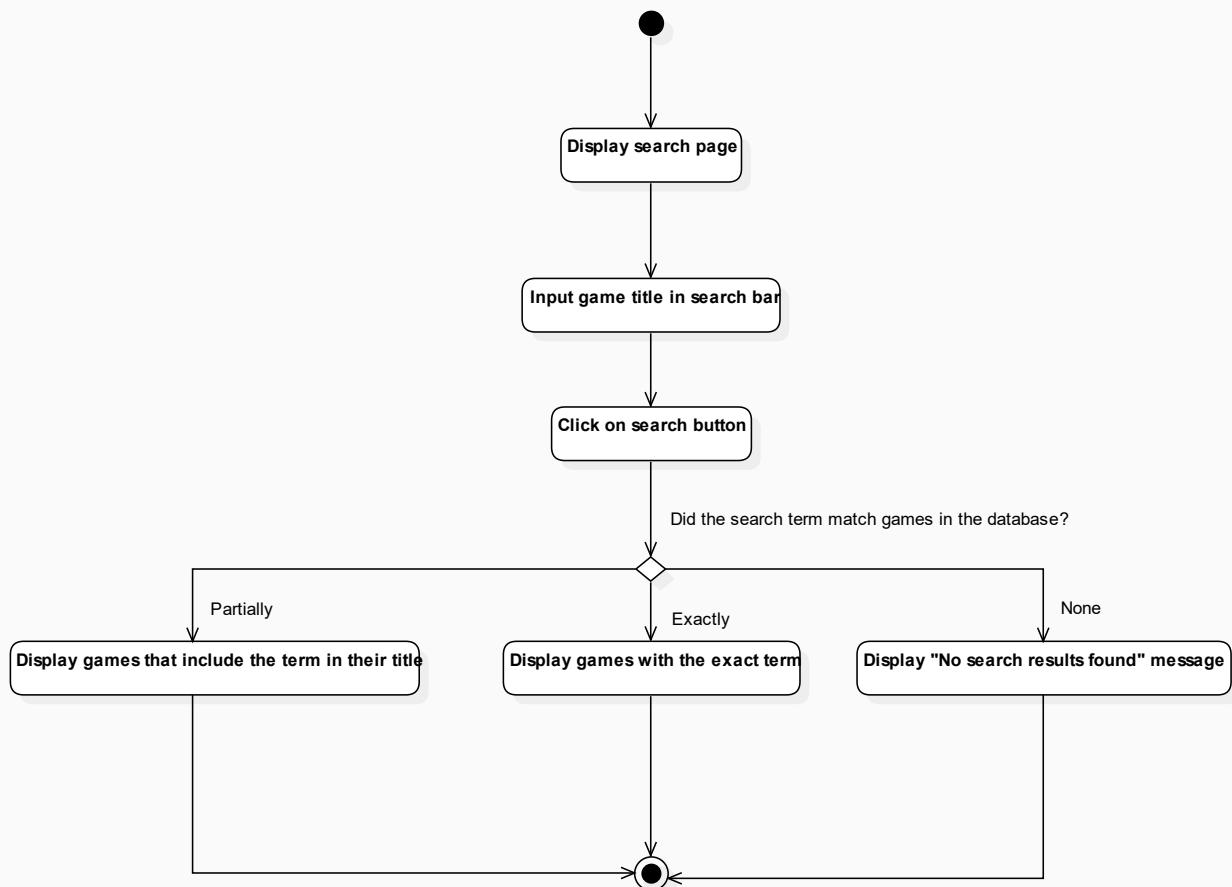


2.9 Feature 8: Search Game by Title

```

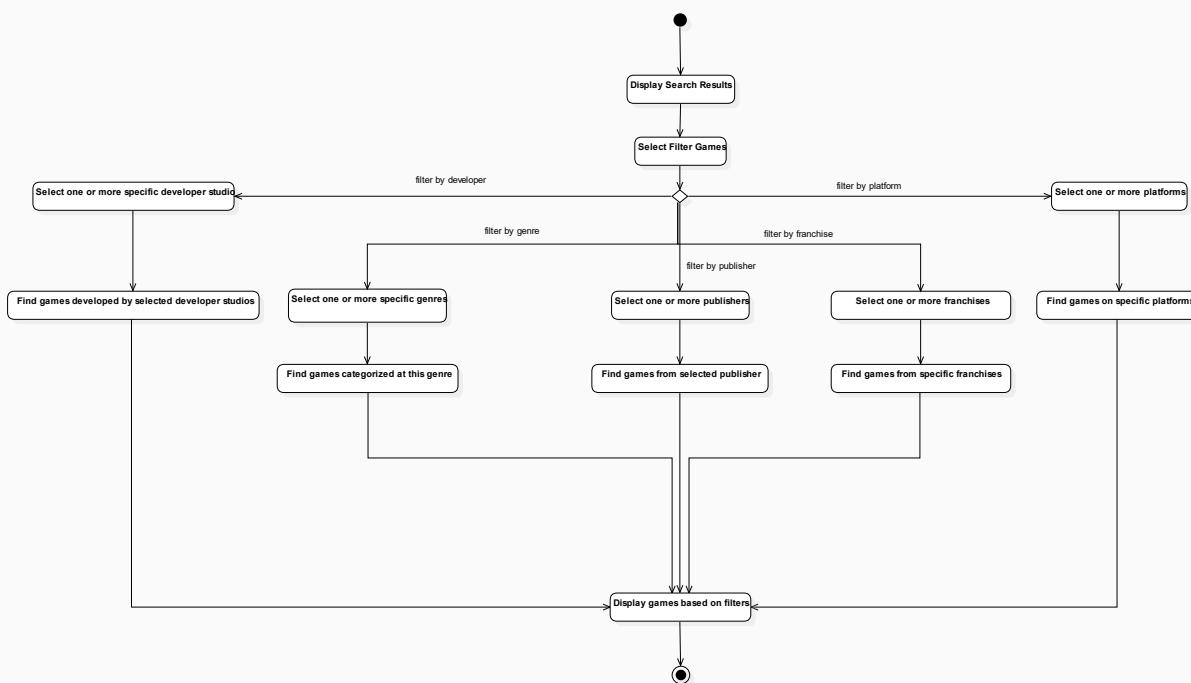
1 Feature: Search a game by title
2   As a user
3     I want to search for games by title
4     So that I can find specific games
5
6 Scenario: Search for a game by exact title match
7   Given I am on the search page
8   When I type "The Legend of Zelda: Breath of the Wild" into the search bar
9   And I click the search button
10  Then I should see a list of search results that includes "The Legend of Zelda: Breath of the Wild"
11
12 Scenario: Search for games by partial title match
13   Given I am on the search page
14   When I type "Mario" into the search bar
15   And I click the search button
16   Then I should see a list of search results that includes games with "Mario" in their title
17
18 Scenario: No search results found
19   Given I am on the search page
20   When I type "abcdefg" into the search bar
21   And I click the search button
22   Then I should see a message that says "No search results found"

```





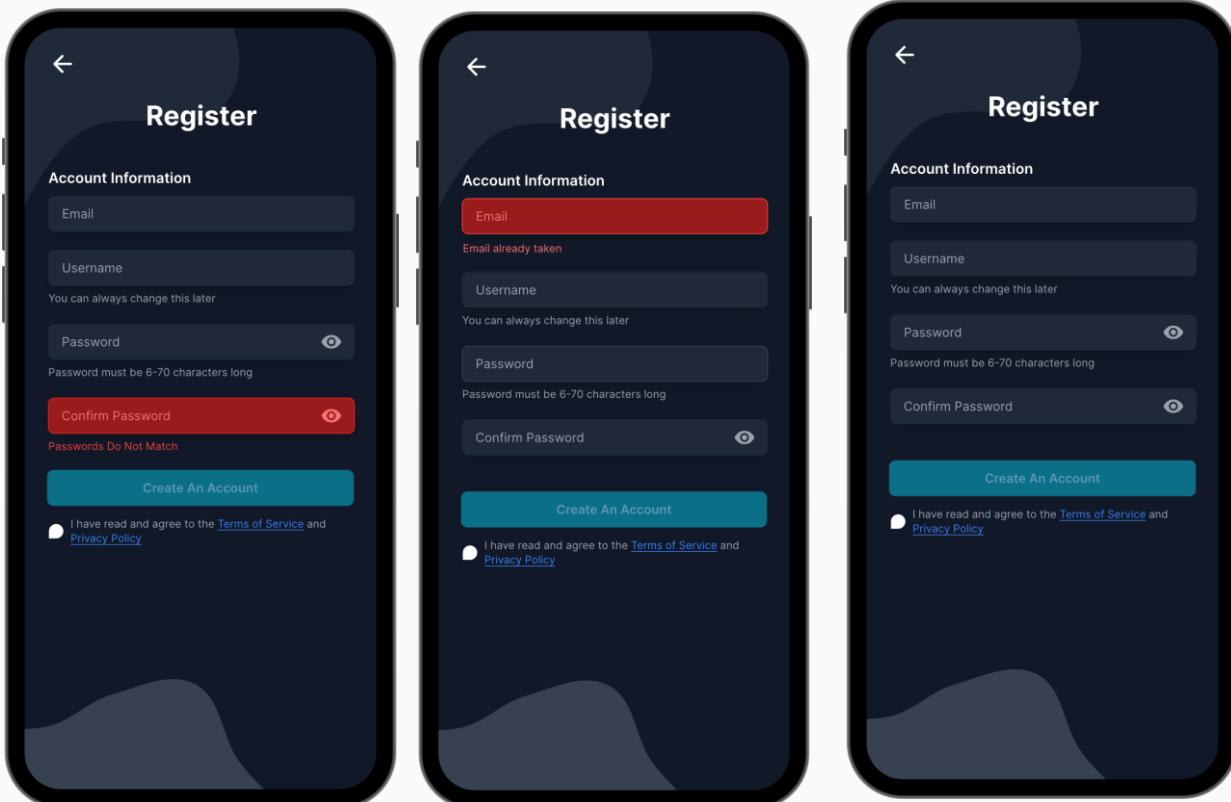
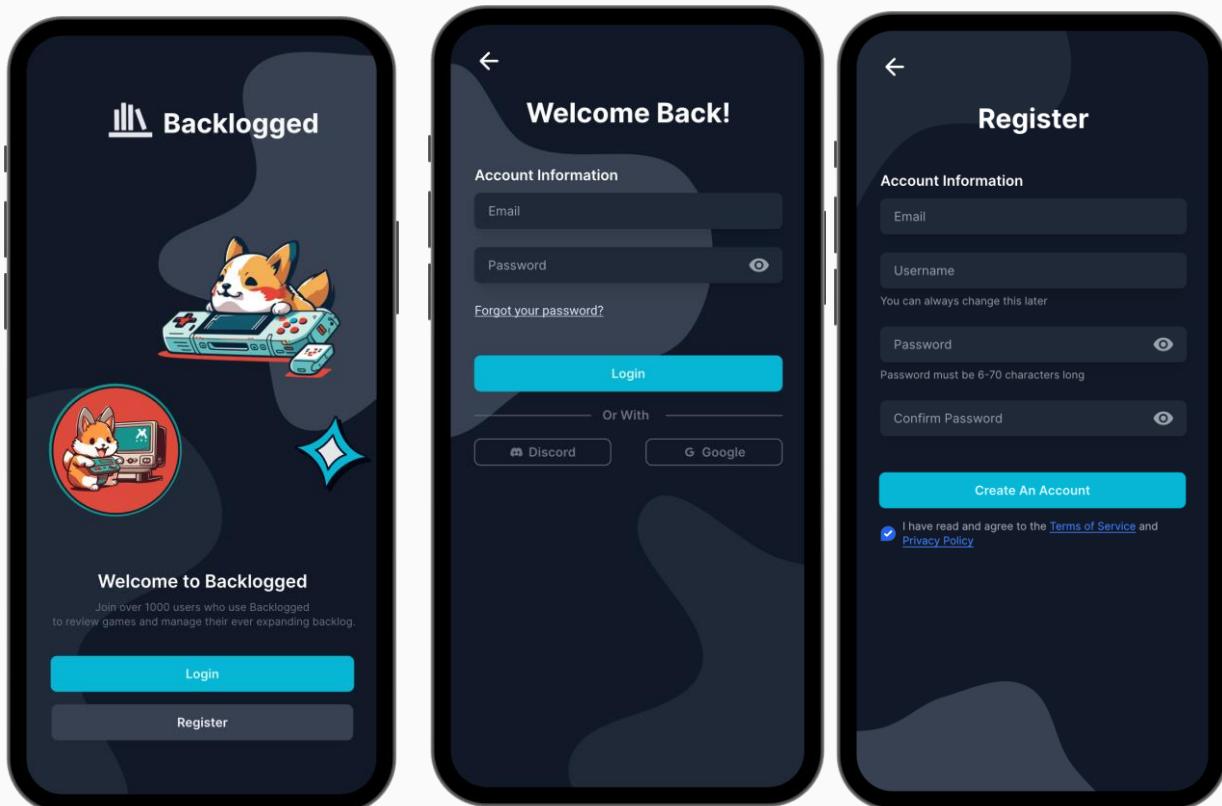
2.10 Feature 9: Filter Search Results





3 Επιδεικτικά γραφικά παράθυρα διεπαφής

3.1 Splash Screen, Login, Register





3.2 Game Page, Add Review

Persona 5
2017 • P-Studio +2 • SEGA

Available on: +1 Add To Backlog

JRPG Turn Based Social Sim Immersive Story-Drama

Wake Up Get Up Get Out There!

Persona 5 marks the return of the award-winning franchise on home consoles since the PS2 generation, and is the first numbered Read More...

Ratings
4.3 ★★★★★ based on 1000 reviews [+ Add A Review](#)

John Doe ★★★★★
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla enim purus, gravida Read More
299 likes 5000 views

John Doe ★★★★★
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla enim purus, gravida Read More
299 likes 5000 views

16

Ratings
4.3 ★★★★★ based on 1000 reviews [+ Add A Review](#)

Home Search Profile

John Doe

All Reviews

Users Also Liked

Persona 4 Golden 2012 • JRPG +1 **4.5★**

Shin Megami 2011 • JRPG

Home Search Profile

Persona 5
2017 • P-Studio +2 • SEGA

Available on: +1 Add To Backlog

JRPG Turn Based Social Sim Immersive Story-Drama

Wake Up Get Up Get Out There!

Persona 5 marks the return of the award-winning franchise on home consoles since the PS2 generation, and is the first numbered Read More...

Ratings
4.3 ★★★★★

Persona 5

Platforms

Developers
 ATLUS P Studio

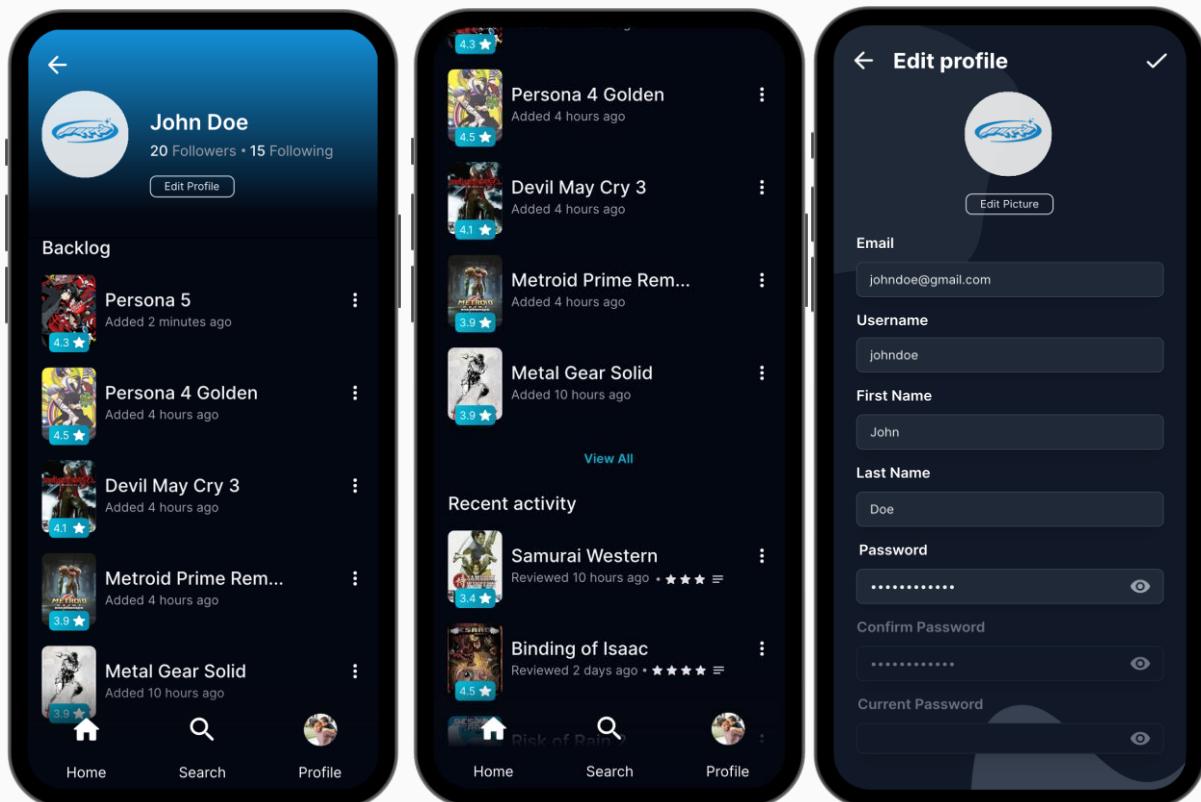
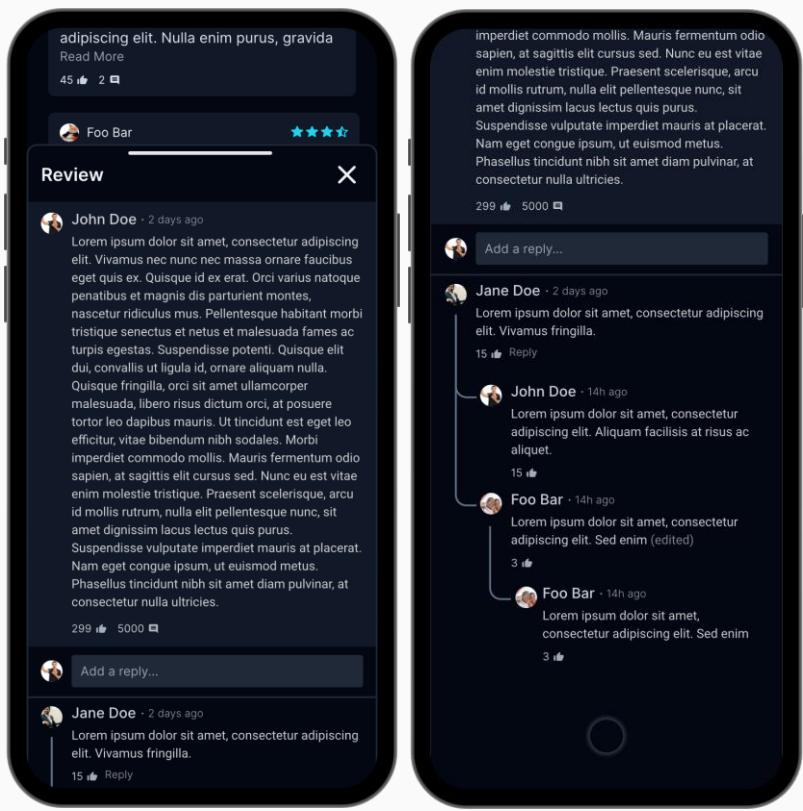
Publishers
 SEGA Of America ATLUS Japan

How did you like this game?
Your feedback helps other people find games they'll enjoy.

Tell us why... [Submit](#)

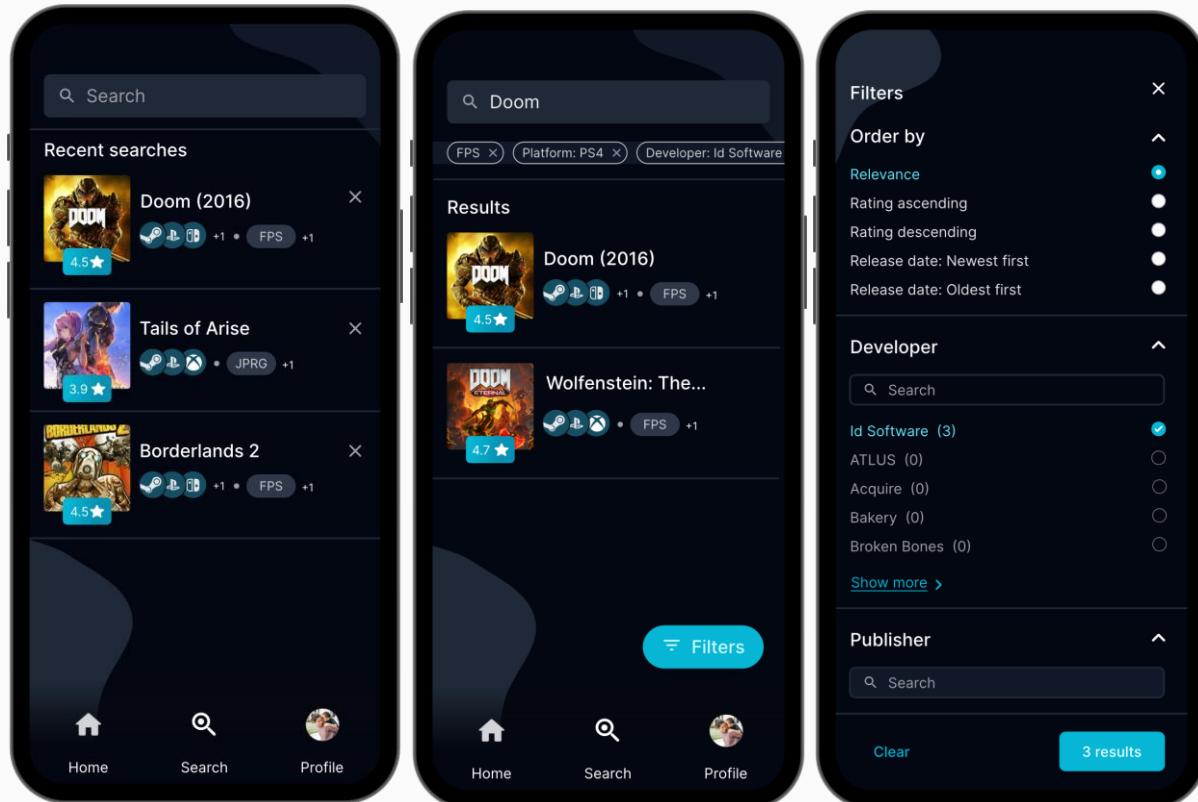
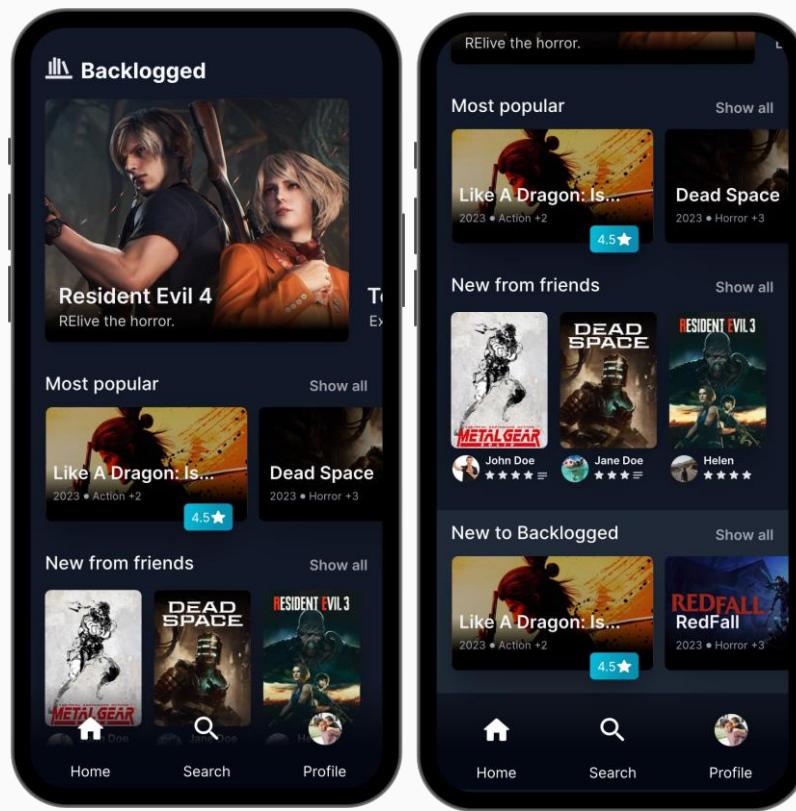


3.3 Review Details, Profile, Edit Profile





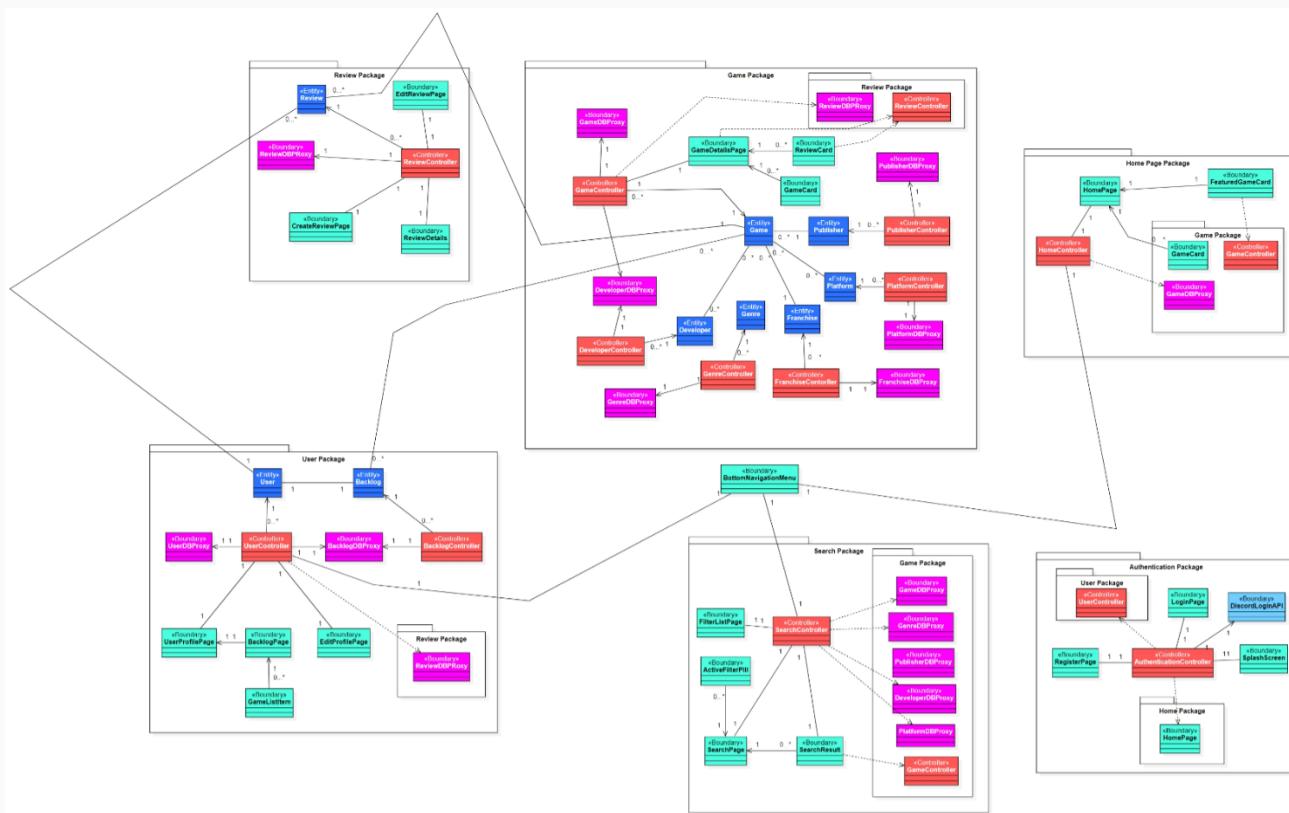
3.3 Home Page, Search Page, Filter Page





4 Στατική Μοντελοποίηση

Διάγραμμα Κλάσεων



«Boundary» BottomNavigationMenu

<p>-homeButton: Button -searchButton: Button -profileButton: Button</p>
<p>+BottomNav(homeButton: Button, searchButton: Button, profileButton: Button): BottomNavigationView +homeClick(): void +profileClick(): void +searchClick(): void</p>

Χαρακτηριστικά κλάσης

- <homeButton: **Button
- <searchButton: **Button**>: Το κουμπί με το εικονίδιο μεγενθυτικού φακού στο κάτω μέρος της οθόνης.
- <profileButton: **Button**>: Το κουμπί με την φωτογραφία προφίλ του χρήστη στο κάτω μέρος της οθόνης.**

Μέθοδοι κλάσης

- <**BottomNavigationView(homeButton: Button, searchButton: Button, profileButton: Button): BottomNavigationView**>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

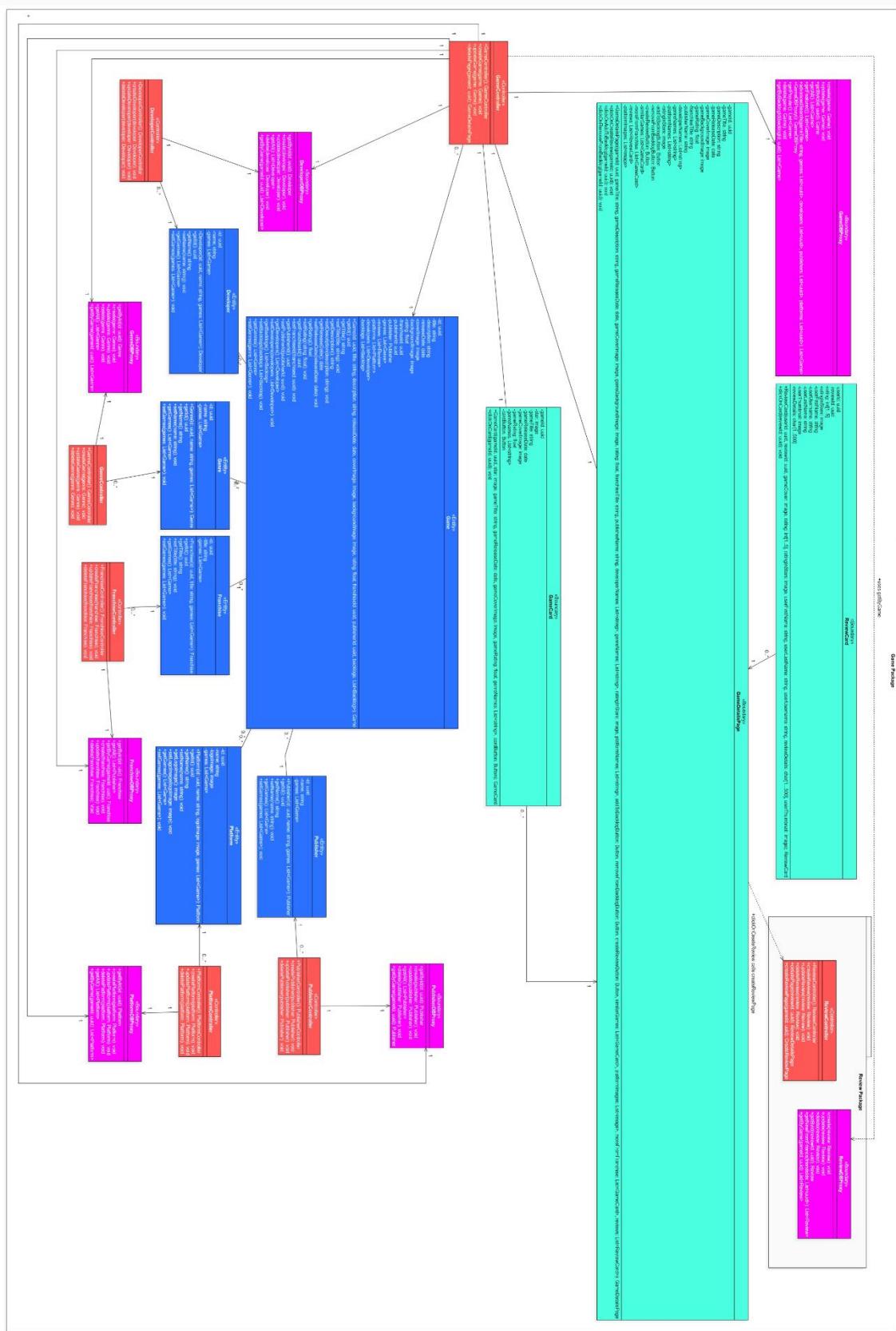


- <`homeClick(): void`>: Καλεί την μέθοδο `displayHomePage(): HomePage` του `HomeController` όταν πατηθεί το `homeButton`.
- <`profileClick(): void`>: Καλεί τη μέθοδο `profilePage(userId: uuid): ProfilePage` του `UserController` όταν πατηθεί το `profileButton`. Το `userId` δίνεται μέσω του `SignInManager` της εφαρμογής.
- <`searchClick(): void`>: Καλεί τη μέθοδο `displaySearchPage(): SearchPage` του `SearchController` όταν πατηθεί το `searchButton`.

ΣΗΜΕΙΩΣΗ: Οι οντότητες που έχουν σχέση μεταξύ τους στη βάση συνδέονται αντίστοιχα και στο διάγραμμα κλάσεων. Επίσης, τα χαρακτηριστικά τους είναι ίδια με αυτά των αντίστοιχων σχέσεων της βάσης, ώστε να γίνεται εύκολη και σωστή απεικόνιση των σχέσεων της βάσης σε αντικείμενα στον κώδικα.



4.1 Game Package





Οντότητες

<Game>

Η κλάση του παιχνιδιού.

Χαρακτηριστικά κλάσης

- <**id**: `uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <**title**: `string`>: Ο επίσημος τίτλος του παιχνιδιού στην αγορά.
- <**description**: `string`>: Μια σύντομη περιγραφή του παιχνιδιού.
- <**releaseDate**: `date`>: Η ημερομηνία κυκλοφορίας του παιχνιδιού.
- <**coverImage**: `image`>: Η φωτογραφία αφίσα του παιχνιδιού. Χρησιμοποιείται στην σελίδα **GamePage** του παιχνιδιού, καθώς και στις κάρτες **GameCard**, στην αναζήτηση στα αποτελέσματα **SearchResult** και στις λίστες των Backlog και στις λίστες πρόσφατης δραστηριότητας **GameListItem**.
- <**backgroundImage**: `Image`>: Η φωτογραφία φόντου του παιχνιδιού. Χρησιμοποιείται στην σελίδα του παιχνιδιού πίσω από τον τίτλο του.
- <**rating**: `float`>: Ο μέσος όρος βαθμολογήσεων του παιχνιδιού από τις κριτικές του.
- <**franchiseId**: `uuid`>: Το ξένο κλειδί για την σύνδεση με την οντότητα σειράς **Franchise**.
- <**publisherId**: `uuid`>: Το ξένο κλειδί για την σύνδεση με την οντότητα εκδότη **Publisher**.
- <**publisher**: `Publisher`>: Ο εκδότης **Publisher** που εξέδωσε το παιχνίδι.
- <**franchise**: `Franchise`>: Η σειρά **Franchise** στην οποία ανήκει το παιχνίδι.
- <**developers**: `List<Developer>`>: Η λίστα των στούντιο ανάπτυξης **Developer** του παιχνιδιού.
- <**genres**: `List<Genre>`>: Η λίστα ειδών **Genre** που αντιπροσωπεύουν το παιχνίδι.
- <**reviews**: `List<Review>`>: Η λίστα των κριτικών **Review** που έχουν γραφτεί για το παιχνίδι.
- <**platforms**: `List<Platform>`>: Η λίστα των πλατφόρμων **Platform** στα οποία έχει κυκλοφορήσει το παιχνίδι.
- <**backlogs**: `List<Backlog>`>: Η λίστα από λίστες **Backlog** στα οποία βρίσκεται μέσα το παιχνίδι.

Μέθοδοι κλάσης

- <`Game(id: uuid, title: string, description: string, releaseDate: date, coverImage: image, backgroundImage: image, rating: float, franchiseId: uuid, publisherId: uuid, publisher: Publisher, franchise: Franchise, developers: List<Developer>, genres: List<Genre>, reviews: List<Review>, platforms: List<Platform>, backlogs: List<Backlog>)`>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`getId(): uuid`>: Επιστρέφει την μεταβλητή `id`.
- <`getTitle(): string`>: Επιστρέφει την μεταβλητή `title`.
- <`setTitle(title: string): void`>: Αλλάζει την μεταβλητή `title` με την τιμή που δίνεται ως όρισμα.



- <`getDescription(): string`>: Επιστρέφει την μεταβλητή `description`.
- <`setDescription(description: string): void`>: Αλλάζει την μεταβλητή `description` με την τιμή που δίνεται ως όρισμα.

- <`getReleaseDate(): date`>: Επιστρέφει την μεταβλητή `releaseDate`.
- <`setReleaseDate(releaseDate: date): void`>: Αλλάζει την μεταβλητή `releaseDate` με την τιμή που δίνεται ως όρισμα.

- <`getRating(): float`>: Επιστρέφει την μεταβλητή `rating`.
- <`setRating(rating: float): void`>: Αλλάζει την μεταβλητή `rating` με την τιμή που δίνεται ως όρισμα.

- <`getFranchiseId(): uuid`>: Επιστρέφει την μεταβλητή `franchiseId`.
- <`setFranchiseId(franchiseId: uuid): void`>: Αλλάζει την μεταβλητή `franchiseId` με την τιμή που δίνεται ως όρισμα.

- <`getFranchise(): Franchise`>: Επιστρέφει την μεταβλητή `franchise`.
- <`setFranchise(franchise: Franchise): void`>: Αλλάζει την μεταβλητή `franchise` με την τιμή που δίνεται ως όρισμα.

- <`getPublisherId(): uuid`>: Επιστρέφει την μεταβλητή `publisherId`.
- <`setPublisherId(publisherId: uuid): void`>: Αλλάζει την μεταβλητή `publisherId` με την τιμή που δίνεται ως όρισμα.

- <`getPublisher(): Publisher`>: Επιστρέφει την μεταβλητή `publisher`.
- <`setPublisher(publisher: Publisher): void`>: Αλλάζει την μεταβλητή `publisher` με την τιμή που δίνεται ως όρισμα.

- <`getDevelopers(): List<Developer>`>: Επιστρέφει την μεταβλητή `developers`.
- <`setDevelopers(developers: List<Developer>): void`>: Αλλάζει την μεταβλητή `developers` με την τιμή που δίνεται ως όρισμα.

- <`getBacklogs(): List<Backlog>`>: Επιστρέφει την μεταβλητή `backlogs`.
- <`setBacklogs(backlogs: List<Backlog>): void`>: Αλλάζει την μεταβλητή `backlogs` με την τιμή που δίνεται ως όρισμα.

- <`getGenres(): List<Genre>`>: Επιστρέφει την μεταβλητή `genres`.
- <`setGenres(genres: List<Genre>): void`>: Αλλάζει την μεταβλητή `genres` με την τιμή που δίνεται ως όρισμα.

ΣΗΜΕΙΩΣΗ 1: Από αυτό το σημείο και πέρα, οι συναρτήσεις `getter` και `setter` θα παραλείπονται για εξοικονόμηση χώρου.

ΣΗΜΕΙΩΣΗ 2: Η απουσία του `setter` της μεταβλητής `id` είναι σκόπιμη καθώς ορίζεται από τη βάση δεδομένων και η αλλαγή της μπορεί να οδηγήσει σε προβλήματα κατά την λειτουργία του συστήματος.



«Entity»
Franchise
-id: <code>uuid</code>
-title: <code>string</code>
-games: <code>List<Game></code>
+Franchise(id: <code>uuid</code> , title: <code>string</code> , games: <code>List<Game></code>): Franchise
+getId(): <code>uuid</code>
+getTitle(): <code>string</code>
+setTitle(title: <code>string</code>): void
+getGames(): <code>List<Game></code>
+setGames(games: <code>List<Game></code>): void

<Franchise>

Η κλάση για τις σειρές παιχνιδιών.

Χαρακτηριστικά κλάσης

- <`id: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <`title: string`>: Ο επίσημος τίτλος της σειράς στην αγορά.
- <`games: List<Game>`>: Η λίστα των παιχνιδιών `Game` που έχουν ανήκουν στην σειρά.

Μέθοδοι κλάσης

- <`Franchise(id: uuid, title: string, games: List<Game>): Franchise`>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

«Entity»
Developer
-id: <code>uuid</code>
-name: <code>string</code>
-games: <code>List<Game></code>
+Developer(id: <code>uuid</code> , name: <code>string</code> , games: <code>List<Game></code>): Developer
+getId(): <code>uuid</code>
+getName(): <code>string</code>
+setName(name: <code>string</code>): void
+getGames(): <code>List<Game></code>
+setGames(games: <code>List<Game></code>): void

<Developer>

Η κλάση των στούντιο ανάπτυξης παιχνιδιών.

Χαρακτηριστικά κλάσης

- <`id: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <`name: string`>: Το όνομα του στούντιο ανάπτυξης.
- <`games: List<Game>`>: Η λίστα των παιχνιδιών `Game` που το στούντιο έχει αναπτύξει.

Μέθοδοι κλάσης



- <Developer(id: `uuid`, name: `string`, games: `List<Game>`): `Developer`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

«Entity» Publisher	
-id: <code>uuid</code>	
-name: <code>string</code>	
-games: <code>List<Game></code>	
+Publisher(id: <code>uuid</code> , name: <code>string</code> , games: <code>List<Game></code>): <code>Publisher</code>	
+getId(): <code>uuid</code>	
+getName(): <code>string</code>	
+setName(name: <code>string</code>): <code>void</code>	
+getGames(): <code>List<Game></code>	
+setGames(games: <code>List<Game></code>): <code>void</code>	

<Publisher>

Η κλάση για τους εκδότες παιχνιδιών.

Χαρακτηριστικά κλάσης

- <`id: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <`name: string`>: Το όνομα του εκδότη.
- <`games: List<Game>`>: Η λίστα των παινιχδιών `Game` που ο εκδότης έχει κυκλοφορήσει.

Μέθοδοι κλάσης

- <`Publisher(id: uuid, name: string, games: List<Game>): Publisher`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

«Entity» Genre	
-id: <code>uuid</code>	
-name: <code>string</code>	
-games: <code>List<Game></code>	
+Genre(id: <code>uuid</code> , name: <code>string</code> , games: <code>List<Game></code>): <code>Genre</code>	
+getId(): <code>uuid</code>	
+getName(): <code>string</code>	
+setName(name: <code>string</code>): <code>void</code>	
+getGames(): <code>List<Game></code>	
+setGames(games: <code>List<Game></code>): <code>void</code>	

<Genre>

Η κλάση για το είδος ενός παιχνιδιού.



Χαρακτηριστικά κλάσης

- <id: **uuid**>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <name: **string**>: Το όνομα του είδους.
- <games: **List<Game>**>: Η λίστα των παιχνιδιών **Game** που ανήκουν στο είδος.

Μέθοδοι κλάσης

- <**Genre**(id: **uuid**, name: **string**, games: **List<Game>**): **Genre**>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

«Entity» Platform	
-id: uuid	
-name: string	
-logolimage: image	
-games: List<Game>	
+Platform(id: uuid , name: string , logolimage: image , games: List<Game>): Platform	
+getId(): uuid	
+getName(): string	
+setName(name: string): void	
+getLogolimage(): image	
+setLogolimage(logolimage: image): void	
+getGames(): List<Game>	
+setGames(games: List<Game>): void	

<Platform>

Η κλάση για μια πλατφόρμα στην οποία κυκλοφορεί κάποιο παιχνίδι.

Χαρακτηριστικά κλάσης

- <id: **uuid**>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <name: **string**>: Το όνομα της πλατφόρμας.
- <logolimage: **image**>: Η φωτογραφία του λογότυπου της πλατφόρμας.
- <games: **List<Game>**>: Η λίστα των παιχνιδιών **Game** που έχουν κυκλοφορήσει στην πλατφόρμα.

Μέθοδοι κλάσης

- <**Platform**(id: **uuid**, name: **string**, games: **List<Game>**): **Platform**>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.



Ελεγκτές



<GameController>

Μέθοδοι κλάσης

- <`GameController()`: `GameController`>:
Η συνάρτηση δόμησης της κλάσης.
- <`detailsPage(gameId: uuid): GameDetailsPage`>: Επιστρέφει το οριακό αντικείμενο της σελίδας του παιχνιδιού `GameDetailsPage`, καλώντας
 1. την συνάρτηση `getById(gameId: uuid): Game` του `GameDBProxy` για τις λεπτομέρειες του παιχνιδιού,
 2. την συνάρτηση `getByGame(gameId: uuid): List<Review>` του `ReviewDBProxy` για τις κριτικές του παιχνιδιού,
 3. την συνάρτηση `getByGame(gameId: uuid): List<Genre>` του `GenreDBProxy` για τα είδη του παιχνιδιού,
 4. την συνάρτηση `getByGame(gameId: uuid): List<Platform>` του `PlatformDBProxy` για τις πλατφόρμες που έχει κυκλοφορήσει το παιχνίδι,
 5. την συνάρτηση `getByGame(gameId: uuid): List<Developer>` του `DeveloperDBProxy` για τα στούντιο ανάπτυξης του παιχνιδιού,
 6. την συνάρτηση `getByGame(gameId: uuid): Publisher` του `PublisherDBProxy` για τον εκδότη,
 7. την συνάρτηση `getByGame(gameId: uuid): Franchise` του `FranchiseDBProxy` για την σειρά,

Αφού πάρει τις πληροφορίες από τις συναρτήσεις, ανανεώνει τις πληροφορίες του αντικειμένου `Game`. Για τα παρόμοια παιχνίδια με βάση το είδος, τρέχει ένας αλγόριθμος με όρισμα την λίστα των ειδών του παιχνιδιού. Για τα παιχνίδια της σειράς καλείται η συνάρτηση `getByFranchise(franchiseId: uuid): List<Game>` του `GameDBProxy`.

Έπειτα, δημιουργεί τα αντίστοιχα `GameCard` αντικείμενα και δημιουργγεί το οριακό αντικείμενο `GameDetailsPage`.

- <`createGame(game: Game): void`>: Προσθέτει ένα καινούργιο παιχνίδι στη βάση δεδομένων, καλώντας την συνάρτηση `create(game: Game)` του `GameDBProxy`.
- <`updateGame(game: Game): void`>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος παιχνιδιού στη βάση δεδομένων, καλώντας την συνάρτηση `update(game: Game)` του `GameDBProxy`.
- <`deleteGame(game: Game): void`>: Διαγράφει ένα παιχνίδι από τη βάση δεδομένων, μέσω της συνάρτησης `delete(game: Game): void` του `GameDBProxy`.



<DeveloperController>

Μέθοδοι κλάσης

- <`DeveloperController(): DeveloperController`>:
Η συνάρτηση δόμησης της κλάσης.
- <`createDeveloper(developer: Developer): void`>: Προσθέτει ένα καινούργιο στούντιο στη βάση δεδομένων, καλώντας την συνάρτηση `create(developer: Developer)` του `DeveloperDBProxy`.
- <`updateDeveloper(developer: Developer): void`>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος στούντιο στη βάση δεδομένων, καλώντας την συνάρτηση `update(developer: Developer)` του `DeveloperDBProxy`.
- <`deleteDeveloper(developer: Developer): void`>: Διαγράφει ένα στούντιο από τη βάση δεδομένων, μέσω της συνάρτησης `delete(developer: Developer): void` του `DeveloperDBProxy`.



<FranchiseController>

Μέθοδοι κλάσης

- <`FranchiseController(): FranchiseController`>:
Η συνάρτηση δόμησης της κλάσης.
- <`createFranchise(franchise: Franchise): void`>: Προσθέτει μια καινούργια σειρά παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση `create(franchise: Franchise)` του `FranchiseDBProxy`.
- <`updateFranchise(franchise: Franchise): void`>: Αλλάζει τα χαρακτηριστικά μιας υπάρχουσας σειράς παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση `update(franchise: Franchise)` του `FranchiseDBProxy`.
- <`deleteFranchise(franchise: Franchise): void`>: Διαγράφει μια σειρά παιχνιδιών από τη βάση δεδομένων, μέσω της συνάρτησης `delete(franchise: Franchise): void` του `FranchiseDBProxy`.



<PublisherController>

Μέθοδοι κλάσης

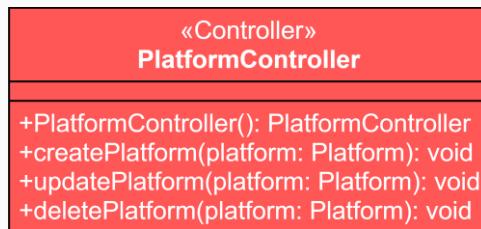
- <**PublisherController(): PublisherController**>:
Η συνάρτηση δόμησης της κλάσης.
- <**createPublisher(publisher: Publisher): void**>: Προσθέτει ένα καινούργιο εκδότη παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση **create(publisher: Publisher)** του **PublisherDBProxy**.
- <**updatePublisher(publisher: Publisher): void**>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος εκδότη παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση **update(publisher: Publisher)** του **PublisherDBProxy**.
- <**deletePublisher(publisher: Publisher): void**>: Διαγράφει ένα εκδότη παιχνιδιών από τη βάση δεδομένων, μέσω της συνάρτησης **delete(publisher: Publisher): void** του **PublisherDBProxy**.



<GenreController>

Μέθοδοι κλάσης

- <**GenreController(): GenreController**>:
Η συνάρτηση δόμησης της κλάσης.
- <**createGenre(genre: Genre): void**>: Προσθέτει ένα καινούργιο είδος παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση **create(genre: Genre)** του **GenreDBProxy**.
- <**updateGenre(genre: Genre): void**>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος είδους παιχνιδιών στη βάση δεδομένων, καλώντας την συνάρτηση **update(genre: Genre)** του **GenreDBProxy**.
- <**deleteGenre(genre: Genre): void**>: Διαγράφει ένα είδος παιχνιδιών από τη βάση δεδομένων, μέσω της συνάρτησης **delete(genre: Genre): void** του **GenreDBProxy**.



<PlatformController>

Μέθοδοι κλάσης

- <`PlatformController(): PlatformController`>: Η συνάρτηση δόμησης της κλάσης.
- <`createPlatform(platform: Platform): void`>: Προσθέτει μια καινούργια πλατφόρμα στη βάση δεδομένων, καλώντας την συνάρτηση `create(platform: Platform)` του `PlatformDBProxy`.
- <`updatePlatform(platform: Platform): void`>: Αλλάζει τα χαρακτηριστικά μιας πλατφόρμας στη βάση δεδομένων, καλώντας την συνάρτηση `update(platform: Platform)` του `PlatformDBProxy`.
- <`deletePlatform(platform: Platform): void`>: Διαγράφει μια πλατφόρμα από τη βάση δεδομένων, μέσω της συνάρτησης `delete(platform: Platform): void` του `PlatformDBProxy`.

<ReviewController>

Αναφέρεται στο [Review Package](#).

Οριακά Αντικείμενα



<GameCard>

Χαρακτηριστικά κλάσης

- <`gameId: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `Game`, χρησιμοποιείται για την μεταφορά του χρήστη στην αντίστοιχη σελίδα του παιχνιδιού.
- <`star: image`>: Το εικονίδιο του αστεριού
- <`gameTitle: string`>: Ο επίσημος τίτλος του παιχνιδιού στην αγορά.
- <`gameReleaseDate: date`>: Η ημερομηνία κυκλοφορίας του παιχνιδιού.
- <`gameCoverImage: image`>: Η φωτογραφία αφίσα του παιχνιδιού.
- <`gameRating: float`>: Ο μέσος όρος βαθμολογήσεων του παιχνιδιού από τις κριτικές του.
- <`genreNames: List<string>`>: Η λίστα με τα ονόματα των ειδών `Genre` που αντιπροσωπεύουν το παιχνίδι.



- <cardButton: **Button**>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στην σελίδα του παιχνιδιού. Κρατάει το id του παιχνιδιού ώστε να καλέσει την **detailsPage**(gameId: **uuid**) του **GameController**.

ΣΗΜΕΙΩΣΗ : Το αντικείμενο **Button** έχει τα χαρακτηριστικά των

- <Type: **Enum**>: Το είδος του (primary, secondary, tertiary)
- <entityType: **Enum**>: Η οντότητα με την οποία σχετίζεται (αν σχετίζεται)
- <entityId: **uuid**>: Το id της οντότητας με την οποία σχετίζεται (αν σχετίζεται)
- <label: **string**>: Η ετικέτα του

Μέθοδοι κλάσης

- <**GameCard**(gameId: **uuid**, star: **image**, gameTitle: **string**, gameReleaseDate: **date**, gameCoverImage: **image**, genres: **List<string>**, cardButton: **Button**): **GameCard**>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <**clickOnCard**(gameId: **uuid**): **void**>: Καλεί την συνάρτηση την **detailsPage**(gameId: **uuid**) του **GameController**.

«Boundary» ReviewCard	
-userId: uuid	
-reviewId: uuid	
-rating: int[1..5]	
-ratingInStars: image	
-userFirstName: string	
-userUsername: string	
-userLastName: string	
-userThumbnail: image	
-reviewDetails: char[1..500]	
+ReviewCard(userId: uuid , reviewId: uuid , gameCover: image , rating: int[1..5] , ratingInStars: image , userFirstName: string , userLastName: string , userUsername: string , reviewDetails: char[1..500] , userThumbnail: image): ReviewCard	
+clickOnCard(reviewId: uuid): void	

<ReviewCard>

Χαρακτηριστικά κλάσης

- <reviewId: **uuid**>: Το βασικό κλειδί και ευρετήριο της οντότητας **Review**, χρησιμοποιείται για να μεταφερθεί ο χρήστης στην σελίδα της κριτικής.
- <userId: **uuid**>: Το βασικό κλειδί της οντότητας **User**, χρησιμοποιείται για να μεταφερθεί ο χρήστης στην σελίδα προφίλ του χρήστη.
- <rating: **int[1..5]**>: Η βαθμολογία της κριτικής.
- <ratingInStars: **image**>: Η αντίστοιχη βαθμολογία σε εικόνα με αστέρια.
- <userFirstName: **string**>: Το όνομα του χρήστη.
- <userLastName: **string**>: Το επώνυμο του χρήστη.
- <userUsername: **string**>: Το username του χρήστη.
- <userThumbnail: **image**>: Η φωτογραφία προφίλ του χρήστη.
- <reviewDetails: **char[1..500]**>: Οι λεπτομέρειες της κριτικής.

Μέθοδοι κλάσης

- <**ReviewCard**(reviewId: **uuid**, userId: **uuid**, rating: **int[1..5]**, ratingInStars: **image**, userFirstName: **string**, userLastName: **string**, userUsername: **string**, userThumbnail: **image**, reviewDetails: **char[1..500]**): **GameCard**>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.



- <`clickOnCard(reviewId: uuid): void`>: Καλεί την συνάρτηση την `detailsPage(reviewId: uuid)` του `ReviewController`.

«Boundary» GameDetailsPage	
-gameId: <code>uuid</code>	
-gameTitle: <code>string</code>	
-gameDescription: <code>string</code>	
-gameReleaseDate: <code>date</code>	
-gameCoverImage: <code>image</code>	
-gameBackgroundImage: <code>image</code>	
-gameRating: <code>float</code>	
-franchiseTitle: <code>string</code>	
-publisherName: <code>string</code>	
-developerNames: <code>List<string></code>	
-genreNames: <code>List<string></code>	
-platformNames: <code>List<string></code>	
-ratingInStars: <code>image</code>	
-addToBacklogButton: <code>Button</code>	
-removeFromBacklogButton: <code>Button</code>	
-createReviewButton: <code>Button</code>	
-similarGames: <code>List<GameCard></code>	
-moreFromFranchise: <code>List<GameCard></code>	
-reviews: <code>List<ReviewCard></code>	
+clickOnCreateReview(gameId: <code>uuid</code>): <code>void</code>	
+clickOnAddToBacklog(gameId: <code>uuid</code>): <code>void</code>	
+clickOnRemoveFromBacklog(gameId: <code>uuid</code>): <code>void</code>	

ΣΗΜΕΙΩΣΗ: Στην παραπάνω φωτογραφία παραλείπεται η συνάρτηση δόμησης, επειδή είναι πολύ μεγάλη. Βρίσκεται κανονικά στο διάγραμμα του πακέτου.

<`GameDetailsPage`>

Χαρακτηριστικά κλάσης

- <`gameId: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <`gameTitle: string`>: Ο επίσημος τίτλος του παιχνιδιού στην αγορά.
- <`gameDescription: string`>: Μια σύντομη περιγραφή του παιχνιδιού.
- <`gameReleaseDate: date`>: Η ημερομηνία κυκλοφορίας του παιχνιδιού.
- <`gameCoverImage: image`>: Η φωτογραφία αφίσα του παιχνιδιού.
- <`gameBackgroundImage: Image`>: Η φωτογραφία φόντου του παιχνιδιού. Χρησιμοποιείται στην σελίδα του παιχνιδιού πίσω από τον τίτλο του.
- <`gameRating: float`>: Ο μέσος όρος βαθμολογήσεων του παιχνιδιού από τις κριτικές του.
- <`franchiseTitle: string`>: Ο τίτλος της σειράς στην οποία ανήκει το παιχνίδι.
- <`publisherName: string`>: Το όνομα του εκδότη του παιχνιδιού.
- <`developerNames: List<string>`>: Η λίστα με τα ονόματα των στούντιο ανάπτυξης `Developer` του παιχνιδιού.
- <`genreNames: List<string>`>: Η λίστα με τα ονόματα των ειδών `Genre` που αντιπροσωπεύουν το παιχνίδι.
- <`platformNames: List<string>`>: Η λίστα με τα ονόματα των πλατφόρμων `Platform` στα οποία έχει κυκλοφορήσει το παιχνίδι.
- <`platformImages: List<images>`>: Η λίστα με τις φωτογραφίες λογότυπα των πλατφόρμων `Platform` στα οποία έχει κυκλοφορήσει το παιχνίδι.
- <`ratingInStars: image`>: Η βαθμολογία του παιχνιδιού σε αστέρια.
- <`addToBacklogButton: Button`>: Το κουμπί που προσθέτει το παιχνίδι στην λίστα `Backlog` του χρήστη.



- <removeFromBacklogButton: **Button**>: Το κουμπί που αφαιρεί το παιχνίδι από την λίστα **Backlog** του χρήστη.
- <createReviewButton: **Button**>: Το κουμπί που μεταφέρει τον χρήστη στην οθόνη δημιουργίας κριτικής για το παιχνίδι
- <similarGames: **List<GameCard>**>: Η λίστα από τα παιχνίδια που είναι παρόμοια με αυτό με βάση τα είδη το αντιπροσωπεύουν.
- <moreFromFranchise: **List<GameCard>**>: Η λίστα από τα παιχνίδια που ανήκουν στην ίδια σειρά με το παιχνίδι.
- <reviews: **List<ReviewCard>**>: Η λίστα από τις κριτικές του παιχνιδιού ως οριακό αντικείμενο **ReviewCard**

Μέθοδοι κλάσης

- <**GameDetailsPage**(gameId: **uuid**, gameTitle: **string**, gameDescription: **string**, gameReleaseDate: **date**, gameCoverImage: **image**, gameBackgroundImage: **Image**, rating: **float**, franchiseTitle: **string**, publisherName: **string**, developerNames: **List<string>**, genreNames: **List<string>**, ratingInStars: **image**, platformNames: **List<string>**, addToBacklogButton: **Button**, removeFromBacklogButton: **Button**, createReviewButton: **Button**, similarGames: **List<GameCard>**, moreFromFranchise: **List<GameCard>**, reviews: **List<ReviewCard>**): **GameDetailsPage**>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <**clickOnCreateReview**(gameId: **uuid**): **void**>: Καλεί την συνάρτηση **createReviewPage**(gameId: **uuid**): του **ReviewController**.
- <**clickOnAddToBacklog**(gameId: **uuid**): **void**>: Καλεί την συνάρτηση **addGame**(gameId: **uuid**, backlogId: **uuid**): **void** του **BacklogController**. Το backlogId βρίσκεται μέσω του **BacklogDBProxy** από το id του χρήστη **User**.
- <**clickOnRemoveFromBacklog**(gameId: **uuid**): **void**>: Καλεί την συνάρτηση **removeGame**(gameId: **uuid**, backlogId: **uuid**): **void** του **BacklogController**. Το backlogId βρίσκεται μέσω του **BacklogDBProxy** από το id του χρήστη **User**.

ΣΗΜΕΙΩΣΗ 1: Δε θα δωθούν περαιτέρω φωτογραφίες των κλάσεων, επειδή θα ξεπεραστεί το όριο των 70 σελίδων. Σε περίπτωση που λόγω μεγέθους των διαγράμματων δεν είναι αρκετά καθαρή η εικόνα (Για αυτό ευθύνονται οι συναρτήσεις δόμησης, καθώς καλούν όλες τις παραμέτρους), είναι δυνατή η λήψη των αρχείων στο [GitHub](#), στον φάκελο [diagrams](#). Η εισαγωγή των διαγραμμάτων σε μορφή SVG ήταν αδύνατη μέσω Word, εμφανίζοντας απλώς μια λευκή εικόνα.

ΣΗΜΕΙΩΣΗ 2: Ο λόγος που βρίσκεται η κλάση **ReviewCard** στο συγκεκριμένο πακέτο και όχι στο πακέτο **Review Package** είναι ότι δεν συνδέεται με καμία κλάση στο πακέτο εκείνο. Η ανάθεση των πληροφοριών του αντικειμένου οντότητας **Review** που επιστρέφει η **getByGame**(gameId: **uuid**): **List<Review>** του **ReviewDBProxy** σε οριακό αντικείμενο **ReviewCard** γίνεται μέσω του **GameController**. Είναι απαραίτητο, σύμφωνα με τις αρχές SOLID, να υπάρχει όσο το δυνατόν μικρότερη σύνδεση μεταξύ κλάσεων (*loose coupling*) και δεν συνηθίζεται ένας ελεγκτής να καλεί μια συνάρτηση ενός άλλου (με μερικές εξαιρέσεις). Για αυτό υπάρχουν οι κλάσεις **DBProxies**, προκειμένου να απομονωθεί η data access layer από το business logic.



<GameDBProxy>

Μέθοδοι κλάσης

- <**create**(game: Game): void>: Καλεί query για δημιουργία ενός παιχνιδιού στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος game.
- <**update**(game: Game): void>: Καλεί query για ανανέωση ενός παιχνιδιού στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος game.
- <**delete**(game: Game): void>: Καλεί query για διαγραφή ενός παιχνιδιού στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος game.
- <**getById**(gameId: uuid): Game>: Καλεί query για εύρεση ενός παιχνιδιού στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <**getAll()**: List<Game>>: Καλεί query για εύρεση όλων των παιχνιδιών της βάσης.
- <**getFeatured()**: List<Game>>: Καλεί query για εύρεση των καινούργιων παιχνιδιών της βάσης με σκοπό να τα διαφημίσει στην αρχική σελίδα.
- <**getPopular()**: List<Game>>: Καλεί query για εύρεση των παιχνιδιών της βάσης που έχουν πολλές κριτικές πρόσφατα (>100 την τελευταία εβδομάδα).
- <**getByBacklogId**(backlogId: uuid): List<Game>>: Καλεί query για εύρεση όλων των παιχνιδιών που ανήκουν σε μία λίστα Backlog με βάση το όρισμα backlogId.
- <**advancedSearch**(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): List<Game>>: Καλεί query για την εύρεση των παιχνιδιών που αντιστοιχούν στα φίλτρα που δίνονται μέσω των βασικών κλειδιών των οντοτήτων Genre, Developer, Publisher και Platform και του τίτλου του παιχνιδιού.

<DeveloperDBProxy>

Μέθοδοι κλάσης

- <**create**(developer: Developer): void>: Καλεί query για δημιουργία ενός στούντιο ανάπτυξης στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος developer.
- <**update**(developer: Developer): void>: Καλεί query για ανανέωση ενός στούντιο ανάπτυξης στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος developer.
- <**delete**(developer: Developer): void>: Καλεί query για διαγραφή ενός στούντιο ανάπτυξης στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος developer.
- <**getById**(developerId: uuid): Developer>: Καλεί query για εύρεση ενός στούντιο ανάπτυξης στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <**getAll()**: List<Developer>>: Καλεί query για εύρεση όλων των στούντιο ανάπτυξης της βάσης.
- < **getByGame**(gameId: uuid): List<Developer>>: Καλεί query για εύρεση όλων των στούντιο ανάπτυξης που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.

<PublisherDBProxy>

Μέθοδοι κλάσης

- <**create**(publisher: Publisher): void>: Καλεί query για δημιουργία ενός εκδότη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος publisher.



- <`update(publisher: Publisher)`: Καλεί query για ανανέωση ενός εκδότη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος publisher.
- <`delete(publisher: Publisher)`: Καλεί query για διαγραφή ενός εκδότη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος publisher.
- <`getById(publisherId: uuid)`: Καλεί query για εύρεση ενός εκδότη στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <`getAll(): List<Publisher>`: Καλεί query για εύρεση όλων των εκδότων της βάσης.
- <`getByGame(gameId: uuid): List<Publisher>`: Καλεί query για εύρεση όλων των εκδότων που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.

<FranchiseDBProxy>

Μέθοδοι κλάσης

- <`create(franchise: Franchise)`: Καλεί query για δημιουργία μίας σειράς στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος franchise.
- <`update(franchise: Franchise)`: Καλεί query για ανανέωση μίας σειράς στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος franchise.
- <`delete(franchise: Franchise)`: Καλεί query για διαγραφή μίας σειράς στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος franchise.
- <`getById(franchiseId: uuid): Franchise`: Καλεί query για εύρεση μίας σειράς στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <`getAll(): List<Franchise>`: Καλεί query για εύρεση όλων των σειρών της βάσης.
- <`getByGame(gameId: uuid): List<Franchise>`: Καλεί query για εύρεση όλων των σειρών που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.

<GenreDBProxy>

Μέθοδοι κλάσης

- <`create(genre: Genre)`: Καλεί query για δημιουργία ενός είδους στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος genre.
- <`update(genre: Genre)`: Καλεί query για ανανέωση ενός είδους στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος genre.
- <`delete(genre: Genre)`: Καλεί query για διαγραφή ενός είδους στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος genre.
- <`getById(genreId: uuid): Genre`: Καλεί query για εύρεση ενός είδους στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <`getAll(): List<Genre>`: Καλεί query για εύρεση όλων των ειδών της βάσης.
- <`getByGame(gameId: uuid): List<Genre>`: Καλεί query για εύρεση όλων των ειδών που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.

<PlatformDBProxy>

Μέθοδοι κλάσης

- <`create(platform: Platform)`: Καλεί query για δημιουργία μιας πλατφόρμας στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος platform.
- <`update(platform: Platform)`: Καλεί query για ανανέωση μιας πλατφόρμας στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος platform.
- <`delete(platform: Platform)`: Καλεί query για διαγραφή μιας πλατφόρμας στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος platform.



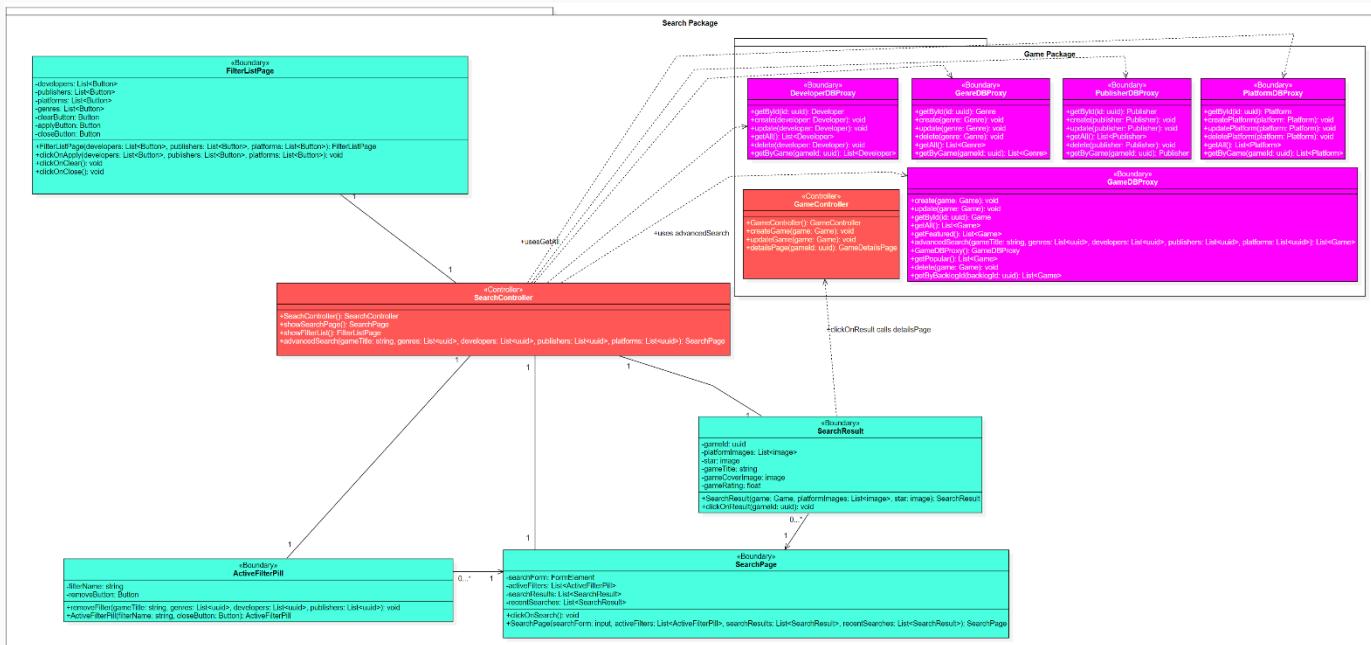
- <`getById(platformId: uuid)`: `Platform`>: Καλεί query για εύρεση μιας πλατφόρμας στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <`getAll()`: `List<Platform>`>: Καλεί query για εύρεση όλων των πλατφόρμων της βάσης.
- <`getByGame(gameId: uuid)`: `List<Platform>`>: Καλεί query για εύρεση όλων των πλατφόρμων που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.

<`ReviewDBPRoxy`>

Αναφέρεται στο [Review Package](#).



4.2 Search Package



Ελεγκτές

<SearchController>

Μέθοδοι κλάσης

- <SearchController(): SearchController>:**
Η συνάρτηση δόμησης της κλάσης.
- <showSearchpage(): SearchPage>:** Επιστρέφει το οριακό αντικείμενο της σελίδας αναζήτησης **SearchPage**.
- <showFilterList(): FilterListPage>:** Επιστρέφει το οριακό αντικείμενο της σελίδας της λίστας φίλτρων **FilterListPage**. Για τις επιλογές, χρησιμοποιούνται οι αντίστοιχες συναρτήσεις `getAll()` των **DeveloperDBProxy**, **FranchiseDBProxy**, **PlatformDBProxy**, **DeveloperDBProxy**, **GenreDBProxy**.
- <advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): SearchPage>:**
Επιστρέφει ανανεωμένη **SearchPage**, καλώντας την συνάρτηση `advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): List<Game>` του **GameDBProxy**. Αφού λάβει τα παιχνίδια σύμφωνα με τα φίλτρα που έχει ορίσει ο χρήστης, ανανεώνει την σελίδα με μία λίστα από το οριακό αντικείμενο **SearchResult** και εμφανίζει τα εμφαρμοσμένα φίλτρα ως μια λίστα από το οριακό αντικείμενο **ActiveFilterPill**. Αν ο χρήστης δεν έχει εμφαρμόσει φίλτρα, απλά αναζητά σύμφωνα με τον τίτλο που έδωσε ο χρήστης.

<GameController>

Αναφέρθηκε στο **Game Package**.



Οριακά Αντικείμενα

<FilterListPage>

Χαρακτηριστικά κλάσης

- <developers: List<Button>>: Λίστα από τα κουμπιά που αντιστοιχούν στα στούντιο ανάπτυξης **Developer**.
- <publishers: List<Button>>: Λίστα από τα κουμπιά που αντιστοιχούν στους εκδότες **Publisher**.
- <platforms: List<Button>>: Λίστα από τα κουμπιά που αντιστοιχούν στις πλατφόρμες **Platform**.
- <genres: List<Button>>: Λίστα από τα κουμπιά που αντιστοιχούν στα είδη **Genre**.
- <applyButton: Button>: Το κουμπί για την εφαρμογή των επιλεγμένων φίλτρων.
- <clearButton: Button>: Το κουμπί για την διαγραφή όλων των ενεργών φίλτρων.
- <closeButton: Button>: Το κουμπί για το κλείσιμο της σελίδας και την επιστροφή στην προηγούμενη.

Μέθοδοι κλάσης

- <FilterListPage(developers: List<Button>, publishers: List<Button>, platforms: List<Button>, genres: List<Button>, applyButton: Button, clearButton: Button, closeButton: Button)>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <clickOnApply(developers: List<Button>, publishers: List<Button>, platforms: List<Button>, genres: List<Button>, applyButton: Button): void>:
Καλεί την συνάρτηση **advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>)**: **SearchPage** του **SearchController**, προκειμένου να εφαρμοστούν τα επιλεγμένα φίλτρα του χρήστη στην αναζήτηση.
- <clickOnClear(): void>: Καλεί την συνάρτηση **advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>)**: **SearchPage** του **SearchController**, προκειμένου να αφαιρεθούν όλα τα ενεργά φίλτρα.
- <clickOnClose(): void>: Καλεί την συνάρτηση **advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>)**: **SearchPage** του **SearchController**, με τις προηγούμενες επιλογές του χρήστη.



<ActiveFilterPill>

Χαρακτηριστικά κλάσης

- <filterName: string>: Το όνομα του φίλτρου με βάση την οντότητα στην οποία αντιστοιχεί.
- <removeButton: Button>: Το κουμπί για να αφαιρεθεί το φίλτρο από τα ενεργά φίλτρα.

Μέθοδοι κλάσης

- <ActiveFilterPill(filterName: string, removeButton: Button): ActiveFilterPill>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <removeFilter(developers: List<Button>, publishers: List<Button>, platforms: List<Button>, genres: List<Button>): void>: Καλεί την συνάρτηση advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): SearchPage του SearchController, προκειμένου αφαιρεθεί το επιλεγόμενο φίλτρο από τα ενεργά φίλτρα ατης αναζήτησης.
- <clickOnClear(): void>: Καλεί την συνάρτηση advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): SearchPage του SearchController, προκειμένου να αφαιρεθούν όλα τα ενεργά φίλτρα.
- <clickOnClose(): void>: Καλεί την συνάρτηση advancedSearch(gameTitle: string, genres: List<uuid>, developers: List<uuid>, publishers: List<uuid>, platforms: List<uuid>): SearchPage του SearchController, με τις προηγούμενες επιλογές του χρήστη.

<SearchResult>

Χαρακτηριστικά κλάσης

- <gameId: uuid>: Το όνομα του φίλτρου με βάση την οντότητα στην οποία αντιστοιχεί.
- <gameTitle: string>: Το κουμπί για να αφαιρεθεί το φίλτρο από τα ενεργά φίλτρα.
- <gameCoverImage: image>: Η εικόνα αφίσα του παιχνιδιού.
- <gameRating: float>: Η γενική βαθμολογία του παιχνιδιού.
- <star: image>: Το εικονίδιο αστεριού για την βαθμολογία του παιχνιδιού.
- <platformImages: List<image>>: Η λίστα εικονιδίων για τις πλατφόρμες στις οποίες έχει κυκλοφορήσει το παιχνίδι.

Μέθοδοι κλάσης

- <SearchResult(gameId: uuid, gameTitle: string, gameCoverImage: image, gameRating: float, star: image, platformImages: List<image>): SearchResult>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.



- <`clickOnResult(gameId: uuid): void`>:
Καλεί την συνάρτηση `detailsPage(gameId: uuid): GameDetailsPage` του `GameController`.

<SearchPage>

Χαρακτηριστικά κλάσης

- <`searchForm: FormElement`>: Η φόρμα αναζήτησης
- <`activeFilters: List<ActiveFilterPill>`>: Η λίστα με τα ενεργά φίλτρα της αναζήτησης, ως οριακό αντικείμενο `ActiveFilterPill`.
- <`searchResults: List<SearchResult>`>: Η λίστα με τα αποτελέσματα της αναζήτησης, ως οριακό αντικείμενο `SearchResult`
- <`recentSearches: List<SearchResult>`>: Η λίστα με τις πρόσφατες αναζητήσεις του χρήστη που μεταφέρθηκες στην σελίδα `GameDetailsPage` τους.

Μέθοδοι κλάσης

- <`SearchPage(searchForm: FormElement, activeFilters: List<ActiveFilterPill>, searchResults: List<SearchResult>, recentSearches: List<SearchResult>): SearchPage`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`clickOnSearch(): void`>: Επιλέγει την φόρμα αναζήτησης, ώστε ο χρήστης να αρχίσει να πληκτρολογεί.

ΣΗΜΕΙΩΣΗ : Το αντικείμενο `FormElement` έχει τα χαρακτηριστικά των

- <`input: string`>: Το κείμενο που εισήγει ο χρήστης
- <`entityType: Enum`>: Η οντότητα με την οποία σχετίζεται (αν σχετίζεται)
- <`entityId: uuid`>: Το id της οντότητας με την οποία σχετίζεται (αν σχετίζεται)
- <`label: string`>: Η ετικέτα του

<GameDBProxy>

Αναφέρθηκε στο Game Package.

<DeveloperDBProxy>

Αναφέρθηκε στο Game Package.

<PublisherDBProxy>

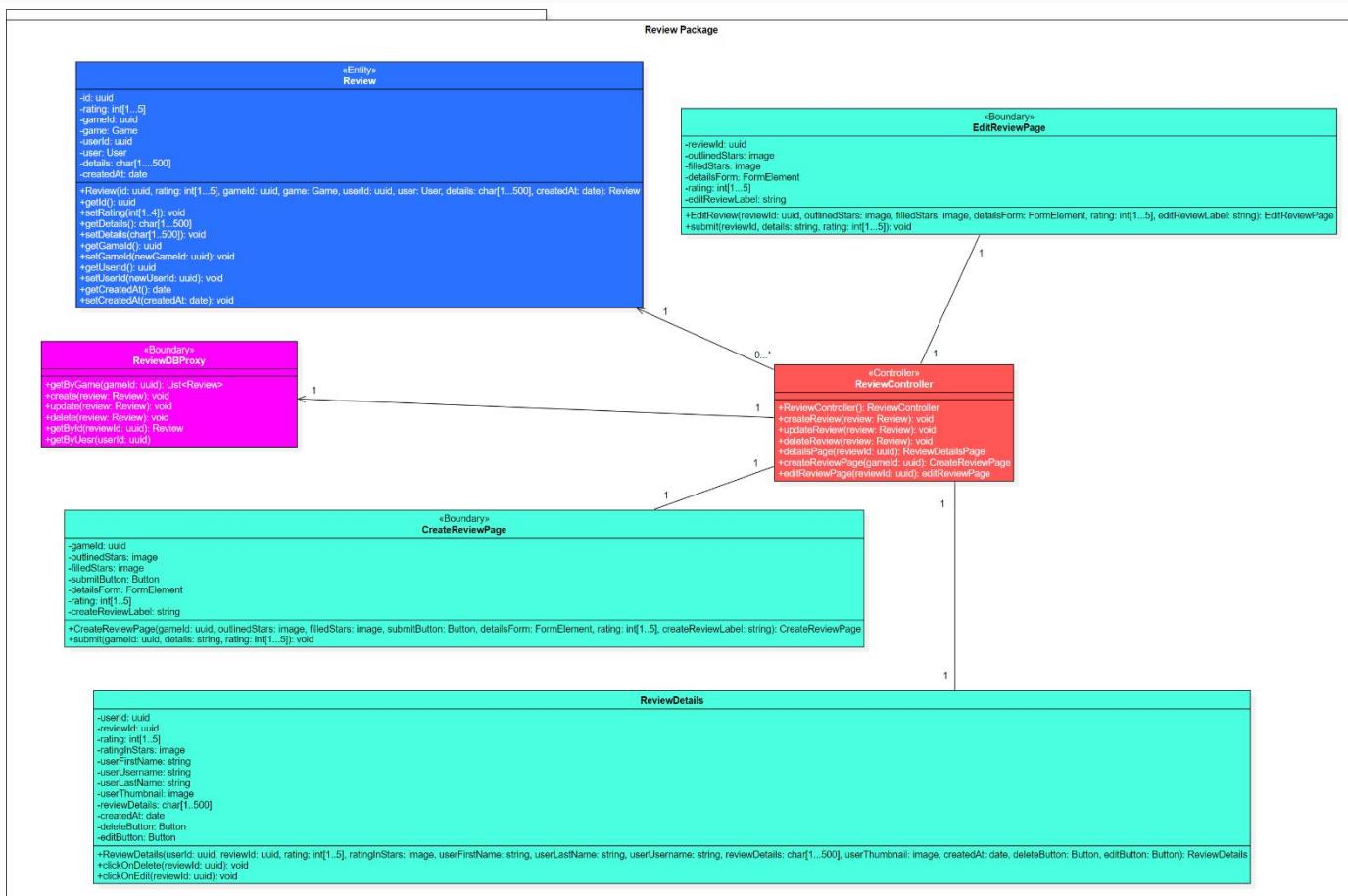
Αναφέρθηκε στο Game Package.

<PlatformDBProxy>

Αναφέρθηκε στο Game Package.



4.3 Review Package



Οντότητες

<Review>

Η κλάση της κριτικής ενός παιχνιδιού.

Χαρακτηριστικά κλάσης

- `<id: uuid>`: Το βασικό κλειδί και ευρετήριο της οντότητας.
- `<rating: integer[1..5]>`: Η βαθμολογία της κριτικής.
- `<gameId: uuid>`: Το ξένο κλειδί για την σύνδεση με την οντότητα του παιχνιδιού **Game**.
- `<game: Game>`: Το παιχνίδι **Game** για το οποίο γράφτηκε η κριτική.
- `<userId: uuid>`: Το ξένο κλειδί για την σύνδεση με την οντότητα του χρήστη **User**.
- `<user: User>`: Ο χρήστης **User** που έγραψε την κριτική.
- `<details: char[1..500]>`: Οι λεπτομέρειες της κριτικής.
- `<createdAt: date>`: Η ημερομηνία που γράφτηκε η κριτική.

Μέθοδοι Κλάσης

- `<Review(id: uuid, rating: int[1...5], gameId: uuid, game: Game, userId: uuid, user: User, details: char[1...500], createdAt: date): Review>`: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.



Ελεγκτές

<ReviewController>

Μέθοδοι κλάσης

- <`ReviewController(): ReviewController`>:
Η συνάρτηση δόμησης της κλάσης.
- <`detailsPage(reviewId: uuid): ReviewDetailsPage`>: Επιστρέφει την σελίδα με τις όλες λεπτομέρειες της κριτικής.
- <`createReview(review: Review): void`>: Προσθέτει μια καινούργια κριτική στη βάση δεδομένων, καλώντας την συνάρτηση `create(review: Review)` του `ReviewDBProxy`.
- <`updateReview (review: Review): void`>: Άλλάζει τα χαρακτηριστικά μιας υπάρχουσας κριτικής στη βάση δεδομένων, καλώντας την συνάρτηση `update(review: Review)` του `ReviewDBProxy`.
- <`deleteReview (review: Review): void`>: Διαγράφει μια κριτική από τη βάση δεδομένων, μέσω της συνάρτησης `delete(review: Review): void` του `ReviewDBProxy`.
- <`createReviewPage(gameId: uuid): CreateReviewPage`>: Επιστρέφει τη σελίδα `CreateReviewPage` που ο χρήστης μπορεί να προσθέσει μια κριτική για το παιχνίδι.
- <`editReviewPage(reviewId: uuid): EditReviewPage`>: Επιστρέφει τη σελίδα `EditReviewPage` που ο χρήστης μπορεί να επεξεργαστεί μια κριτική του.

Οριακά Αντικείμενα

<ReviewDBProxy>

Μέθοδοι κλάσης

- <`create(review: Review): void`>: Καλεί query για δημιουργία μιας κριτικής στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος `review`.
- <`update(review: Review): void`>: Καλεί query για ανανέωση μιας κριτικής στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος `review`.
- <`delete(review: Review): void`>: Καλεί query για διαγραφή μιας κριτικής στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος `review`.
- <`getById(reviewId: uuid): Review`>: Καλεί query για εύρεση μιας κριτικής στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.
- <`getByGame(gameId: uuid): List<Review>`>: Καλεί query για εύρεση όλων των κριτικών που αντιστοιχούν σε ένα παιχνίδι με το βασικό κλειδί του παιχνιδιού που δίνεται ως όρισμα.
- <`getByUser(userId: uuid): List<Review>`>: Καλεί query για εύρεση όλων των κριτικών που έχει κάνει ένας χρήστης

<CreateReviewPage>

Χαρακτηριστικά κλάσης

- <`gameId: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `Game`.



- <outlinedStars: `image`>: Η φωτογραφία με τα μη χρωματισμένα αστέρια της κριτικής.
- <filledStars: `Image`>: Η φωτογραφία με τα χρωματισμένα αστέρια της κριτικής.
- <submitButton: `Button`>: Το κουμπί που πατάει ο χρήστης για να υποβάλει την κριτική.
- <detailsForm: `FormElement`>: Η φόρμα στην οποία ο χρήστης γράφει την κριτική του.
- <rating: `integer[1..5]`>: Η βαθμολογία της κριτικής.
- <createReviewLabel: `string`>: Η ετικέτα της σελίδας.

Μέθοδοι κλάσης

- `CreateReviewPage(gameId: uuid, rating: int[1...5], outlinedStars: image, filledStars: image, submitButton: Button, detailsForm: FormElement, createReviewLabel: string)`: CreateReviewPage>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- `<submit(gameId: uuid, rating: int[1...5], details: string): void`: Υποβάλλει την κριτική με βάση το βασικό κλειδί του game, την βαθμολογία και τις λεπτομέρειες της κριτικής, καλώντας την συνάρτηση `createReview(review: Review): void` του `ReviewController`.

<EditReviewPage>

Χαρακτηριστικά κλάσης

- <reviewId: `uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `Review`.
- <outlinedStars: `image`>: Η φωτογραφία με τα μη χρωματισμένα αστέρια της κριτικής.
- <filledStars: `Image`>: Η φωτογραφία με τα χρωματισμένα αστέρια της κριτικής.
- <detailsForm: `FormElement`>: Η φόρμα στην οποία ο χρήστης επεξεργάζεται την κριτική του.
- <rating: `integer[1..5]`>: Η βαθμολογία της κριτικής.
- <editReviewLabel: `string`>: Η ετικέτα της σελίδας.

Μέθοδοι κλάσης

- <`EditReview(id: uuid, rating: int[1...5], outlinedStars: image, filledStars: image, detailsForm: FormElement, editReviewLabel: string)`: EditReviewPage>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`<submit(reviewId: uuid, rating: int[1...5], details: string): void`: Υποβάλλει την επεξεργασμένη κριτική με βάση το βασικό κλειδί της κριτικής `Review`, την βαθμολογία και τις λεπτομέρειες της κριτικής, καλώντας την συνάρτηση `updateReview (review: Review): void` του `ReviewController`.

<ReviewDetails>

Χαρακτηριστικά κλάσης

- <reviewId: `uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `Review`.
- <userId: `uuid`>: Το βασικό κλειδί της οντότητας `User`, χρησιμοποιείται για να μεταφερθεί ο χρήστης στην σελίδα προφίλ του χρήστη.



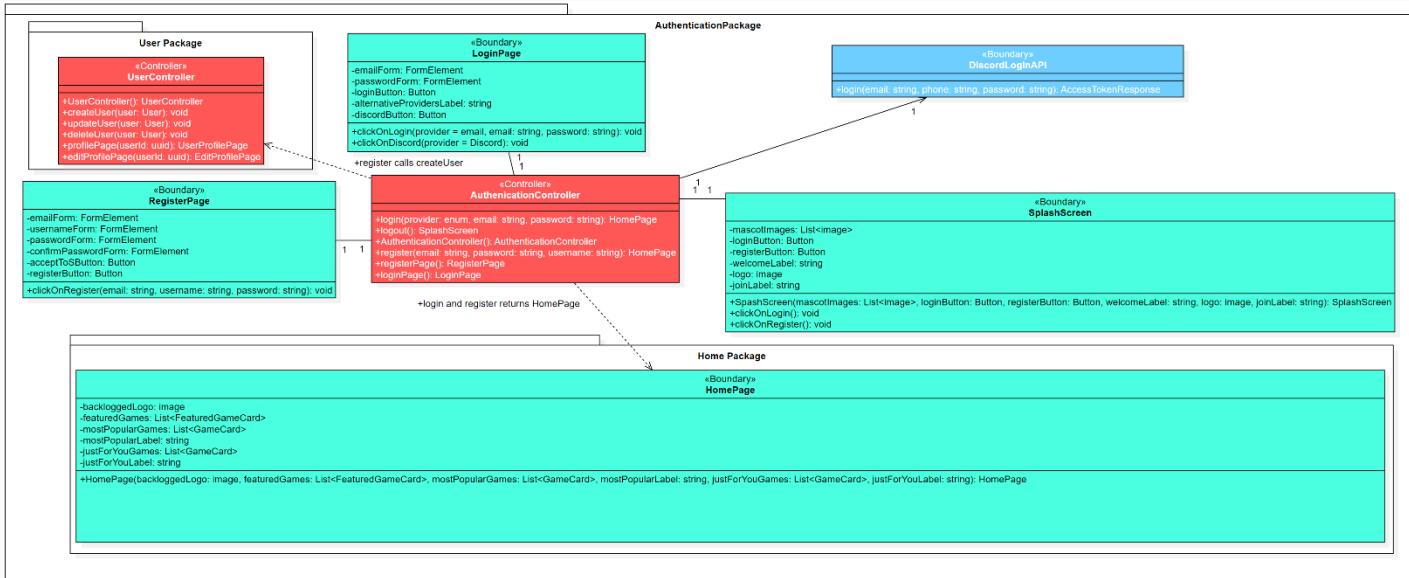
- `<rating: int[1..5]>`: Η βαθμολογία της κριτικής.
- `<ratingInStars: image>`: Η αντίστοιχη βαθμολογία σε εικόνα με αστέρια.
- `<userFirstName: string>`: Το όνομα του χρήστη.
- `<userLastName: string>`: Το επώνυμο του χρήστη.
- `<userUsername: string>`: Το username του χρήστη.
- `<userThumbnail: image>`: Η φωτογραφία προφίλ του χρήστη.
- `<reviewDetails: char[1..500]>`: Οι λεπτομέρειες της κριτικής.
- `<createdAt: date>`: Η ημερομηνία που γράφτηκε η κριτική.
- `<deleteButton: Button>`: Το κουμπί για την διαγραφή της κριτικής.
- `<editButton: Button>`: Το κουμπί για την επεξεργασία της κριτικής.

Μέθοδοι κλάσης

- `<ReviewCard(reviewId: uuid, userId: uuid, rating: int[1..5], ratingInStars: image, userFirstName: string, userLastName: string, userThumbnail: image, reviewDetails: char[1..500], closeButton: Button, createdAt: date)>`: ReviewDetails:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- `<clickonDelete(reviewId: void): void>`: Καλεί την συνάρτηση `deleteReview` (`review: Review`): `void` του `ReviewController`, αφού ο χρήστης επιβεβαιώσει την επιλογή του.
- `<clickOnEdit(reviewId: void): void>`: Καλεί την συνάρτηση `editReviewPage` (`reviewId: void`): `ReviewEditPage` του `ReviewController` για αυτή την κριτική.



4.4 Authentication Package



Ελεγκτές

<AuthenticationController>

Μέθοδοι κλάσης

- <**AuthenticationController()**: **AuthenticationController**>: Η συνάρτηση δόμησης της κλάσης.
- <**logout()**: **SplashScreen**>: Αποσυνδέει τον χρήστη και επιστρέφει το οριακό αντικείμενο **SplashScreen**.
- <**login(provider: enum, email: string, password: string)**: **HomePage**>: Επιστρέφει το οριακό αντικείμενο της αρχικής σελίδας **HomePage**, αφού επιβεβαιώσει τα στοιχεία του χρήστη. Το όρισμα provider ξεχωρίζει αν γίνεται μέσω email ή μέσω discord. Αν γίνεται μέσω discord, καλείται η συνάρτηση **login(email: string, phone: string, password: string)**: **AccessTokenResponse**.
- <**register(email: string, password: string, username: string)**: **void**>: Προσθέτει έναν καινούργιο χρήστη στη βάση δεδομένων, καλώντας την συνάρτηση **create(user: User)** του **UserController**.
- <**registerPage()**: **RegisterPage**>: Επιστρέφει το οριακό αντικείμενο της σελίδας εγγραφής **RegisterPage**.
- <**loginPage()**: **LoginPage**>: Επιστρέφει το οριακό αντικείμενο της σελίδας σύνδεσης **LoginPage**.

<UserController>

Αναφέρεται στο **User Package**.

Οριακά Αντικείμενα

<SplashScreen>

Χαρακτηριστικά κλάσης

- <**mascotImages: List<image>**>: Τα εικονίδια μασκότ της εφαρμογής.
- <**loginButton: Button**>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στη σελίδα σύνδεσης.



- <registerButton: `Button`>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στη σελίδα εγγραφής.
- <welcomeLabel: `string`>: Το κείμενο που καλωσορίζει τον χρήστη.
- <logo: `image`>: Το λογότυπο της σελίδας
- <joinLabel: `string`>: Το κείμενο που πληροφορεί τον χρήστη για την σελίδα.

Μέθοδοι κλάσης

- <`SplashScreen`(mascotImages: `List<image>`, loginButton: `Button`, registerButton: `Button`, welcomeLabel: `string`, logo: `image`, joinLabel: `string`): `SplashScreen`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`clickOnLogin(): void`>: Καλεί την συνάρτηση `loginPage()`: `LoginPage` του `AuthenticationController`.
- <`clickOnRegister(): void`>: Καλεί την συνάρτηση `registerPage()`: `RegisterPage` του `AuthenticationController`.

<LoginPage>

Χαρακτηριστικά κλάσης

- <`emailForm: FormElement`>: Η φόρμα για να γράψει ο χρήστης το email του.
- <`passwordForm: FormElement`>: Η φόρμα για να γράψει ο χρήστης τον κωδικό του.
- <`loginButton: Button`>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στη σελίδα σύνδεσης.
- <`alternativeProvidersLabel: string`>: Το κείμενο που ενημερώνει τον χρήστη αν θέλει να συνδεθεί μέσω διαφορετικού provider.
- <`discordButton: Button`>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στη σελίδα σύνδεσης μέσω discord.

Μέθοδοι κλάσης

- <`clickOnLogin(provider: email, email: string, password: string): void`>: Καλεί την συνάρτηση `login(provider = email, email: string, password: string)` του `AuthenticationController`.
- <`clickOnDiscord(provider= discord): void`>: Καλεί την συνάρτηση `login(provider = discord, email = null, password = null)` του `AuthenticationController`.



<RegisterPage>

Χαρακτηριστικά κλάσης

- <emailForm: **FormElement**>: Είναι η φόρμα για να γράψει ο χρήστης το email του.
- <usernameForm: **FormElement**>: Είναι η φόρμα για να γράψει ο χρήστης το username του.
- <passwordForm: **FormElement**>: Είναι η φόρμα για να γράψει ο χρήστης τον κωδικό του.
- <confirmPasswordForm: **FormElement**>: Είναι η φόρμα για να ξανάγράψει ο χρήστης τον κωδικό του για επιβεβαίωση.
- <acceptToSButton: **Button**>: Το κουμπί που πατάει ο χρήστης για να αποδεχτεί τους όρους χρήσης.
- <registerButton: **Button**>: Το κουμπί που πατάει ο χρήστης για να εγγραφεί.

Μέθοδοι κλάσης

- <**clickOnRegister**(email: **string**, username: **string**, password: **string**): **void**>:
Καλεί την συνάρτηση **register**(email: **string**, password: **string**, username: **string**) του **AuthenticationController**.

<HomePage>

Αναφέρεται στο **Home Package**.

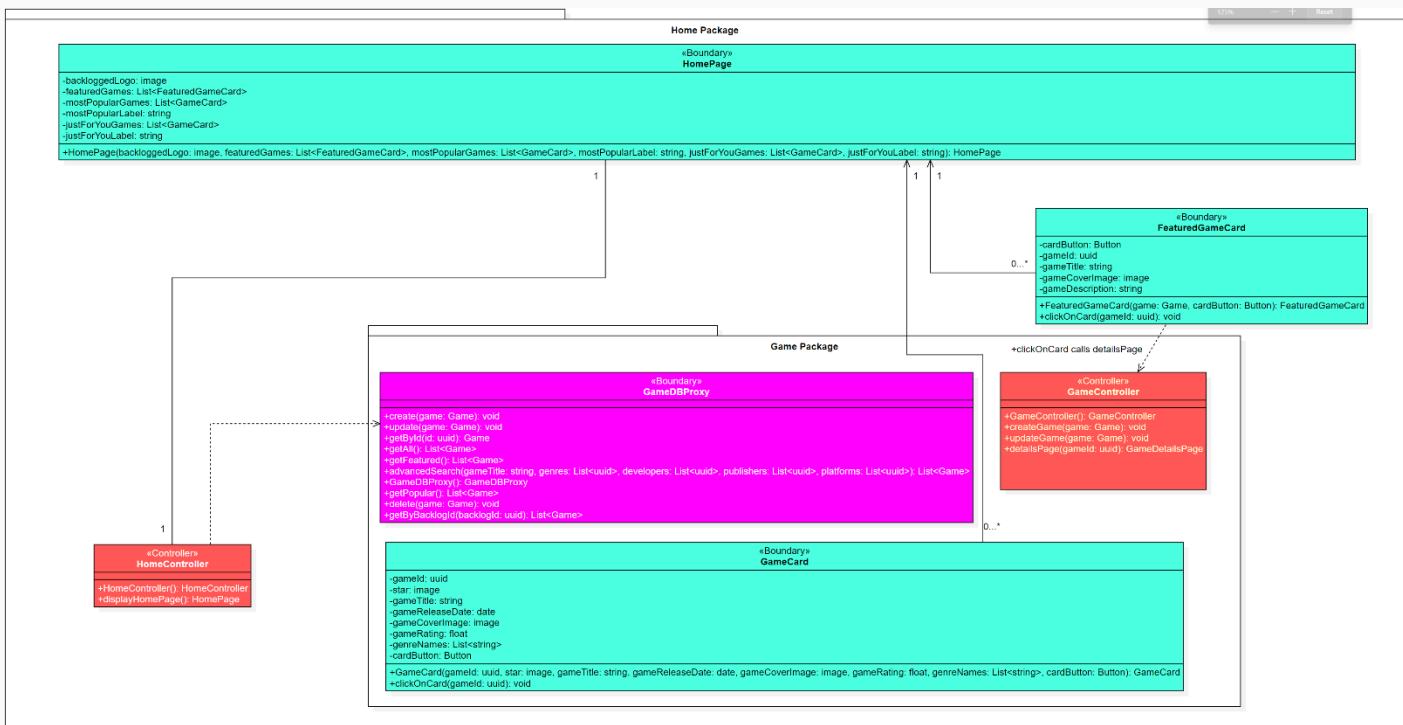
<DiscordLoginAPI>

Μέθοδοι κλάσης

- <**login**(email: **string**, phone: **string**, password: **string**):
AccessTokenResponse>: Κάνει επιχείρηση ταυτοποίησης των στοιχείων του χρήστη μέσω των στοιχείων του Discord. Επιστρέφει ένα αντίκειμενο **AccessTokenReponse** μέσω JSON πίσω στην εφαρμογή.



4.5 Home Package



Ελεγκτές

<HomeController>

Μέθοδοι κλάσης

- `<HomeController(): HomeController>`: Η συνάρτηση δόμησης της κλάσης.
- `<displayHomePage(): HomePage>`: Επιστρέφει το οριακό αντικείμενο της αρχικής σελίδας **HomePage**. Χρησιμοποιεί το **GameDBProxy** για να βρει τα διαφημιζόμενα παιχνίδια μέσω της `getFeatured(): List<Game>`, τα δημοφιλή παιχνίδια μέσω της `getPopular(): List<Game>` και παιχνίδια που ταιριάζουν στο χρήστη μέσω της `getAll(): List<Game>` και μετά εφαρμόζοντας έναν αλγόριθμο για να ελέγξει ποια ταιριάζουν στον χρήστη. Για τα διαφημιζόμενα παιχνίδια δημιουργεί μια λίστα από **FeaturedGameCard** και για τα υπόλοιπα δημιουργεί τις λίστες από **GameCard**.

<GameController>

Αναφέρθηκε στο **Game Package**.

Οριακά Αντικείμενα

<HomePage>

Χαρακτηριστικά κλάσης

- `<backloggedLogo: image>`: Η φωτογραφία λογότυπο της σελίδας.
- `<featuredGames: List<FeaturedGameCard>>`: Η λίστα με τα διαφημιζόμενα παιχνίδια.
- `<mostPopularGames: List<GameCard>>`: Η λίστα με τα πιο δημοφιλή παιχνίδια της εβδομάδας.



- <mostPopularLabel: string>: Η ετικέτα των δημοφιλών παιχνδιών.
- <justForYouGames: List<GameCard>>: Παιχνίδια που ταιριάζουν στον χρήστη σύμφωνα με τη λίστα **Backlog** του και τις κριτικές του.
- <justForYouLabel: string>: Η ετικέτα για τα παιχνίδια που ταιριάζουν στο χρήστη.

Μέθοδοι κλάσης

- <HomePage(backloggedLogo: image, featuredGames: List<FeaturedGameCard>, mostPopularGames: List<GameCard>, mostPopularLabel: string, newReviewsFromFriends: List<ReviewCardHome>, newFromFriendsLabel: string, justForYouGames: List<GameCard>, justForYouLabel: string): HomePage>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

<FeaturedGameCard>

Χαρακτηριστικά κλάσης

- <cardButton: Button>: Το κουμπί που πατάει ο χρήστης για να μεταφερθεί στην σελίδα **GameDetailsPage** του παιχνιδιού.
- <gameId: uuid>: Το βασικό κλειδί της οντότητας **Game**, χρησιμοποιείται για να μεταφερθεί ο χρήστης στην σελίδα του παιχνιδιού.
- <gameTitle: string>: Ο τίτλος του παιχνιδιού.
- <gameCoverImage: image>: Η φωτογραφία εξωφύλλου του παιχνιδιού.
- <gameDescription: string>: Η περιγραφή του παιχνιδιού.

Μέθοδοι κλάσης

- <FeaturedGameCard(game: Game, cardButton: Button): FeaturedGameCard>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <clickOnCard(gameId: uuid): void>: Καλεί την συνάρτηση την **detailsPage**(gameId: **uuid**) του **GameController**.

<GameCard>

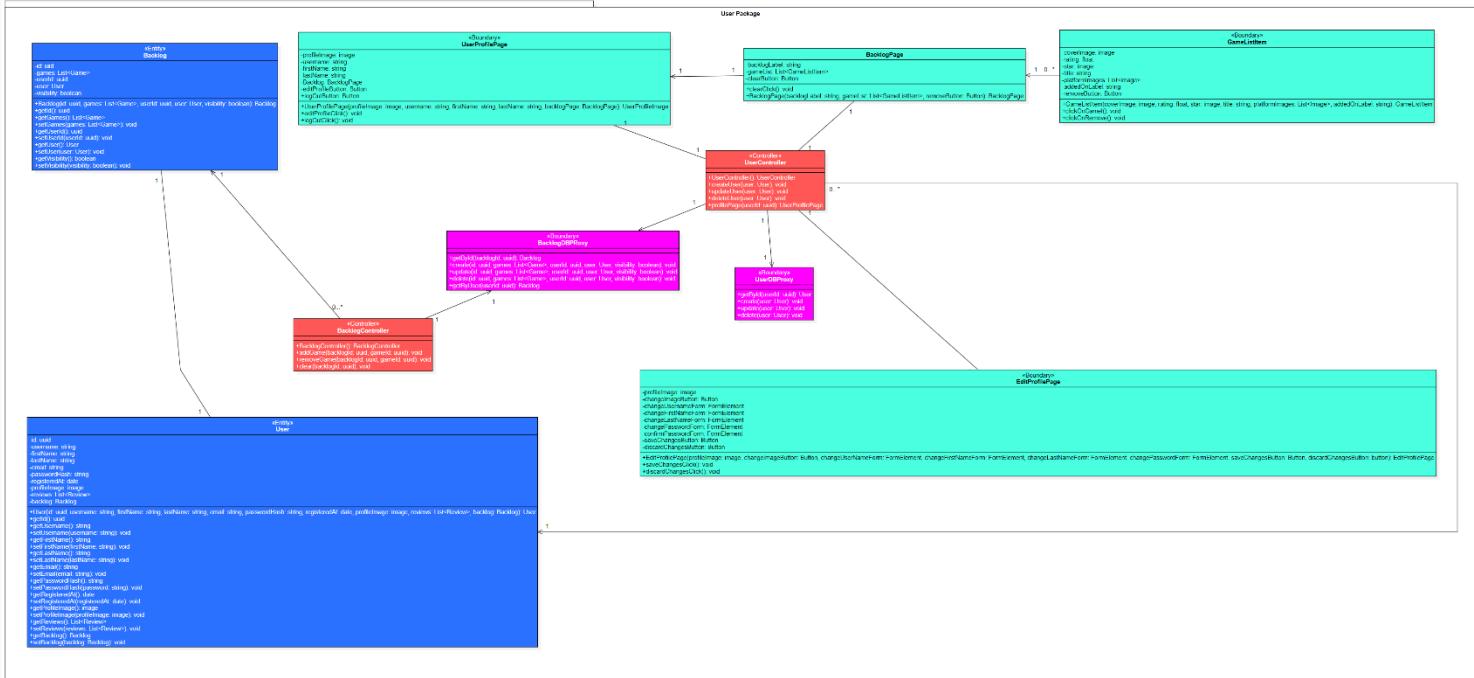
Αναφέρθηκε στο **Game Package**.

<GameDBProxy>

Αναφέρθηκε στο **Game Package**.



4.6 User Package



Οντότητες

<User>

Η κλάση του χρήστη.

Χαρακτηριστικά κλάσης

- **<id: `uuid`>**: Το βασικό κλειδί και ευρετήριο της οντότητας.
- **<username: `string`>**: Το username του χρήστη.
- **<firstName: `string`>**: Το όνομα του χρήστη.
- **<lastName: `string`>**: Το επώνυμο του χρήστη.
- **<email: `string`>**: Το email του χρήστη.
- **<passwordHash: `string`>**: Ο κωδικός του χρήστη.
- **<profileImage: `image`>**: Η φωτογραφία προφίλ του χρήστη.
- **<registeredAt: `date`>**: Η ημερομηνία εγγραφής του χρήστη.
- **<reviews: `List<Review>`>**: Η λίστα των **Review** που έχει κάνει ο χρήστης.
- **<backlog: `Backlog`>**: To **Backlog**.

Μέθοδοι κλάσης

- **<User(id: `uuid`, username: `string`, firstName: `string`, lastName: `string`, email: `string`, passwordHash: `string`, profileImage: `image`, registeredAt: `date`, reviews: `List<Review>`, backlog: `Backlog`)>**: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.



<Backlog>

Η κλάση της λίστας Backlog του χρήστη.

Χαρακτηριστικά κλάσης

- <`id: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `Backlog`.
- <`userId: uuid`>: Το βασικό κλειδί και ευρετήριο της οντότητας `User`.
- <`games: List<Game>`>: Η λίστα των `Game` που είναι στο `Backlog`.
- <`visibility: boolean`>: Αν το backlog είναι ορατό προς όλους.
- <`user: User`>: Ο χρήστης `User`.

Μέθοδοι κλάσης

- <`Backlog(id: uuid, userId: uuid, games: List<Game>, visibility: boolean, user: User)`>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

Ελεγκτές

<UserController>

Μέθοδοι κλάσης

- <`UserController(): UserController`>: Η συνάρτηση δόμησης της κλάσης.
- <`createUser(user: User): void`>: Προσθέτει έναν καινούριο χρήστη στη βάση δεδομένων, καλώντας την συνάρτηση `create(user: User)` του `UserDBProxy`.
- <`updateUser(user: User): void`>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος χρήστη στη βάση δεδομένων, καλώντας την συνάρτηση `update(user: User)` του `UserDBProxy`, αφού επιβεβαιώσει τα χαρακτηριστικά του, εφ' όσον έχει αλλάξει το username ή το password του.
- <`deleteUser (user: User): void`>: Διαγράφει έναν χρήστη από τη βάση δεδομένων, μέσω της συνάρτησης `delete(user: User): void` του `UserDBProxy`.
- <`profilePage(userId: uuid): UserProfilePage`>: Επιστρέφει την σελίδα του προσωπικού προφίλ του χρήστη `UserProfilePage`, καλώντας τις συναρτήσεις `getByUser(userId: uuid): Backlog` του `BacklogDBProxy` και `getById(userId: uuid): User` του `UserDBProxy`, και έπειτα αναθέτωντας τις πληροφορίες των οντοτήτων στα οριακά αντικείμενα `BacklogPage` και `UserProfilePage` αντίστοιχα. Η πρόσφατη δραστηριότητα είναι οι τελευταίες κριτικές που έκανε ο χρήστης μέσω της `getByUser(userId: uuid): List<Review>` του `ReviewDBProxy` και ταξινομώντας σύμφωνα με τα πιο πρόσφατα και δημιουργώντας καινούργια αντικείμενα `GameListItem` για αυτά.
- <`editProfilePage(userId: uuid): EditProfilePage`>: Επιστρέφει την σελίδα του της επεξεργασίας των στοιχείων του χρήστη `EditProfilePage`.

<BacklogController>

Μέθοδοι κλάσης

- <`BacklogController(): BacklogController`>: Η συνάρτηση δόμησης της κλάσης.



- <`addGame(backlogId: uuid, gameId: uuid): void`>: Προσθέτει ένα καινούριο παιχνίδι στο `Backlog`, καλώντας την συνάρτηση `update(id: uuid, userId: uuid, games: List<Game, visibility: boolean, user: User>)` του `BacklogDBProxy`.
- <`removeGame(backlogId: uuid, gameId: uuid): void`>: Διαγράφει έναν παιχνίδι από το `Backlog`, καλώντας την συνάρτηση `update(id: uuid, userId: uuid, games: List<Game, visibility: boolean, user: User>)` του `BacklogDBProxy`.
- <`clear(backlogId: uuid): void`>: Διαγράφει όλα τα παιχνίδια από το `Backlog` καλώντας την συνάρτηση `update(id: uuid, userId: uuid, games: List<Game, visibility: boolean, user: User>)` του `BacklogDBProxy`.

Οριακά Αντικείμενα

<UserDBProxy>

Μέθοδοι κλάσης

- <`create(user: User): void`>: Καλεί query για δημιουργία ενός χρήστη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου user που δίνεται ως όρισμα.
- <`update(user: User): void`>: Καλεί query για ανανέωση ενός χρήστη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου user που δίνεται ως όρισμα.
- <`delete(user: User): void`>: Καλεί query για διαγραφή ενός χρήστη στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου user που δίνεται ως όρισμα.
- <`getById(userId: uuid): User`>: Καλεί query για εύρεση ενός χρήστη στη βάση δεδομένων με το βασικό κλειδί που δίνεται ως όρισμα.

<BacklogDBProxy>

Μέθοδοι κλάσης

- <`create(backlog: Backlog): void`>: Καλεί query για δημιουργία ενός Backlog στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου backlog που δίνεται ως όρισμα.
- <`update(id: uuid, userId: uuid, games: List<Game, visibility: boolean, user: User>): void`>: Καλεί query για ανανέωση ενός Backlog στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου backlog που δίνεται ως όρισμα.
- <`delete(id: uuid, userId: uuid, games: List<Game, visibility: boolean, user: User>): void`>: Καλεί query για διαγραφή ενός Backlog στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου backlog που δίνεται ως όρισμα.
- <`getById(backlogId: uuid): Backlog`>: Καλεί query για εύρεση ενός Backlog με τα παιχνίδια του στη βάση δεδομένων με το βασικό κλειδί του Backlog που δίνεται ως όρισμα.
- < `getByUser(userId: uuid): Backlog`>: Καλεί query για εύρεση ενός Backlog με τα παιχνίδια του στη βάση δεδομένων με το βασικό κλειδί του User που δίνεται ως όρισμα.

<UserProfilePage>

Χαρακτηριστικά κλάσης

- <`profileImage: image`>: Η φωτογραφία προφίλ του χρήστη.
- <`username: string`>: To username του χρήστη.



- <firstName: string>: Το όνομα του χρήστη.
- <lastName: string>: Το επώνυμο του χρήστη.
- <backlog: BacklogPage>: Το αντικείμενο **BacklogPage** του χρήστη.
- <editProfileButton: Button>: Το κουμπί που πατάει ο χρήστης για να κάνει επεξεργασία το προφίλ του.
- <logOutButton: Button>: Το κουμπί που πατάει ο χρήστης για να αποσυνδεθεί.

Μέθοδοι κλάσης

- <UserProfilePage(profileImage: image>, username: string, firstName: string, lastName: string, Backlog: BacklogPage, editProfileButton: Button, logOutButton: Button): UserProfileImage>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <clickOnEditProfile(): void>: Καλεί την συνάρτηση **editProfilePage(userId: uuid): EditProfilePage** του **UserController**.
- <logOutClick(): void>: Καλεί την συνάρτηση **logout(): SplashScreen** του **AuthenticationController**.

<EditProfilePage>

Χαρακτηριστικά κλάσης

- <profileImage: image>: Η φωτογραφία προφίλ του χρήστη.
- <changeImageButton: Button>: Το κουμπί που πατάει ο χρήστης για να αλλάξει την φωτογραφία προφίλ.
- <changeUsernameForm: FormElement>: Η φόρμα που χρησιμοποιεί ο χρήστης για να αλλάξει το username του.
- <changeFirstNameForm: FormElement>: Η φόρμα που χρησιμοποιεί ο χρήστης για να αλλάξει το όνομά του.
- <changeLastNameForm: FormElement>: Η φόρμα που χρησιμοποιεί ο χρήστης για να αλλάξει το επώνυμό του.
- <changePasswordForm: FormElement>: Η φόρμα που χρησιμοποιεί ο χρήστης για να αλλάξει το password του.
- <confirmPasswordForm: FormElement>: Η φόρμα που χρησιμοποιεί ο χρήστης για να επιβεβαιώσει το νέο του password.
- <saveChangesButton: Button>: Το κουμπί που πατάει ο χρήστης για να αποθηκεύσει τις αλλαγές στο προφίλ του.
- <discardChangesButton: Button>: Το κουμπί που πατάει ο χρήστης για να ακυρώσει τις αλλαγές στο προφίλ του.

Μέθοδοι κλάσης

- <EditProfilePage(profileImage: image>, changeImageButton: Button, changeUsernameForm: FormElement, changeFirstNameForm: FormElement, changeLastNameForm: FormElement, changePasswordForm: FormElement, confirmPasswordForm: FormElement, saveChangesButton: Button, discardChangesButton: Button): EditProfilePage>:



Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.

- <`saveChangesClick(): void`>: Καλεί την συνάρτηση `updateUser(user: User): void` του `UserController`.
- <`discardChangesClick(): void`>: Καλεί την συνάρτηση `profilePage(userId: uuid): ProfilePage` του `UserController`.

<GameListItem>

Μέρος της λίστας της πρόσφατης δραστηριότητας του χρήστη ή του Backlog του.

Χαρακτηριστικά κλάσης

- <`coverImage: image`>: Η φωτογραφία αφίσα του παιχνιδιού.
- <`rating: float`>: Ο μέσος όρος βαθμολογήσεων του παιχνιδιού από τις κριτικές του.
- <`star: image`>: Η φωτογραφία με τα αστέρια της κριτικής.
- <`title: string`>: Ο επίσημος τίτλος του παιχνιδιού στην αγορά.
- <`platformImages: List<image>`>: Η λίστα με τις φωτογραφίες των πλατφόρμων `Platform` στις οποίες έχει κυκλοφορήσει το παιχνίδι.
- <`addedOnLabel: string`>: Η ετικέτα που αναφέρει πότε ο χρήστης έγραψε την κριτική ή προσέθεσε το παιχνίδι στο backlog του
- <`removeButton: Button`>: Το κουμπί που πατάει ο χρήστης για να αφαιρέσει ένα παιχνίδι από το Backlog του.

Μέθοδοι κλάσης

- <`GameListItem(coverImage: image, rating: float, star: image, title: string, platformImages: List<image>, addedOnLabel: string, removeButton: Button): GameListItem`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`clickOnGame(gameId: uuid): void`>: Καλεί την συνάρτηση `detailsPage(gameId: uuid): GameDetailsPage` του `GameController`.
- <`clickOnRemove(gameId: uuid): void`>: Καλεί την συνάρτηση `removeGame(backlogId: uuid, gameId: uuid): void` του `BacklogController`. Το `backlogId` είναι γνωστό από το `BacklogPage`.

<BacklogPage>

Χαρακτηριστικά κλάσης

- <`backlogLabel: string`>: Η ετικέτα του backlog.
- <`gameList: List<GameListItem>`>: Η λίστα με τα παιχνίδια που έχει λάβει από την `getByUserId(userId: uuid): Backlog` του `UserController` ως οριακά αντικείμενα `GameList`.
- <`clearButton: Button`>: Το κουμπί που πατάει ο χρήστης για να καθαρίσει την σελίδα του Backlog.



Μέθοδοι κλάσης

- <`BacklogPage`(backlogLabel: `string`, `List<GameListItem>`, `clearButton: Button`): `BacklogPage`>:
Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <`clearClick(): void`>: Καλεί την συνάρτηση `clear(backlogId: uuid): void`. Το `backlogId` είναι γνωστό από το `BacklogPage`.



Παράρτημα I – Γλωσσάριο

Το σετ των ακρωνυμίων που χρησιμοποιείτε στο έγγραφο

ΛΑ-xxxx	Λειτουργική Απαίτηση xxxx
ΜΛΑ-xxxx	Μη Λειτουργική Απαίτηση xxxx
ΟΑ	Ομάδα Εργασίας