



# *PartyUp*

Find the best party around

## Απαιτήσεις Χρηστών και Προδιαγραφές Συστήματος

Version 1.8

Βασίλειος Ευαγγελόπουλος [vasileie@ece.auth.gr](mailto:vasileie@ece.auth.gr)  
Σταύρος Βασίλειος Μπουλιόπουλος [smpoulis@ece.auth.gr](mailto:smpoulis@ece.auth.gr)  
Νικόλαος Γιαννόπουλος [ngiannop@ece.auth.gr](mailto:ngiannop@ece.auth.gr)  
Εμμανουήλ Κωτούλας [emmakoto@ece.auth.gr](mailto:emmakoto@ece.auth.gr)

21/4/2022



## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	<a href="mailto:asymeon@issel.ee.auth.gr">asymeon@issel.ee.auth.gr</a>
Βασίλειος Ευαγγελόπουλος	25	vasileie@ece.auth.gr
Σταύρος Βασίλειος Μπουλιόπουλος	25	smpoulio@ece.auth.gr
Νικόλαος Γιαννόπουλος	25	ngiannop@ece.auth.gr
Εμμανουήλ Κωτούλας	25	emmakoto@ece.auth.gr

### Πίνακας Περιεχομένων

### Πίνακας περιεχομένων

Πίνακας Περιεχομένων .....	2
1 Απαιτήσεις Συστήματος .....	5
1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης) .....	5
1.2 Χρήστες και εξωτερικά συστήματα.....	8
1.3 Σημαντικές μη λειτουργικές απαιτήσεις.....	8
1.3.1 Απαιτήσεις επίδοσης .....	8
1.3.2 Απαιτήσεις ασφάλειας (Security) .....	9
1.3.3 Απαιτήσεις χρηστικότητας (Usability).....	9
1.3.4 Απαιτήσεις Φορητότητας (Portability) .....	10
1.3.5 Απαιτήσεις πολιτικής (Policy - legal, corporate) .....	10
1.10 Λεξιλογικοί Προσδιορισμοί.....	10
2 Σενάρια Χρήσης.....	12
2.1 Διάγραμμα σεναρίων χρήσης.....	12
2.2 <Feature 1: Add a Friend> (Feature με βάση την ορολογία Gherkin).....	12
2.3 <Feature 2: View active parties> (Feature με βάση την ορολογία Gherkin) .....	13
2.4 <Feature 3: Filter active parties> (Feature με βάση την ορολογία Gherkin).....	16
2.5 <Feature 4: Make a donation> (Feature με βάση την ορολογία Gherkin) .....	17
2.6 <Feature 5: Host a party> (Feature με βάση την ορολογία Gherkin) .....	19
2.7 <Feature 6: Review a party> (Feature με βάση την ορολογία Gherkin).....	20
2.8 <Feature 7: Email user> (Feature με βάση την ορολογία Gherkin) .....	21



---

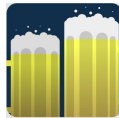
2.9 <Feature 8: Search users> (Feature με βάση την ορολογία Gherkin) .....	22
2.10 <Feature 9: View participants> (Feature με βάση την ορολογία Gherkin) .....	23
2.11 <Feature 10: View user's reviews> (Feature με βάση την ορολογία Gherkin).....	24
2.12 <Feature 11: Join Party > (Feature με βάση την ορολογία Gherkin).....	25
2.13 <Feature 12: Leave from a party> (Feature με βάση την ορολογία Gherkin).....	27
2.14 <Feature 13: View recently joined parties> (Feature με βάση την ορολογία Gherkin) .....	28
3 Επιδεικτικά γραφικά παράθυρα διεπαφής .....	30
3.1 Σελίδα Σύνδεσης/Εγγραφής και ανάκτηση κωδικού .....	30
3.2 Σελίδα αρχική οθόνης .....	31
3.3 Σελίδα διοργανωτή .....	34
3.4 Σελίδα ιστορικού .....	35
3.5 Σελίδα Φίλων .....	36
4 Στατική Μοντελοποίηση.....	37
4.1 <Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας> .....	37
4.1.1 Πακέτο Party and User.....	37
4.1.3 Πακέτο Principal Component .....	43
4.1.4 Πακέτο Home Page .....	49
4.1.5 Πακέτο HostPage .....	53
4.1.6 Πακέτο PartyPage.....	56
4.2 <Πακέτο(α) λεξιλογίου σεναρίων μέσης προτεραιότητας> .....	60
4.2.1 Πακέτο FriendsPage .....	60
4.3 <Πακέτο(α) λεξιλογίου σεναρίων χαμηλής προτεραιότητας >.....	63
4.3.1 Πακετο History Page .....	63
4.3.1 Ορισμού - Πακέτο HistoryPage .....	63
Παράρτημα Ι - Γλωσσάριο.....	67
Παράρτημα ΙΙ - Ανοιχτά Θέματα.....	68



---

## Λίστα Σχημάτων

Εικόνα 2. 1 Use cases diagram .....	12
Εικόνα 2. 2 Activity Diagram - Add Friend.....	13
Εικόνα 2. 3 Activity Diagram - View Active Parties.....	15
Εικόνα 2. 4 Activity Diagram - Filter Active Parties.....	17
Εικόνα 2. 5 Activity Diagram - Donate .....	18
Εικόνα 2. 6 Activity Diagram - Host a Party .....	20
Εικόνα 2. 7 Activity Diagram - Review Party .....	21
Εικόνα 2. 8 Activity Diagram - Email User .....	22
Εικόνα 2. 9 Activity Diagram - Search Users .....	23
Εικόνα 2. 10 Activity Diagram - View Participants.....	24
Εικόνα 2. 11 Activity Diagram - View Reviews .....	25
Εικόνα 2. 12 Activity Diagram - Join Party .....	27
Εικόνα 2. 13 Activity Diagram - Leave Party .....	28
Εικόνα 2. 14 Activity Diagram - View Recently Joined Parties.....	29
Εικόνα 3. 1 Login/Register/Forget Page - Mockup .....	30
Εικόνα 3. 2 Home/Filters Page- Mockup.....	31
Εικόνα 3. 3 MapPage/PartyPage- Mockup .....	32
Εικόνα 3. 4 Participants/Donate - Mockup .....	33
Εικόνα 3. 5 Host a Party - Mockup .....	34
Εικόνα 3. 6 History/Review-Mockup.....	35
Εικόνα 3. 7 FriendsPage/SearchUsers -Mockup .....	36



## 1 Απαιτήσεις Συστήματος

### 1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)

#### <ΛΑ-1>

Ο χρήστης πρέπει να μπορεί να διοργανώνει πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να κοινοποιεί τις λεπτομέρειες του πάρτι που διοργανώνει, δηλαδή, την ημερομηνία, την τοποθεσία που λαμβάνει χώρα, τον μέγιστο αριθμό ατόμων και την θεματολογία.

**User Priority: 5/5** Η διοργάνωση πάρτι αποτελεί πρωταρχικής σημασίας λειτουργικότητα για το χρήστη και κύρια εφαρμογή του συστήματος μας.

**Technical Priority: 5/5** Η απαίτηση αυτή είναι ιδιαίτερα σημαντική για το σύστημα, διότι η μη διοργάνωση πάρτι αφαιρεί λειτουργικότητα και το καθιστά κενό.

#### <ΛΑ-2>

Ο χρήστης πρέπει να μπορεί να βλέπει τα ενεργά πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να βλέπει τα κοντινά του πάρτι σε ακτίνα των 4 km με κέντρο την τρέχουσα τοποθεσία του χρήστη.

**User Priority: 5/5** Η επίβλεψη των ενεργών πάρτι αποτελεί καίριας σημασίας λειτουργικότητα για τον χρήστη, διότι είναι το πρώτο άμεσο βήμα για την εκδήλωση ενδιαφέροντος του.

**Technical Priority: 5/5** Η απαίτηση αυτή είναι αρκετά σημαντική για το σύστημα, διότι η βασική λειτουργία του και η κατάσταση του επηρεάζονται.

#### <ΛΑ-3>

Ο χρήστης πρέπει να μπορεί να συμμετέχει σε πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να δηλώνει συμμετοχή σε πάρτι.

**User Priority: 5/5** Η συμμετοχή σε πάρτι είναι πρωταρχικής σημασίας λειτουργικότητα για τον χρήστη και την σκοπιμότητα της εφαρμογής.

**Technical Priority: 4/5** Η απαίτηση αυτή είναι εξίσου σημαντική και για το σύστημα, διότι η απουσία της το καθιστά ατελές ως προς την έγκυρη ενημέρωση διαθέσιμου χώρου.

#### <ΛΑ-4>

Ο χρήστης πρέπει να μπορεί να αξιολογεί πάρτι στα οποία έχει συμμετάσχει.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να αξιολογεί πάρτι στα οποία έχει συμμετάσχει μετά των τερματισμών τους.

**User Priority: 3/5** Η ύστερη αξιολόγηση των πάρτι αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη.

**Technical Priority: 2/5** Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, διότι η κατάσταση και η λειτουργία του είναι ανεξάρτητες από αυτήν.

#### <ΛΑ-5>



Ο χρήστης πρέπει να μπορεί να βλέπει την αξιολόγηση του διοργανωτή.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να βλέπει τον μέσο όρο των αξιολογήσεων του διοργανωτή στην σελίδα του πάρτι.

**User Priority: 3/5** Η προβολή αξιολογήσεων του διοργανωτή είναι επιθυμητή λειτουργικότητα για τον χρήστη.

**Technical Priority: 2/5** Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, διότι μπορεί να λειτουργήσει αποτελεσματικά και χωρίς αυτήν.

#### <ΛΑ-6>

Ο χρήστης πρέπει να μπορεί να φιλτράρει τα ενεργά πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να αναζητεί βάσει της απόστασης, αξιολογήσεων διοργανωτή, διαθέσιμου χώρου, θεματολογίας όπως και του πλήθους των συμμετεχόντων προκειμένου να περιορίσει την λίστα των εμφανιζόμενων ενεργών πάρτι.

**User Priority: 3/5** Η δυνατότητα επιλεκτικής προβολής ενεργών πάρτι βάσει μιας λίστας χαρακτηριστικών αποτελεί απλώς μια επιθυμητή λειτουργικότητα για τον χρήστη.

**Technical Priority: 1/5** Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, διότι η κατάσταση του και η λειτουργία του είναι αρκετά ανεξάρτητες από αυτήν.

#### <ΛΑ-7>

Ο χρήστης πρέπει να μπορεί να συνεισφέρει οικονομικά στο πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να συνεισφέρει οικονομικά στο πάρτι μέσω μεταφοράς χρημάτων *Paypal* για να διατηρείται το πνεύμα φιλίας, ομαδικότητας μεταξύ των χρηστών και να ανεβαίνει το επίπεδο ποιότητας των πάρτι.

**User Priority: 3/5** Η οικονομική συνεισφορά στο πάρτι αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη.

**Technical Priority: 2/5** Η απαίτηση αυτή δεν επηρεάζει σημαντικά την ζωτική κατάσταση του συστήματος.

#### <ΛΑ-8>

Ο χρήστης πρέπει να μπορεί να προσθέσει έναν φίλο.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να προσθέσει έναν φίλο πατώντας το κουμπί “Προσθήκη Φίλου” προκειμένου να παραμένει δικτυωμένος.

**User Priority: 3/5** Η προσθήκη φίλου αποτελεί αρεστή λειτουργικότητα για τον χρήστη.

**Technical Priority: 2/5** Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, διότι η κατάστασή του και η λειτουργία του δεν επηρεάζονται σε σημαντικό βαθμό από την προσθήκη φίλου.

#### <ΛΑ-9>

Το σύστημα πρέπει να μπορεί να ειδοποιεί τον χρήστη με email για συμμετοχές φίλων του σε πάρτι.

**Περιγραφή:** Το σύστημα πρέπει να μπορεί να ειδοποιεί με email τον χρήστη για συμμετοχές φίλων του σε πάρτι χρησιμοποιώντας τα στοιχεία του πάρτι και τους συμμετέχοντες.



**User Priority: 2/5** Η ειδοποίηση του χρήστη με email για συμμετοχές φίλων του σε πάρτι έχει χαμηλή αξία λειτουργικότητας για αυτόν.

**Technical Priority: 2/5** Η απαίτηση αυτή είναι μικρής σημασίας για το σύστημα, διότι οι κύριες λειτουργίες του δεν επηρεάζονται από αυτήν και παραμένουν ευσταθείς.

#### <ΛΑ-10>

Ο χρήστης πρέπει να μπορεί να αναζητεί άλλους χρήστες.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να αναζητεί άλλους χρήστες βάσει ονόματος χρήστη σε μία μπάρα αναζήτησης.

**User Priority: 3/5** Η αναζήτηση άλλων χρηστών αποτελεί επιθυμητή λειτουργικότητα για τον χρήστη.

**Technical Priority: 2/5** Η απαίτηση αυτή δεν είναι αρκετά σημαντική για το σύστημα, επειδή η βασική λειτουργία του είναι ανεξάρτητη από αυτήν.

#### <ΛΑ-11>

Ο χρήστης πρέπει να μπορεί να βλέπει τους χρήστες που συμμετέχουν στο πάρτι.

**Περιγραφή:** Ο χρήστης πρέπει να μπορεί να δει ποιοι χρήστες συμμετέχουν σε ένα πάρτι.

**User Priority: 4/5** Η προβολή των συμμετεχόντων σε ένα πάρτι αποτελεί χρήσιμη λειτουργικότητα για τον χρήστη.

**Technical Priority: 3/5** Η απαίτηση αυτή είναι σημαντική για την κατάσταση του συστήματος αλλά όχι απαραίτητη μιας και μπορεί να διατηρήσει ευσταθή την λειτουργία του.

#### <ΛΑ-12>

Ο χρήστης πρέπει να μπορεί να αποχωρήσει από ένα πάρτι.

**Περιγραφή:** Ο χρήστης από την στιγμή που για οποιοδήποτε λόγο μπορεί να χρειαστεί να αποχωρήσει από ένα πάρτι, πρέπει να του δίνεται η δυνατότητα να αφαιρεί την συμμετοχή του από το πάρτι αυτό.

**User Priority: 4/5** Οι συμμετέχοντες ενός πάρτι, όπως και το πλήθος αυτών, είναι ένας εξαιρετικά σημαντικός παράγοντας για τους χρήστες, ο οποίος πρέπει να είναι ακριβής για τον λόγο ότι αποτελεί ένα από τα βασικότερα κριτήρια συμμετοχής κάποιου χρήστη σε ένα πάρτι.

**Technical Priority: 3/5** Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, διότι η κατάσταση του και η λειτουργία του είναι αρκετά ανεξάρτητες από αυτήν.

#### <ΛΑ-13>

Ο χρήστης πρέπει να μπορεί να βλέπει τα πάρτι στα οποία έχει συμμετάσχει.

**Περιγραφή:** Ο χρήστης μπορεί να θελήσει να δει σε πόσα και σε ποια πάρτι έχει συμμετάσχει στο πρόσφατο παρελθόν.

**User Priority: 2/5** Δεν αποτελεί απαραίτητη λειτουργία για τον χρήστη να μπορεί να βλέπει τα πάρτι στα οποία έχει συμμετάσχει αλλά σίγουρα είναι επιθυμητή.

**Technical Priority: 3/5** Η απαίτηση αυτή δεν είναι σημαντική για το σύστημα, διότι η κατάσταση του και η λειτουργία του είναι αρκετά ανεξάρτητες από αυτήν.



## 1.2 Χρήστες και εξωτερικά συστήματα

Το σύστημα αυτό χρησιμοποιείται από τους χρήστες του με σκοπό την συμμετοχή και διοργάνωση πάρτι. Το σύστημα χρησιμοποιεί μία εξωτερική βάση δεδομένων για την αποθήκευση και ανάκτηση πληροφοριών που σχετίζονται με τους χρήστες όπως και τα πάρτι. Επιπλέον χρησιμοποιεί ένα σύστημα ειδοποιήσεων για την ενημέρωση των χρηστών του για διάφορα πάρτι που λαμβάνουν χώρα για τα οποία έχουν δείξει ενδιαφέρον φίλοι του χρήστη. Εν συνεχεία, για την εύκολη μετάβαση των χρηστών στα πάρτι, χρησιμοποιείται ένα εξωτερικό σύστημα πλοήγησης. Τέλος, ένα εξωτερικό σύστημα πληρωμών είναι αναγκαίο, έτσι ώστε να μπορούν οι χρήστες να συνεισφέρουν οικονομικά στα πάρτι που επιθυμούν για να στηρίξουν τους διοργανωτές αυτών.

### 1.2.1<Εγγεγραμμένος χρήστης(Registered User)>

Αποτελεί την μοναδική κατηγορία χρηστών στο σύστημα. Ο χρήστης έχει έναν λογαριασμό και για να χρησιμοποιήσει την εφαρμογή θα πρέπει να συνδεθεί στο σύστημα μέσω ενός λογαριασμού.

### 1.2.2<Βάση δεδομένων(Database)>

Ένα αναγκαίο εξωτερικό σύστημα για την ύπαρξη της συγκεκριμένης εφαρμογής αποτελεί μία βάση δεδομένων, η οποία διατηρεί όλα τα στοιχεία των χρηστών της όπως και πληροφορίες για τα πάρτι που διοργανώνονται. Τέλος, διατηρεί μία λίστα ιστορικού για τα πάρτι που έχει συμμετάσχει κάθε χρήστης της εφαρμογής.

### 1.2.3<Σύστημα ειδοποιήσεων(Notification System)>

Είναι ένα αναγκαίο εξωτερικό σύστημα το οποίο θα ειδοποιεί τους χρήστες μέσω email όταν οι “φίλοι” του χρήστη συμμετέχουν σε ένα πάρτι.

### 1.2.4<Σύστημα πλοήγησής (Google Maps)>

Είναι ένα αναγκαίο εξωτερικό σύστημα για την ύπαρξη της εφαρμογής καθώς θα δίνει την δυνατότητα στον χρήστη να δει την τοποθεσία των πάρτι σε σχέση με την τρέχουσα τοποθεσία του. Το εξωτερικό σύστημα αυτό χρησιμοποιεί ένα έτοιμο σύστημα βασισμένο στο Google Maps.

### 1.2.5<Σύστημα πληρωμών (Paypal)>

Είναι ένα αναγκαίο εξωτερικό σύστημα με στόχο ο κάθε χρήστης να μπορεί να συνεισφέρει οικονομικά και με αυτόν τον τρόπο να υποστηρίξει την διοργάνωση και διεξαγωγή των πάρτι.

## 1.3 Σημαντικές μη λειτουργικές απαιτήσεις

### 1.3.1 Απαιτήσεις επίδοσης

<ΜΛΑ- 1>: Το σύστημα πρέπει να βρίσκει τα πάρτι που λαμβάνουν χώρα σε ακτίνα 4km με κέντρο την τοποθεσία του χρήστη με χρόνο απόκρισης μικρότερο του ενός δευτερολέπτου.





**Περιγραφή:** Το σύστημα πρέπει να μπορεί να βρίσκει τα κοντινότερα πάρτι βάσει την τοποθεσία του χρήστη υπό τον περιορισμό ενός χρονικού παραθύρου, με σκοπό την βελτίωση της εμπειρίας του χρήστη.

**User Priority: 4/5** Είναι αρκετά σημαντικό για τον χρήστη να ανταποκρίνεται γρήγορα η εφαρμογή καθώς θα βελτιώσει την εμπειρία του χρήστη και θα αυξήσει τις πιθανότητες να συνεχίσει να την χρησιμοποιεί.

**Technical Priority: 4/5** Η απαίτηση αυτή είναι σημαντική για το σύστημα καθώς μπορεί να είναι δύσκολο να ανταποκρίνεται με τέτοιο ρυθμό.

**<ΜΛΑ- 2>:** Το σύστημα πρέπει να μπορεί να εξυπηρετήσει ταυτόχρονα τουλάχιστον 500 χρήστες.

**Περιγραφή:** Εφόσον το σύστημα θα μπορεί να χρησιμοποιηθεί για την οργάνωση πάρτι μεγάλης χωρητικότητας είναι απαραίτητο να μπορούν να χρησιμοποιήσουν την ίδια στιγμή ένας μεγάλος αριθμός ατόμων.

**User Priority: 5/5** Για τους χρήστες είναι μία απολύτως απαραίτητη απαίτηση αφού διαφορετικά δεν θα έχουν την δυνατότητα να χρησιμοποιήσουν την εφαρμογή.

**Technical Priority: 5/5** Αποτελεί επίσης μία σημαντική απαίτηση και για το ίδιο το σύστημα, διότι, ο περιορισμός αυτός ίσως επηρεάσει την σχεδίαση αυτού με σκοπό να τον ικανοποιεί.

### 1.3.2 Απαιτήσεις ασφάλειας (Security)

**<ΜΛΑ- 3>:** Το σύστημα πρέπει να ικανοποιεί την μέγιστη ασφάλεια των προσωπικών δεδομένων του κάθε χρήστη.

**Περιγραφή:** Το σύστημα πρέπει να μπορεί να αντιμετωπίσει κακόβουλες επιθέσεις, ώστε να επιτυγχάνεται η ασφάλεια των προσωπικών δεδομένων των χρηστών, να έχει ασφαλή διασύνδεση με το περιβάλλον συναλλαγής της Payral και να διασφαλίζεται η ομαλή λειτουργία του.

**User Priority: 4/5** Η απαίτηση αυτή είναι ιδιαίτερα σημαντική για τον χρήστη, καθώς θα ήταν διστακτικός απέναντι στη χρήση του, αν δεν υπήρχε αξιόπιστη διασφάλιση των προσωπικών δεδομένων του.

**Technical Priority: 5/5** Η απαίτηση αυτή είναι εξίσου σημαντική για το σύστημα, καθώς εάν δεν μπορεί, να αντιμετωπίσει με επιτυχία διάφορες απόπειρες επίθεσης, κινδυνεύει να καταρρεύσει και να υποκλαπούν δεδομένα.

### 1.3.3 Απαιτήσεις χρηστικότητας (Usability)

**<ΜΛΑ- 4>:** Ο χρήστης πρέπει να μπορεί να μάθει να χρησιμοποιεί το σύστημα σε λιγότερο από 10 λεπτά.

**Περιγραφή:** Ο χρήστης πρέπει να μάθει να χρησιμοποιεί το σύστημα σε λιγότερο από 10 λεπτά με βοηθητική επεξήγηση ενός μονοπατιού με την βασική λειτουργία της εκάστοτε εφαρμογής.



**User Priority: 5/5** Η απαίτηση αυτή είναι πρωταρχικής σημασίας για τον χρήστη διότι σκοπός της εφαρμογής είναι να προσφέρει έναν απλό τρόπο δικτύωσής του με άλλους χρήστες για την ανεπιτήδευτη διεξαγωγή πάρτι.

**Technical Priority: 1/5** Δεν κρίνεται απαραίτητο για την ορθή λειτουργία του συστήματος η ευκολία χρησιμοποίησης αυτού από τον χρήστη .

#### 1.3.4 Απαιτήσεις Φορητότητας (Portability)

**<ΜΛΑ- 5>**: Το σύστημα πρέπει να υποστηρίζει τα λειτουργικά συστήματα iOS και Android. Συγκεκριμένα πρέπει να υποστηρίζει τις Android εκδόσεις από το 8.0 (Oreo) και μετά όπως και τις iOS εκδόσεις από το iOS 11 και μετά.

##### Περιγραφή:

Το σύστημα πρέπει να υποστηρίζει τα λειτουργικά συστήματα Android και iOS από την στιγμή που αποτελούν τα δύο δημοφιλέστερα λειτουργικά συστήματα για “mobile” εφαρμογές.

**User Priority: 5/5** Η απαίτηση αυτή είναι μεγάλης σημασίας για τον χρήστη διότι η εφαρμογή πρέπει να υποστηρίζεται από την συσκευή του χρήστη.

**Technical Priority: 5/5** Η απαίτηση της επιλογής των λειτουργικών συστημάτων στα οποία θα αναπτυχθεί η εφαρμογή είναι άκρως σημαντική από την στιγμή που επηρεάζει σε μεγάλο βαθμό την σχεδίαση και ανάπτυξη αυτού.

#### 1.3.5 Απαιτήσεις πολιτικής (Policy - legal, corporate)

**<ΜΛΑ- 6>**: Το σύστημα πρέπει να ικανοποιεί τους περιορισμούς ασφάλειας που τίθενται από το GDPR.

**Περιγραφή:** Τα δεδομένα των χρηστών πρέπει να τηρούν το πρωτόκολλο κανονισμού GDPR και οι χρήστες να αποδέχονται τις σχετικές ενημερώσεις με την πολιτική διαχείρισης των δεδομένων τους

**User Priority: 4/5** Η απαίτηση αυτή είναι αρκετά σημαντική για τον χρήστη καθώς ο σύγχρονος πολίτης ενδιαφέρεται σημαντικά για τον τρόπο διαχείρισης των προσωπικών του δεδομένων.

**Technical Priority: 3/5** Η απαίτηση αυτή δεν επηρεάζει άμεσα το σύστημα. Ωστόσο πρέπει να δίνεται προσοχή στην τήρηση τους, καθώς τυχούσα παραβίαση της μπορεί να οδηγήσει σε νομικά ζητήματα που πιθανόν να υποχρεώσουν τον τερματισμό της λειτουργίας του συστήματος.

#### 1.10 Λεξικογραφικοί Προσδιορισμοί

**Πάρτι(party)** : Προσδιορίζει την τοποθεσία, ημερομηνία που λαμβάνει χώρα, την χωρητικότητα της τοποθεσίας στην οποία θα διεξαχθεί, όπως και την θεματολογία του πάρτι.

**Χρήστης(user)** : Ο άνθρωπος που θα χρησιμοποιεί την εφαρμογή.

**Σύστημα(system)** : Αποτελεί τα μέρη που επιδρούν στην πραγμάτωση και στην ομαλή λειτουργία της εφαρμογής κατά τη διάρκεια χρήσης.



**Διοργάνωση(host party)** : Διαδικασία δημιουργίας και καταχώρησης λεπτομερειών ενός πάρτι.

**Αξιολόγηση(review)** : Αποτελεί την βαθμολογία που αποδίδει ο χρήστης σε ένα πάρτι με γνώμονα την καλή διεξαγωγή του.

**Οικονομική συνεισφορά(donation)** : Ο κάθε χρήστης που θα συμμετέχει στο πάρτι μπορεί να βοηθήσει τον διοργανωτή οικονομικά μέσω PayPal.

**Ενεργά Πάρτι(active parties)** : Πάρτι τα οποία θα λάβουν χώρα μελλοντικά η λαμβάνουν χώρα την παρούσα στιγμή.

**Φίλος(friend)** : Άλλοι χρήστες που έχουν προστεθεί σε μια λίστα χρηστών.

**Συμμετοχή(join party)** : Η εκδήλωση ενδιαφέροντος ενός χρήστη να παρευρεθεί σε ένα πάρτι

**Προσθήκη Φίλου (add friend)** : Ο χρήστης μπορεί να προσθέσει σε μία λίστα φίλων έναν άλλον χρήστη.

**Προσωπικά στοιχεία(personal information)** : Αποτελείται από φύλο, ηλικία, ονοματεπώνυμο, αξιολογήσεις, συμμετοχές.

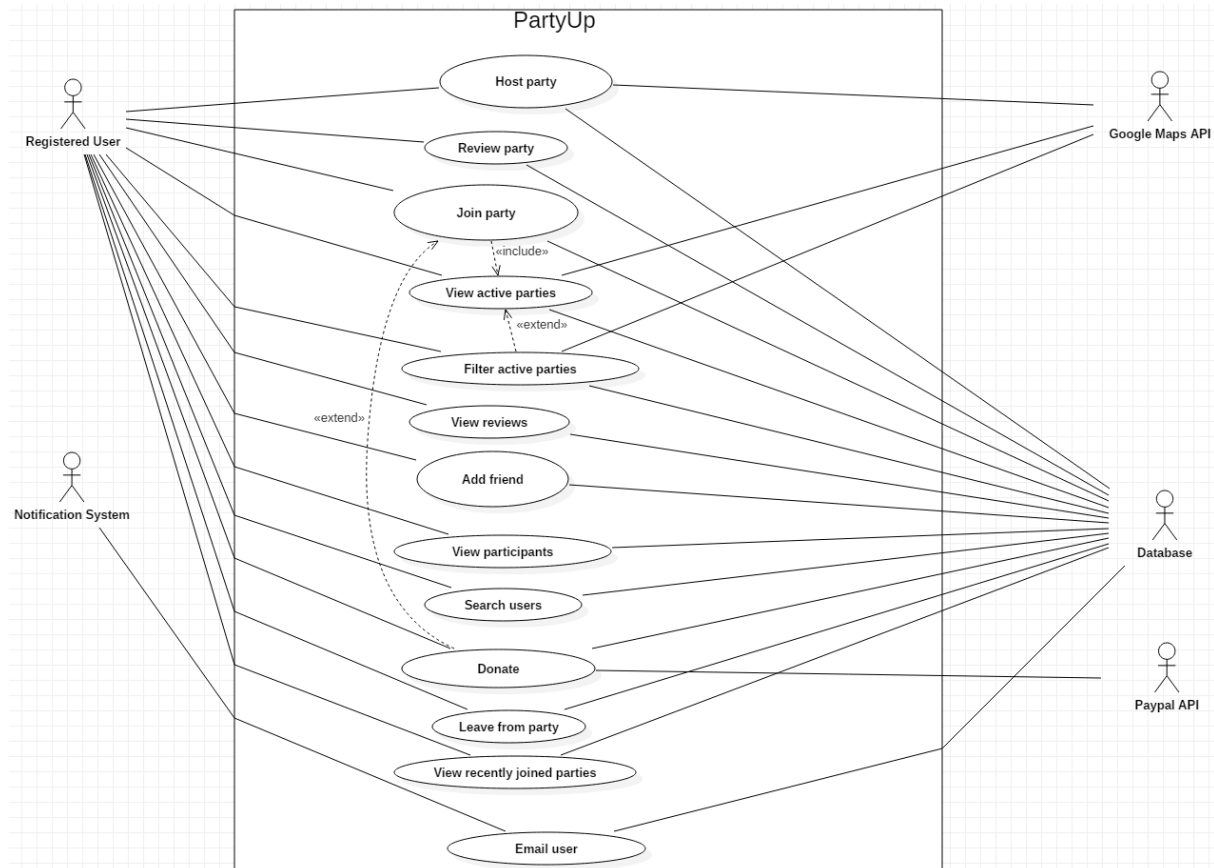
**Χρόνο απόκρισης(response time)** : Ο ελάχιστος χρόνος που απαιτείται για την ομαλή λειτουργία του συστήματος.

**Προφίλ(profile)** : Η σελίδα προβολής των προσωπικών στοιχείων του χρήστη και των αξιολογήσεών του ως διοργανωτή πάρτι.



## 2 Σενάρια Χρήσης

### 2.1 Διάγραμμα σεναρίων χρήσης



Εικόνα 2. 1 Use cases diagram

### 2.2 <Feature 1: Add a Friend> (Feature με βάση την ορολογία Gherkin)

**Feature:** Add a Friend

**Background:**

**Given** that I am the user

**And** I am logged in

**Scenario:** Successful addition of a friend

**Given** that I am on friends page or participants page

**And** he is not already your friend

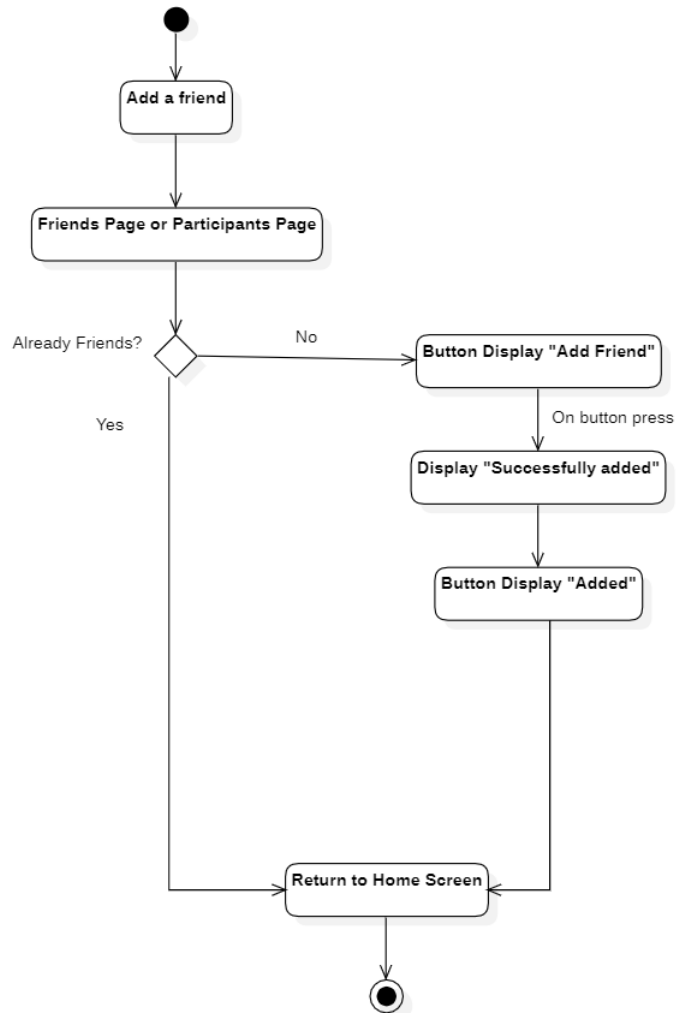
**And** can see the button "Add Friend"

**When** I press the button "Add Friend"

**Then** I should see a message saying "Successfully added."

**And** the button change status to "Added"

**And** I should be able to go back to the home page



Εικόνα 2. 2 Activity Diagram - Add Friend

### 2.3 <Feature 2: View active parties> (Feature με βάση την ορολογία Gherkin)

**Feature:** View active parties

**Background:**

**Given** that I am the user

**And** I am logged in

**Scenario:** Successful view active parties

**Given** that I am on home page

**And** there are active parties at a distance of 4 km

**Then** I should see a list of active parties:



#Name	#Example	#Description/Limits
Tittle of Party	["Dance and Drink with your soul"]	Maximum string spacing 60 characters
Theme of the party	["90", "Hip-Hop", "Electronic music", "Mixed", "Greek", "Other"]	
They can be selected from 0 to 7		
Start Time	["9:00 pm"]	Any time without restrictions

**Scenario:** There is no active party

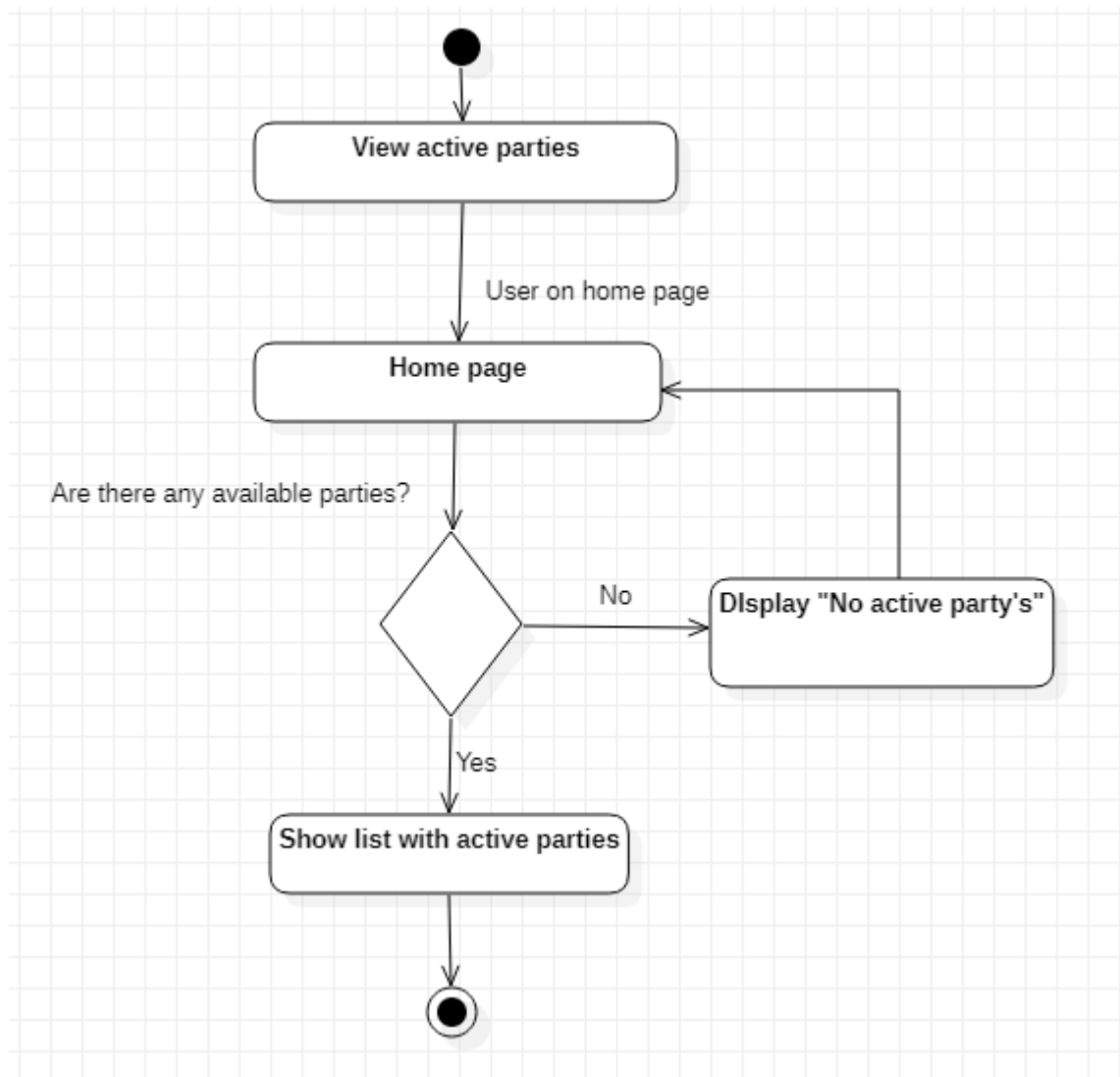
**Given** that I am on home page

**And** there are no active parties within 4 km

**Then** I can't see a list of active parties:

#Name	#Example	#Description/Limits
Tittle of Party	["Dance and Drink with your soul"]	Maximum string spacing 60 characters
Theme of the party	["90", "Hip-Hop", "Techno", "Electronic music", "Mixed", "Greek", "Other"]	
]   They can be selected from 0 to 7		
Start Time	["9:00 pm"]	Any time without restrictions

**And** I should see a message saying "No active parties"



Εικόνα 2. 3 Activity Diagram – View Active Parties



## 2.4 <Feature 3: Filter active parties> (Feature με βάση την ορολογία Gherkin)

**Feature:** Filter active parties

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** Filtering active parties

**Given** that I am on Home page

**And** there is not already filter active parties

**And** can see the button "Filter"

**When** I press the button "Filter"

**Then** scroll bar appears:

#Name	#Example	#Description/Limits
The best organizer	Return the highest hosters, maximum 5	
Themes of the party	["90", "Hip-Hop", "Electronic music", "Mixed", "Greek", "Other"]	
They can be selected from 0 to 7		
Distance	["Near You", "5km", "5km+"]	They can be selected from 0 to 3
Capacity	[50, 100, 200, 200+]	They can be selected from 0 to 4

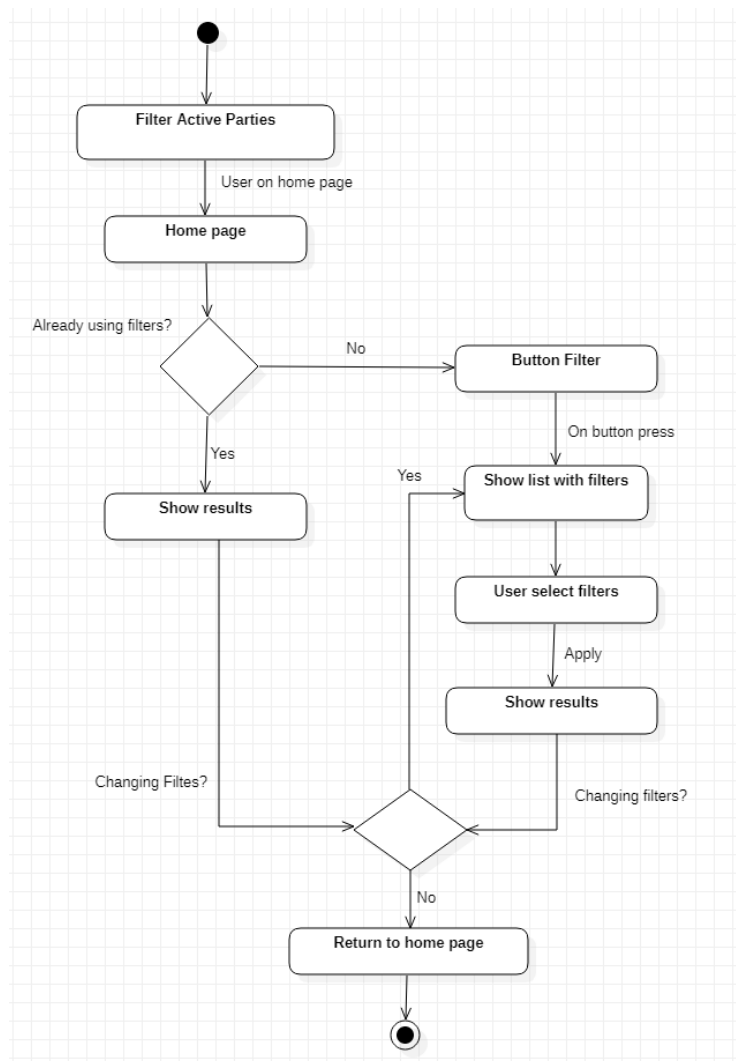
**When** I press apply

**Then** I need to see the list of choices I made

**And** I should be able to change again the filters

**And** I should be able to go back to the home page





Εικόνα 2. 4 Activity Diagram – Filter Active Parties

## 2.5 <Feature 4: Make a donation> (Feature με βάση την ορολογία Gherkin)

**Feature:** Donate

**Background:**

**Given** that I am the user

**And** I am logged in

**Scenario:** Make a donation

**Given** that I have joined a party

**And** I am at the party page

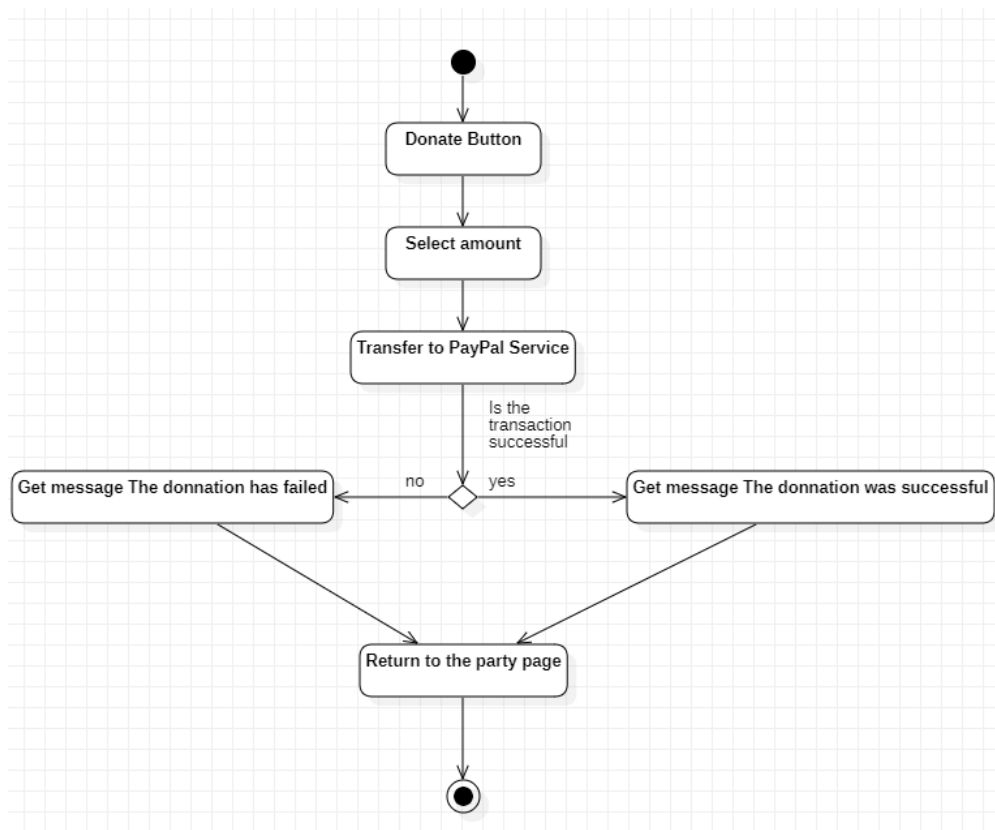
**When** I click the "Donate" button



**And** I specify how much money should be donated  
**Then** I am redirected to the PayPal service  
**When** the transaction is successful  
**Then** I get the message "The donation was successful"  
**And** I am returned to the party page

**Scenario:** Unsuccessful donation

**Given** that I have joined a party  
**And** I am at the party page  
**When** I click the "Donate to party" button  
**And** I specify how much money should be donated  
**Then** I am redirected to the PayPal service  
**But** the transaction is unsuccessful  
**Then** I get the message "The donation has failed"  
**And** I am returned to the party page



Εικόνα 2. 5 Activity Diagram – Donate



---

## 2.6 <Feature 5: Host a party> (Feature με βάση την ορολογία Gherkin)

**Feature:** Host a Party

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** Host party

**Given:** I am in the home page

**And** I want to host a party

**When** I click the "Host a party" button

**And** I fill in the details of the party:

| Title | "Drink with your soul" | Maximum 50 characters |

| Date | "13/5/2022" |

| Location | "Λεωφόρος Βασιλίσσης Όλγας 33" |

| Capacity | [50, 100, 200, 200+] | Maximum 1 |

| Theme | ["90'", "Hip-Hop", "Electronic music", "Mixed", "Greek", "Other" ] | Maximum 1 |

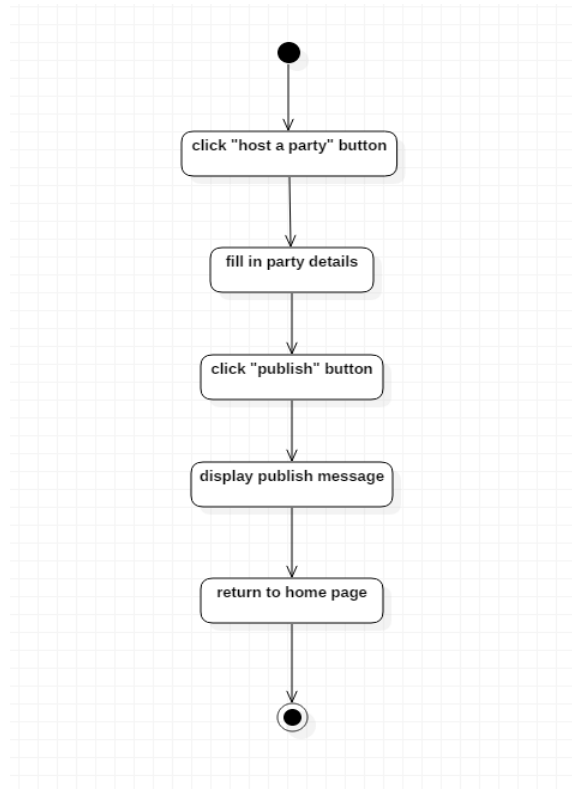
**And** I click the "Publish" button

**Then** I should see a message "You have successfully published a party"

**And** return to the home page



Εικόνα 2. 6 Activity Diagram – Host a Party



## 2.7 <Feature 6: Review a party> (Feature με βάση την ορολογία Gherkin)

**Feature:** Review a Party

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** Successfully Review Party

**Given** I have joined a party

**And** I am on History page

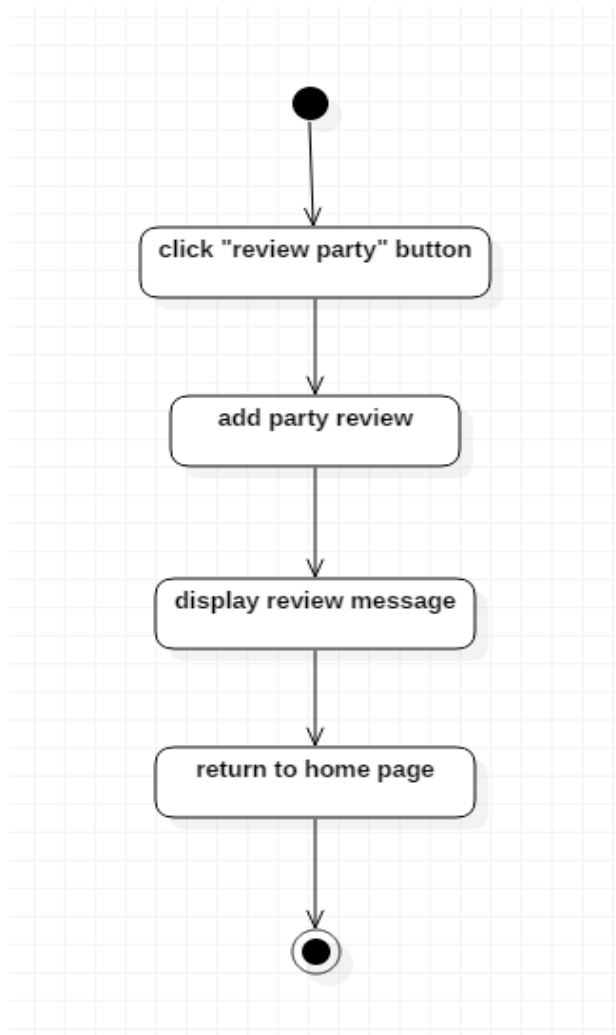
**When** I click the “Review Party” button

**And** I add my review:

#Name	#Example	#Description/Limits
Star	[1,2,3,4,5]	Maximum 1

**Then** I should see a message “Thank you for your valuable feedback”

**And** return to the home page



Εικόνα 2. 7 Activity Diagram – Review Party

## 2.8 <Feature 7: Email user> (Feature με βάση την ορολογία Gherkin)

**Feature:** Email user

**Background:**

**Given** that I am the notification system

**Scenario:** Successful email sent to user

**When** a friend joins a party

**And** I get triggered

**Then** I send email to the user's email ID with the details of the party

**And** the email has been received successfully

**Scenario:** Unsuccessful email sent to user

**When** a friend joins a party

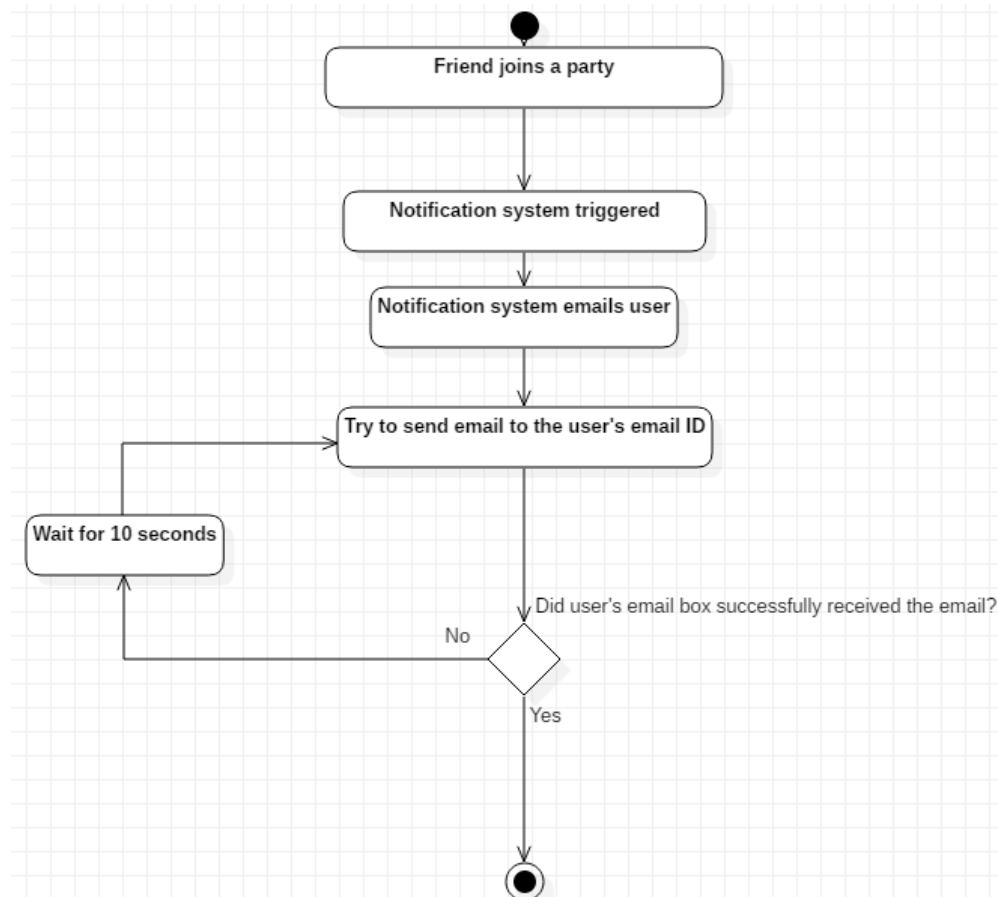
**And** I get triggered



**Then** I send email to the user's email ID with the details of the party

**But** the email has not been received successfully

**Then** I retry in 10 seconds to resend the email



Εικόνα 2. 8 Activity Diagram – Email User

## 2.9 <Feature 8: Search users> (Feature με βάση την ορολογία Gherkin)

**Feature:** Search users

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** Searching of users matched results

**Given** I am on the Friends page

**When** I click the "Search for users" button

**Then** keyboard section pops out on my phone

**And** I specify the name of the user I want to search



#Name	#Example
search input	[ 'Kostas', 'Maria' ]

**When** the searching is successful

**Then** I should see a message "Search results for users matched the below"

**And** I should see a list of user names

#Name	#Example
search results	[ 'Kostas Papadopoulos ', ' Kostas Tzortziou', 'Maria Bukitsi' ]

**Scenario:** Searching of users did not match any results

**Given** I am on the Friend List page

**When** I click the "Search for users" button

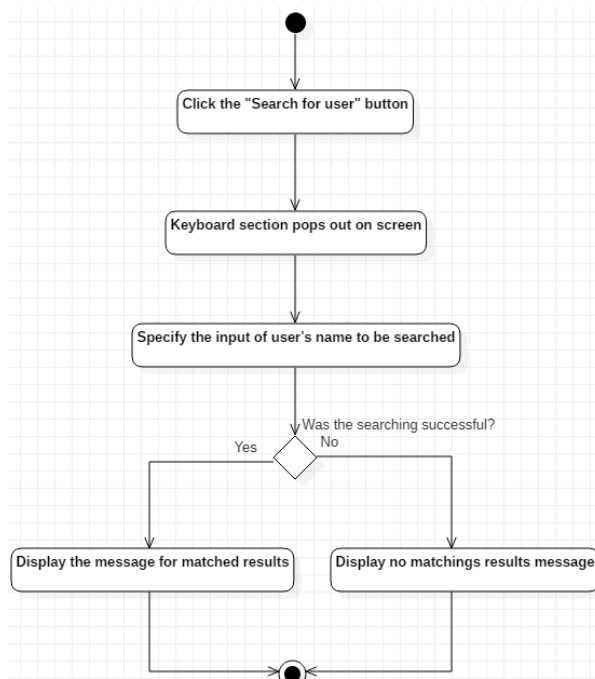
**Then** keyboard section pops out on my phone

**And** I specify the name of the user I want to search

**But** the searching is unsuccessful

**Then** I should see a message "Your search didn't match any user."

**And** I am returned to the home page



Εικόνα 2. 9 Activity Diagram – Search Users

## 2.10 <Feature 9: View participants> (Feature με βάση την ορολογία Gherkin)

**Feature:** View participants

**Background:**

**Given** that I am a user



And I am logged in

**Scenario:** Successful view of participants

**Given** that I am on the party page

**When** I click the "View participants" button

**Then** I see a list of the participants of the party by this form :

#Name	#Example	#Description/Limits
Friend users	‘Giorgos Papadopoulos’	Projected at the top of the participant list
Non-friend users	‘Maria Georgiadou’	Projected sequentially after friend users

**Scenario:** Unsuccessful view of participants

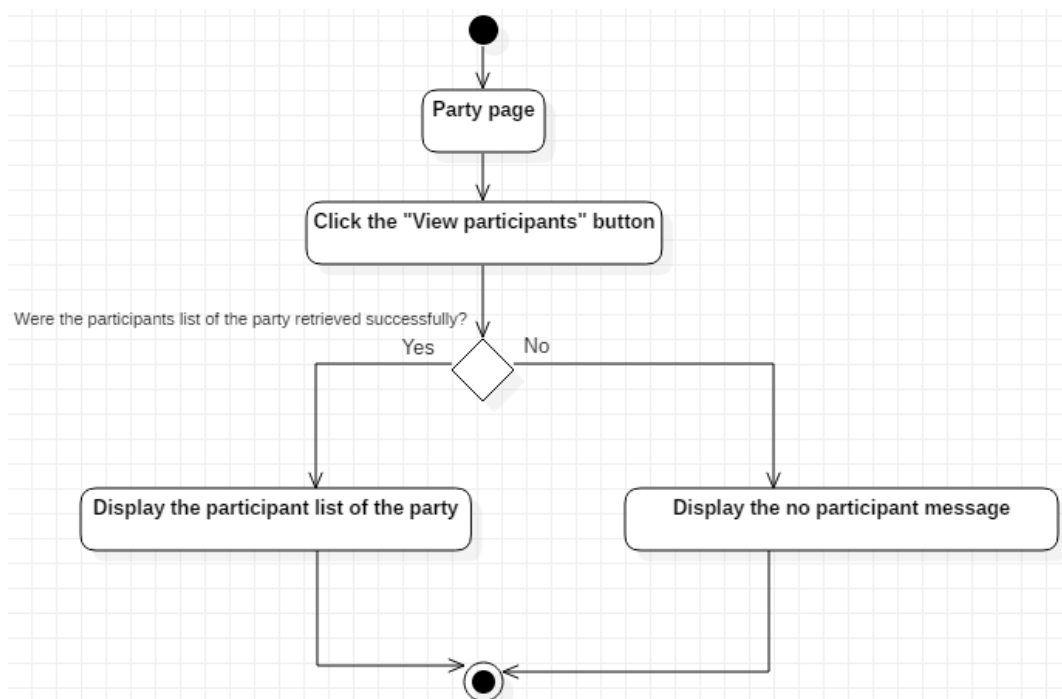
**Given** that I am on the party page

**When** I click the "View participants" button

**But** participant list has 0 participants

**And** I should see a message “No participants in this party yet. Party up first!”

**And** I am returned to the party page



Εικόνα 2. 10 Activity Diagram – View Participants

## 2.11 <Feature 10: View user's reviews> (Feature με βάση την ορολογία Gherkin)

**Feature:** View user's reviews

**Background:**

**Given** that I am a user





And I am logged in

**Scenario:** View user's reviews

**Given** that a party is active

**When** I select it on the map

**Then** I am provided with the details of the party

**And** I can see the reviews of the host

**Scenario:** Host has no reviews

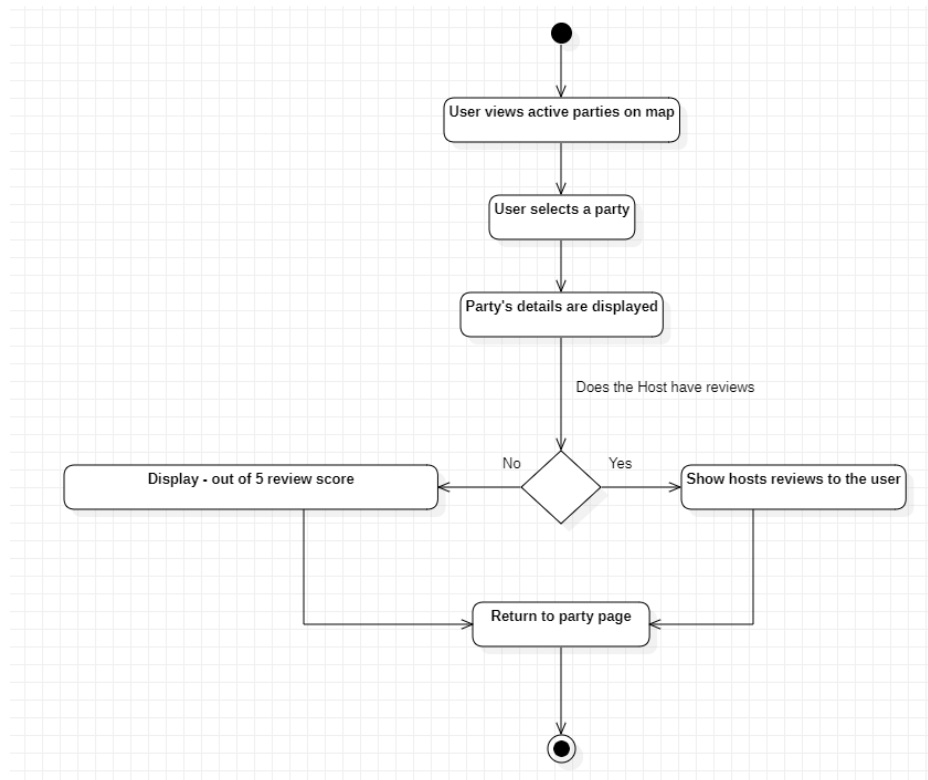
**Given** that a party is active

**When** I select it on the map

**But** the host has no reviews

**Then** I am provided with the details of the party

**And** the host's review score is “-/5”



Εικόνα 2. 11 Activity Diagram – View Reviews

## 2.12 <Feature 11: Join Party > (Feature με βάση την ορολογία Gherkin)

**Feature:** Join a party

**Background:**



**Given** that I am a user

**And** I am logged in

**Scenario:** Join a party

**Given** that a party is active

**When** I select it on the map

**Then** I get directed to party page

**And** I am provided with the details of the party

**When** I click the "Join Party" button

**And** there is enough availability

**Then** I get the message "Joined party successfully"

**And** the availability of the party is updated

**Scenario:** Party is full

**Given** that a party is active

**When** I select it on the map

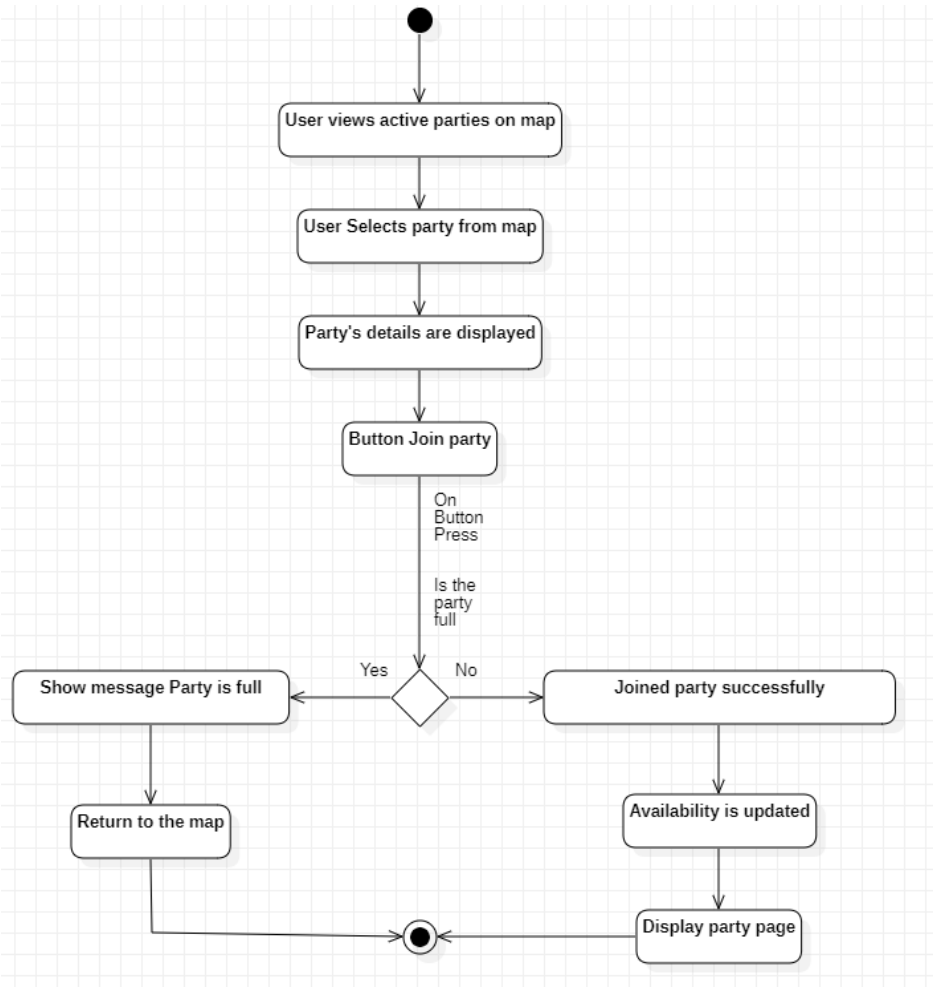
**Then** I am provided with the details of the party

**When** I click the "Join Party" button

**But** The party is full

**Then** I get the message "Party is full"

**And** I am returned to the map



Εικόνα 2. 12 Activity Diagram – Join Party

### 2.13 <Feature 12: Leave from a party> (Feature με βάση την ορολογία Gherkin)

**Feature:** Leave from a party

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** Leave from a party

**Given** that I have joined a party

**And** I want to leave

**When** I click the “Leave party” button

**And** confirm my choice

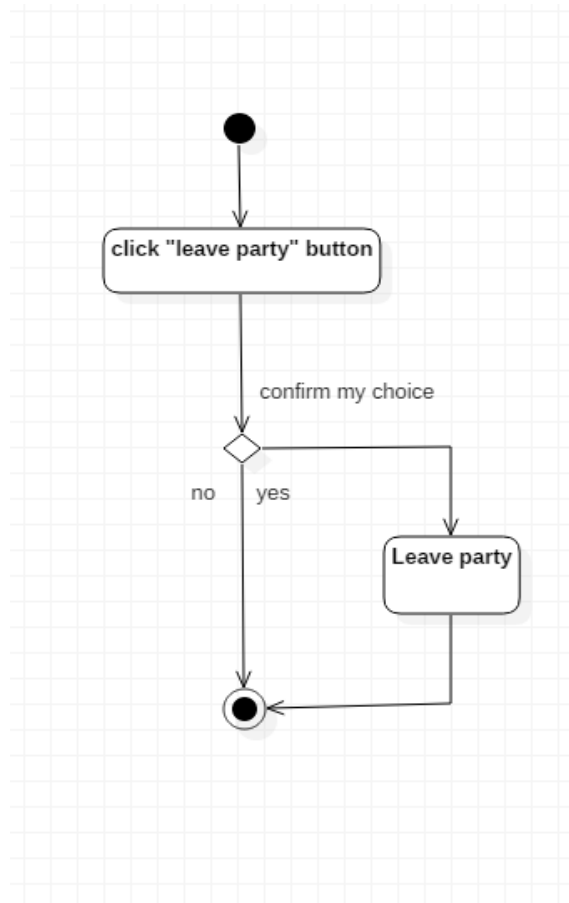
**Then** I should no longer be a member of the party

**Scenario:** Attempted to leave but changed my mind

**Given** that I have joined a party



**And** I want to leave  
**When** I click the “Leave party” button  
**But** I don’t confirm my choice  
**Then** I should still be a member of the party



Εικόνα 2. 13 Activity Diagram – Leave Party

## 2.14 <Feature 13: View recently joined parties> (Feature με βάση την ορολογία Gherkin)

**Feature:** View recently joined parties

**Background:**

**Given** that I am a user

**And** I am logged in

**Scenario:** There are parties that I have joined

**Given** that I have joined in at least one party

**And** I want to view the parties I have joined

**When** I click the “View recently joined parties” button

**Then** I should be able to view the parties that I have joined in the past

**Scenario:** I haven’t joined in any party so far



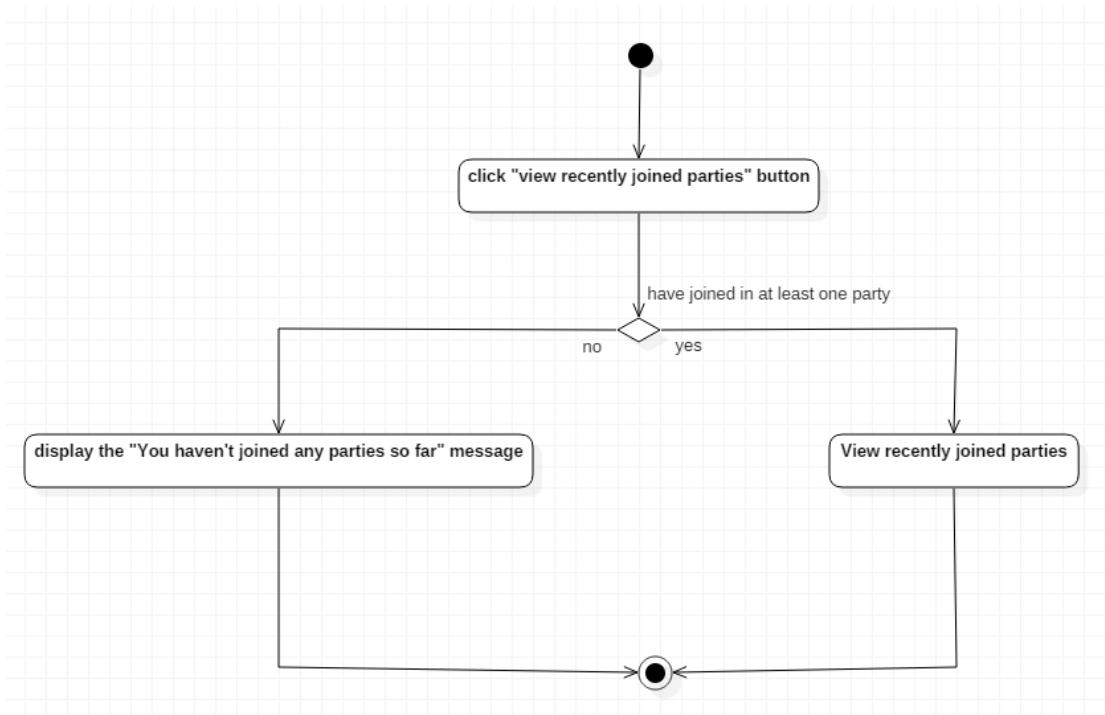
**Given** that I haven't yet joined a party

**And** I want to view the parties I have joined

**When** I click the "View recently joined parties" button

**Then** the "You haven't joined any parties so far" message should be displayed

**And** remain in that window

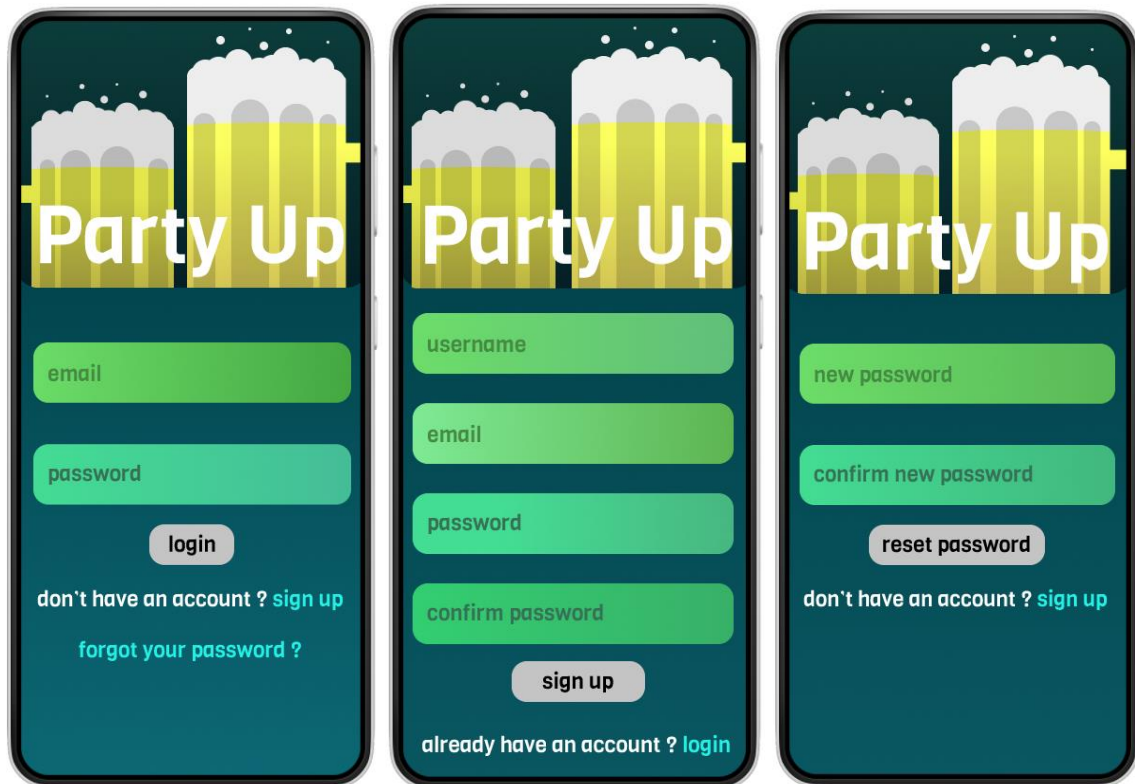


Εικόνα 2. 14 Activity Diagram – View Recently Joined Parties



### 3 Επιδεικτικά γραφικά παράθυρα διεπαφής

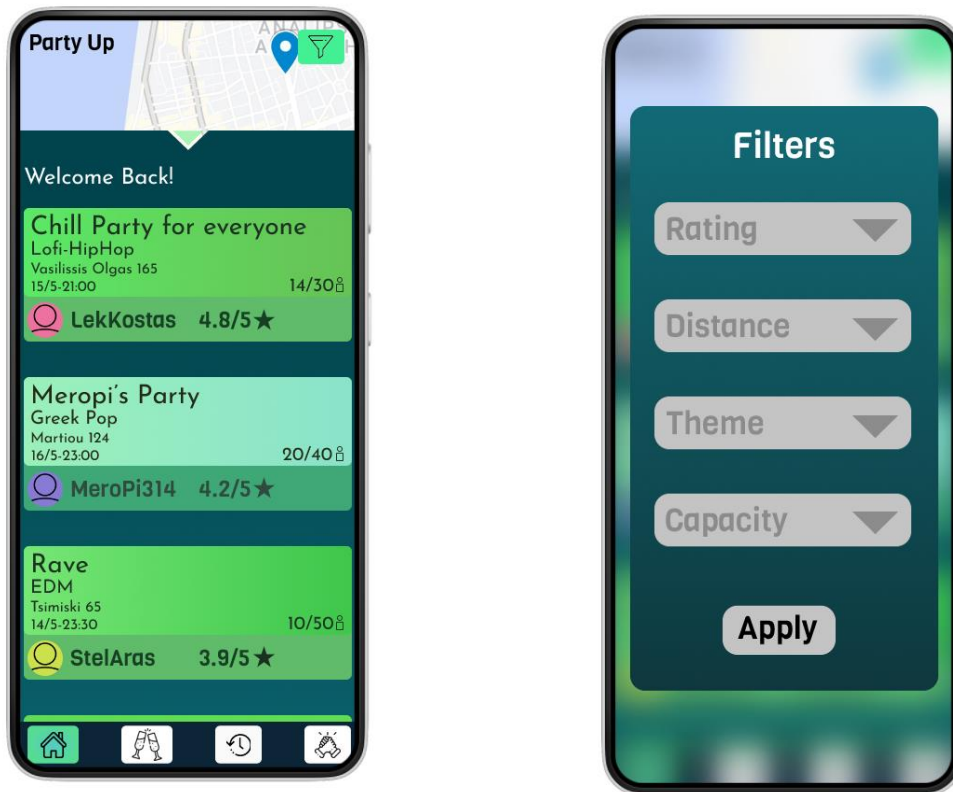
#### 3.1 Σελίδα Σύνδεσης/Εγγραφής και ανάκτηση κωδικού



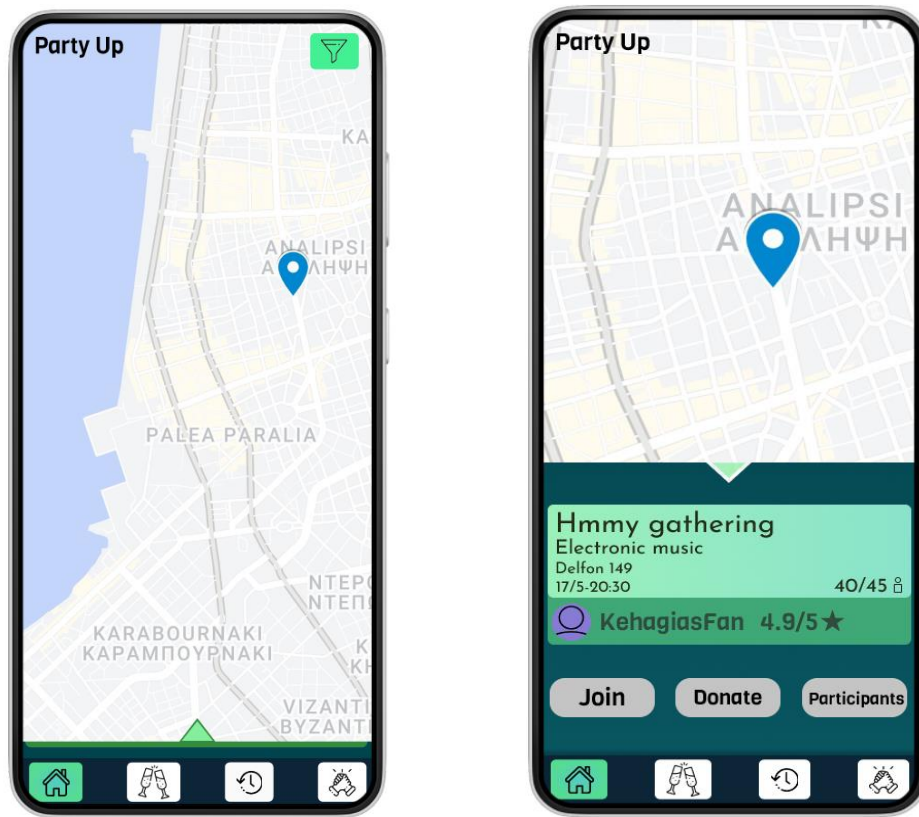
Εικόνα 3. 1 Login/Register/Forget Page – Mockup



### 3.2 Σελίδα αρχική οθόνης

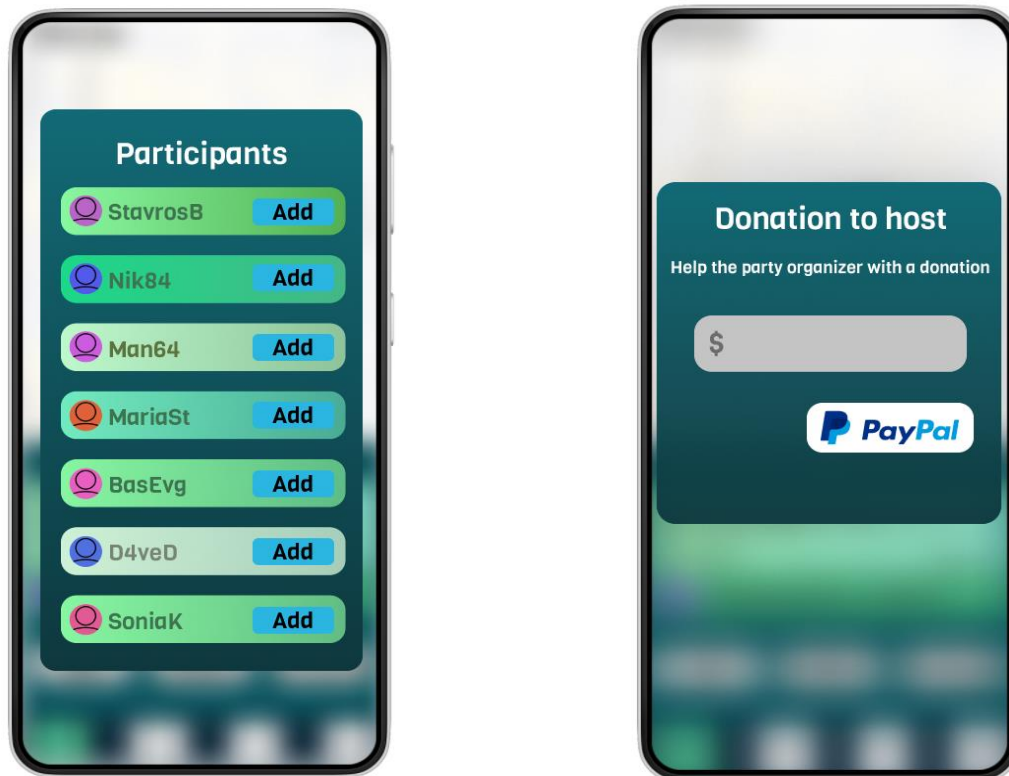
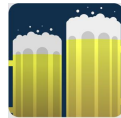


Εικόνα 3. 2 Home/Filters Page- Mockup



Εικόνα 3. 3 MapPage/PartyPage- Mockup





Εικόνα 3. 4 Participants/Donate - Mockup

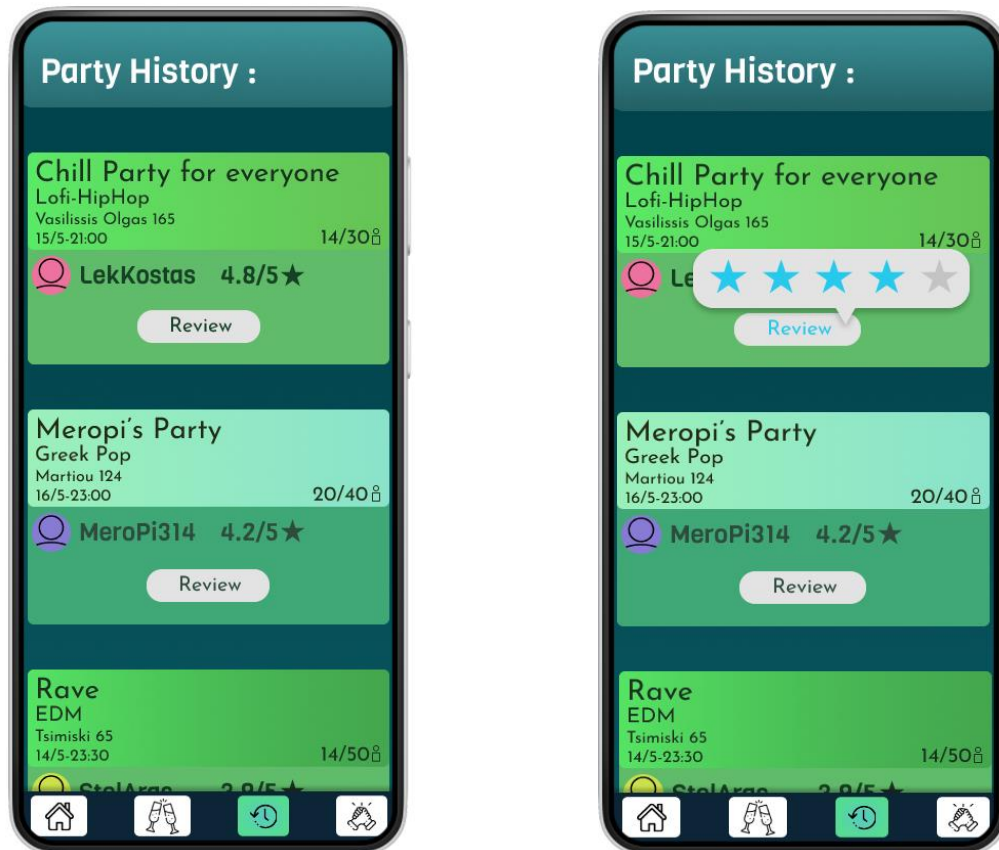


### 3.3 Σελίδα διοργανωτή

Εικόνα 3. 5 Host a Party - Mockup



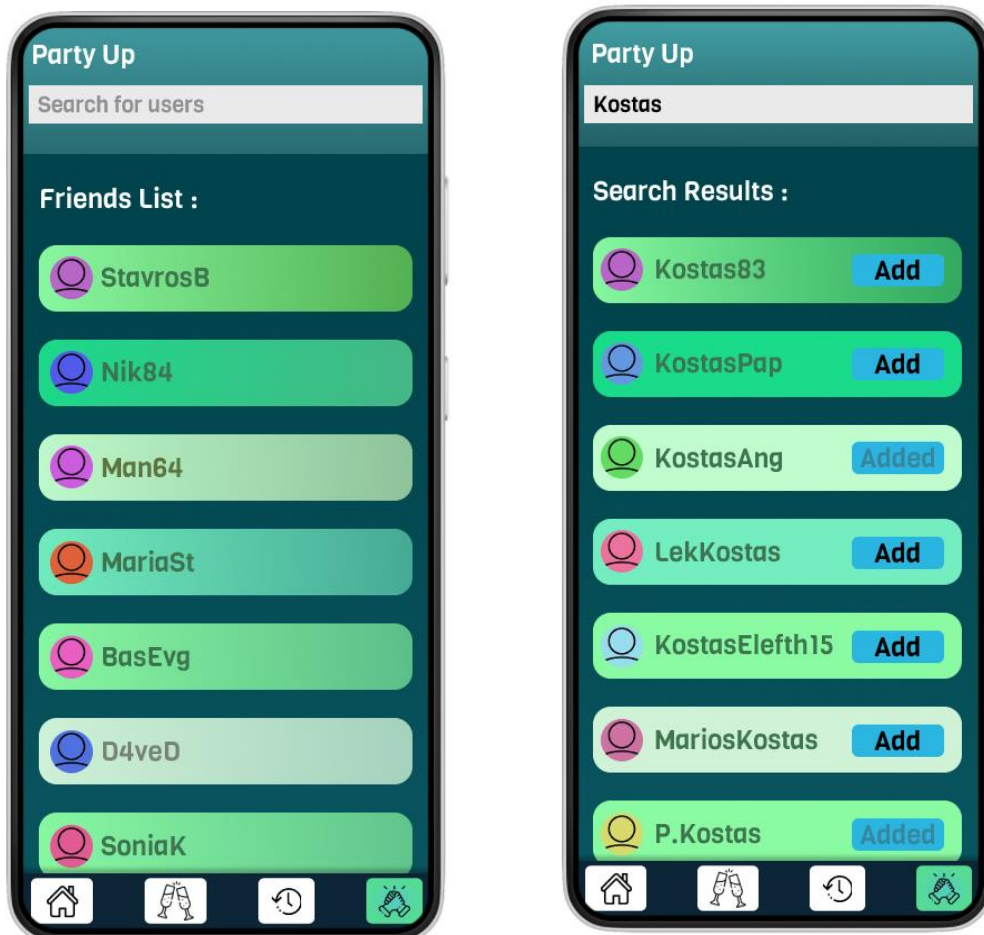
### 3.4 Σελίδα ιστορικού



Εικόνα 3. 6 History/Review-Mockup



### 3.5 Σελίδα Φίλων



Εικόνα 3. 7 FriendsPage/SearchUsers -Mockup

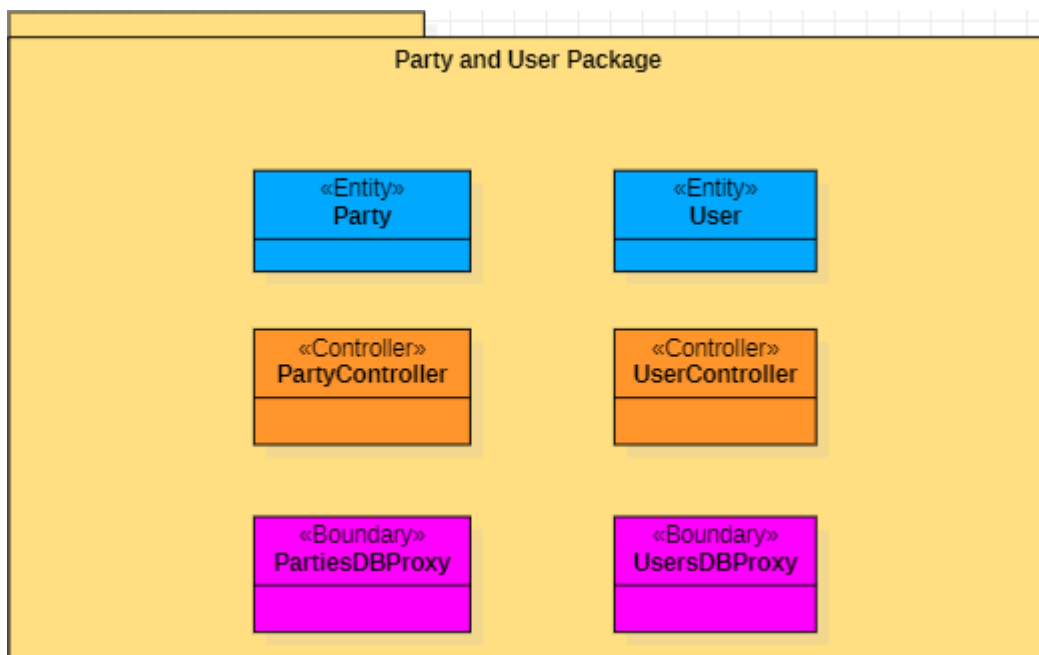


## 4 Στατική Μοντελοποίηση

### 4.1 <Πακέτο(α) λεξιλογίου σεναρίων υψηλής προτεραιότητας>

#### 4.1.1 Πακέτο Party and User

Στο πακέτο αυτό ορίζονται οι κλάσεις οντοτήτων του χρήστη και του πάρτι με τους αντίστοιχους ελεγκτές τους και τις βάσεις δεδομένων τους για την περιγραφή θεμελιωδών χαρακτηριστικών και μεθόδων που καλούν κι άλλα πακέτα κατά την αλληλεπίδραση.





#### 4.1.1.1 Ορισμοί - Πακέτο Party and User

##### Entity Party

«Entity» Party
-partyID: int -active: boolean -participants: list<userID> -theme: string -location: list<float> -date: string -host: userID -capacity: int -rating: float
+Party(active: boolean, theme: string, location: list<float>, date: string, host: userID, capacity: int): void +setPartyID(partyID: int): void +getPartyID(): int +setActive(active: boolean): void +getActive(): boolean +setParticipants(participants: list<userID>): void +getParticipants(): list<userID> +setTheme(theme: string): void +getTheme(): string +setLocation(location: list<float>): void +getLocation(): list<float> +setDate(date: string): void +getDate(): string +setHost(host: userID): void +getHost(): userID +setCapacity(capacity: int): void +getCapacity(): int +setRating(rating: float): void +getRating(): float

Η κλάση αυτή περιέχει τα χαρακτηριστικά και τις μεθόδους για την πλήρη ανάλυση της οντότητας του πάρτι.

Χαρακτηριστικά του πάρτι:

- partyID: Ο μοναδικός χαρακτηριστικός αριθμός ενός πάρτι.
- active: Ο επιθετικός προσδιορισμός αν ένα πάρτι είναι ενεργό(true) ή όχι(false).
- participants: Οι συμμετέχοντες του πάρτι.
- theme: Η θεματική ενότητα του πάρτι.
- location: Η τοποθεσία του πάρτι.
- date: Η ημερομηνία εκκίνησης του πάρτι.
- host: Ο διοργανωτής χρήστης του πάρτι.
- capacity: Η μέγιστη χωρητικότητα του πάρτι.
- rating: Η βαθμολογία του πάρτι.



## Μέθοδοι του πάρτι:

- Party(...): Ο constructor της κλάσης όταν δημιουργείται ένα πάρτι μέσω του HostPage.
- setPartyID(partyID:int): Θέτει τον μοναδικό χαρακτηριστικό αριθμό του πάρτι.
- getPartyID(): Επιστρέφει τον μοναδικό χαρακτηριστικό αριθμό του πάρτι.
- setActive(active:boolean): Θέτει τον επιθετικό προσδιορισμό αν ένα πάρτι είναι ενεργό.
- getActive(): Επιστρέφει τον επιθετικό προσδιορισμό αν ένα πάρτι είναι ενεργό.
- setParticipants(participants:list<userID>): Θέτει τους συμμετέχοντες του πάρτι.
- getParticipants(): Επιστρέφει τους συμμετέχοντες του πάρτι.
- setTheme(theme:string): Θέτει την θεματική ενότητα του πάρτι.
- getTheme(): Επιστρέφει την θεματική ενότητα του πάρτι.
- setLocation(location:list<float>): Θέτει την τοποθεσία του πάρτι.
- getLocation(): Επιστρέφει την τοποθεσία του πάρτι.
- setDate(date:string): Θέτει την ημερομηνία εκκίνησης του πάρτι.
- getData(): Επιστρέφει την ημερομηνία εκκίνησης του πάρτι.
- setHost(host:userID): Θέτει τον διοργανωτή χρήστη του πάρτι.
- getHost(): Επιστρέφει τον διοργανωτή χρήστη του πάρτι.
- setCapacity(capacity:int): Θέτει την χωρητικότητα του πάρτι.
- getCapacity(): Επιστρέφει την χωρητικότητα του πάρτι.
- setRating(in rating:float): Θέτει την βαθμολογία του πάρτι.
- getRating(): Επιστρέφει την βαθμολογία του πάρτι.

## Entity User

«Entity» User
-email: string -username: string -password: string -userID: int -joinedParties: list<partyID> -friends: list<userID> -rating: float -position: list<float> -age: int -gender: string
+User(email: string, username: string, password: string, age: int, gender: string): void +setEmail(email: string): void +getEmail(): string +setUsername(username: string): void +getUsername(): string +setPassword(password: string): void +getPassword(): string +setUserID(userID: int): void +getUserID(): int +setJoinedParties(joinedParties: list<partyID>): void +getJoinedParties(): list<partyID> +setFriends(friends: list<userID>): void +getFriends(): list<userID> +setRating(rating: float): void +getRating(): float +setPosition(position: list<float>): void +getPosition(): list<float> +setAge(age: int): void +getAge(): int +setGender(gender: string): void +getGender(): string



Ομοίως, η κλάση αυτή περιέχει τα χαρακτηριστικά και τις μεθόδους για την πλήρη ανάλυση της οντότητας του χρήστη.

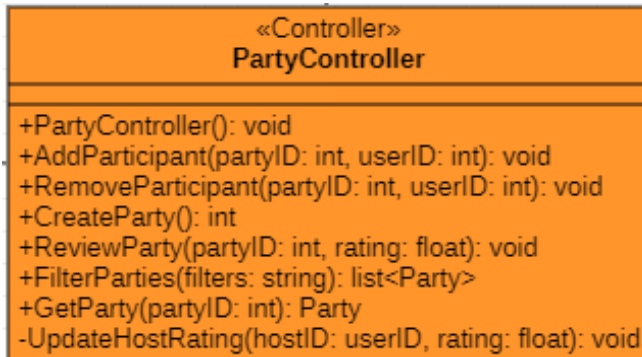
Χαρακτηριστικά του χρήστη:

- email: Η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.
- username: Το ψευδώνυμο του χρήστη.
- password: Ο κωδικός πρόσβασης του χρήστη.
- userID: Ο μοναδικός χαρακτηριστικός αριθμός ενός χρήστη.
- joinedParties: Τα συμμετέχοντα πάρτι του χρήστη.
- friends: Οι φίλοι του χρήστη.
- rating: Η βαθμολογία του χρήστη.
- position: Η στιγμιαία τοποθεσία του χρήστη.
- age: Η ηλικία του χρήστη.
- gender: Το κοινωνικό φύλο του χρήστη.

Μέθοδοι του χρήστη:

- User(...): Ο constructor της κλάσης όταν δημιουργείται ένας χρήστης.
- setEmail(in email:string): Θέτει την διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.
- getEmail(): Επιστρέφει την διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.
- setUsername(in username:string): Θέτει το ψευδώνυμο του χρήστη.
- getUsername(): Επιστρέφει το ψευδώνυμο του χρήστη.
- setPassword(in password:string): Θέτει τον κωδικό πρόσβασης του χρήστη.
- getPassword(): Επιστρέφει τον κωδικό πρόσβασης του χρήστη.
- setUserID(in userID:int): Θέτει τον μοναδικό χαρακτηριστικό αριθμό του χρήστη.
- getUserID(): Επιστρέφει τον μοναδικό χαρακτηριστικό αριθμό του χρήστη.
- setJoinedParties(in joinedParties:list<partyID>): Θέτει τα συμμετέχοντα πάρτι του χρήστη.
- getJoinedParties(): Επιστρέφει τα συμμετέχοντα πάρτι του χρήστη.
- setFriends(in friends:list<userID>): Θέτει τους φίλους του χρήστη.
- getFriends(): Επιστρέφει τους φίλους του χρήστη.
- setRating(in rating:float): Θέτει την βαθμολογία του χρήστη.
- getRating(): Επιστρέφει την βαθμολογία του χρήστη.
- setPosition(in position:list<float>): Θέτει την στιγμιαία τοποθεσία του χρήστη.
- getPosition(): Επιστρέφει την στιγμιαία τοποθεσία του χρήστη.
- setAge(in age:int): Θέτει την ηλικία του χρήστη.
- getAge(): Επιστρέφει την ηλικία του χρήστη.
- setGender(in gender:string): Θέτει το κοινωνικό φύλο του χρήστη.
- getGender(): Επιστρέφει το κοινωνικό φύλο του χρήστη.

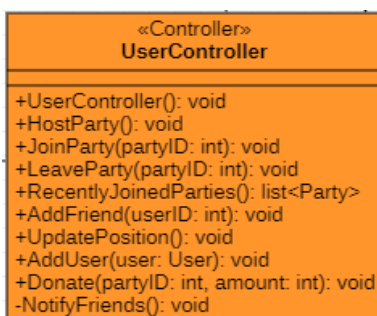


**Controller PartyController**

Η κλάση αυτή αποτελεί τον ελεγκτή που είναι υπεύθυνος για την διαχείριση κλήσεων από ελεγκτές γραφικής περιήγησης και την συσχέτιση αυτών των κλήσεων με ήδη υπάρχοντες οντότητες πάρτι στην βάση δεδομένων.

Μέθοδοι της κλάσης:

- PartyController(): Ο default constructor της κλάσης.
- AddParticipant(partyID: int, userID: int): Παίρνει σαν όρισμα το partyID και το userID προκειμένου να προσθέσει συμμετέχοντα στο πάρτι.
- RemoveParticipant(partyID: int, userID: int): Παίρνει σαν όρισμα το partyID και το userID προκειμένου να διαγράψει ήδη συμμετέχοντα από πάρτι.
- CreateParty(): Δημιουργεί ένα πάρτι και επιστρέφει έναν ακέραιο που αντιστοιχεί στο partyID.
- ReviewParty(partyID: int, rating: float): Υπολογίζει εσωτερικά την νέα βαθμολογία ενός πάρτι με συγκεκριμένο partyID συνυπολογίζοντας το νέο rating που έχει δεχτεί το πάρτι αυτό.
- FilterParties(filters: string): Παίρνει σαν όρισμα ένα string βάσει του οποίου ψάχνει στην βάση δεδομένων των πάρτι και ταξινομεί αναλόγως τα πάρτι προκειμένου να επιστρέψει μια λίστα από Party.
- GetParty(partyID): Επιστρέφει μια οντότητα Party βάσει ορίσματος partyID.
- UpdateHostRating(hostID: userID, rating: float): Ιδιωτική μέθοδος, η οποία καλείται εσωτερικά της ReviewParty(), και επανυπολογίζεται ο νέος μέσος όρος βαθμολογίας του χρήστη.

**Controller UserController**

Ομοίως, η κλάση αυτή αποτελεί τον ελεγκτή που είναι υπεύθυνος για την διαχείριση κλήσεων από ελεγκτές γραφικής περιήγησης και την συσχέτιση αυτών των κλήσεων με ήδη υπάρχοντες οντότητες χρηστών στην βάση δεδομένων.

Μέθοδοι της κλάσης:

- UserController(): Ο default constructor της κλάσης.
- HostParty(): Αποτελεί την μέθοδο που ένας χρήστης κατά την δημιουργία ενός πάρτι αυτοκυρύσσεται ως host αυτού αντιστοιχίζοντας το userID αυτού με το hostID του πάρτι.



- `JoinParty(partyID: int)`: Η μέθοδος αυτή βασιζόμενη στο όρισμα `partyID` κηρύσσει τον χρήστη ως συμμετέχων στο αντίστοιχο `Party`.
- `LeaveParty(partyID: int)`: Η μέθοδος αυτή χρησιμεύει για όταν ο χρήστης αποχωρεί από ένα συγκεκριμένο πάρτι, το οποίο έχει το `partyID`, και δεν ανήκει πλέον στους συμμετέχοντες.
- `RecentlyJoinedParties()`: Επιστρέφει την λίστα των πάρτι στα οποία έχει συμμετάσχει πρόσφατα ο χρήστης.
- `AddFriend(userID: int)`: Προσθέτει στον χρήστη έναν φίλο με `userID` αυτό του ορίσματος.
- `UpdatePosition()`: Ανανεώνει την τωρινή γεωγραφική τοποθεσία του χρήστη.
- `AddUser(User)`: Η μέθοδος αυτή καλείται για την προσθήκη της οντότητας `User` στην βάση δεδομένων.
- `Donate(partyID: int, amount:int)`: Χρήματα ποσότητας `amount` χαρίζονται από τον χρήστη σχετικά με το πάρτι που έχει `partyID`.
- `NotifyFriends()`: Η ιδιωτική αυτή μέθοδος καλείται εσωτερικά της `JoinParty()` και είναι υπεύθυνη για την ειδοποίηση των φίλων του χρήστη όταν αυτός συμμετέχει σε ένα πάρτι.

### Boundary PartiesDBProxy

«Boundary» PartiesDBProxy
+PartiesDBProxy(): void +LoadParties(): list<Party> +StoreParty(Party): void

Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με την βάση δεδομένων των πάρτι, καθώς μέσω αυτής πραγματοποιείται η αποθήκευση και η άντληση οντοτήτων πάρτι.

Μέθοδοι της κλάσης:

- `PartiesDBProxy()`: Ο default constructor της κλάσης.
- `LoadParties()`: Φορτώνει από την βάση δεδομένων και επιστρέφει μια λίστα από πάρτι.
- `StoreParty(Party)`: Αποθηκεύει ένα πάρτι στην βάση δεδομένων των πάρτι.

### Boundary UserDBProxy

«Boundary» UserDBProxy
+UserDBProxy(): void +LoadUser(userID: int, username: string): User +StoreUser(user: User): void

Ομοίως, αυτή η κλάση χρησιμοποιείται για την επικοινωνία με την βάση δεδομένων των πάρτι για την πραγματοποίηση αποθήκευσης και άντλησης οντοτήτων χρήστη.

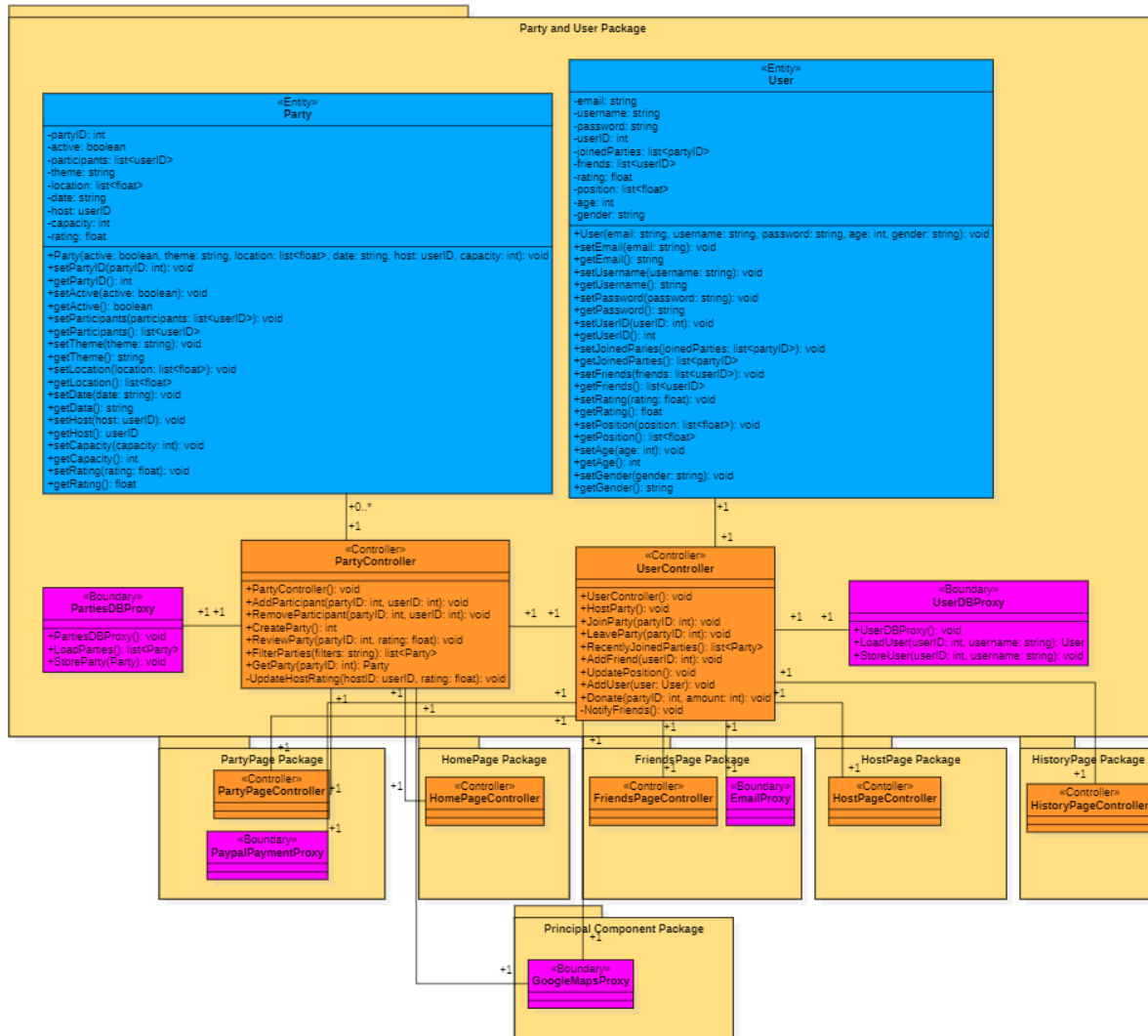
Μέθοδοι της κλάσης:

- `UserDBProxy()`: Ο default constructor της κλάσης.
- `LoadUser(userID:int, username:string)`: Αναζητεί στην βάση δεδομένων χρηστών βασισμένη στα ορίσματα `userID` και `username` με σκοπό την επιστροφή ενός χρήστη.



- StoreUser(user: User): Αποθηκεύει έναν χρήστη στην βάση δεδομένων των χρηστών.

#### 4.1.1.2 Διαγράμματα κλάσεων πακέτου Party and User



#### 4.1.3 Πακέτο Principal Component

Το πακέτο αυτό αναφέρεται στην δυνατότητα του χρήστη να βλέπει τα προτεινόμενα πάρτι σε απόσταση 4Km και να μπορεί να εφαρμόσει ενέργειες όπως η συμμετοχή σε πάρτι (Join) , δωρεά στον

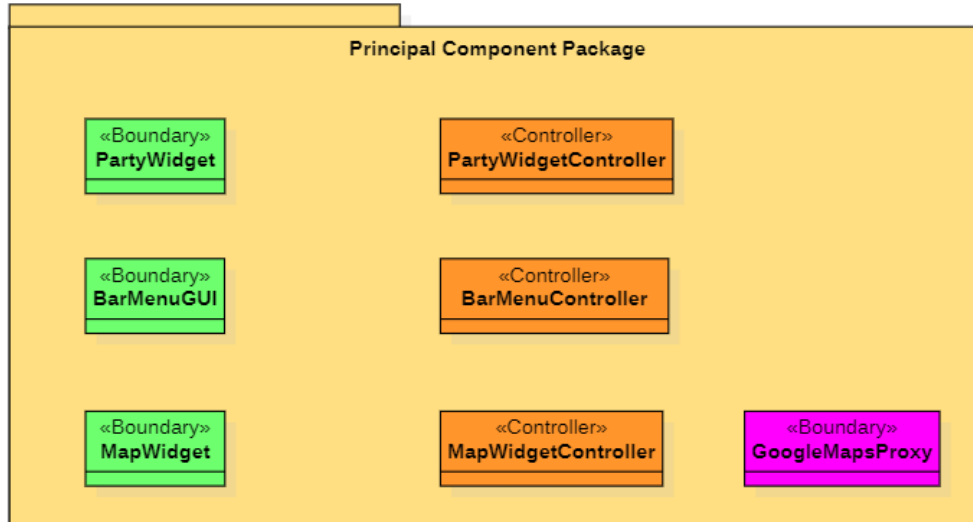
διοργανωτή του πάρτι (Donate) επιπλέον να μπορεί να βλέπει τους συμμετέχοντες του επιλεγμένου πάρτι.

Επιπρόσθετα το βασικό μενού του κάθε χρήστη που του δίνει την δυνατότητα εναλλαγής λειτουργίας όπως Home , Host, History, Friends.

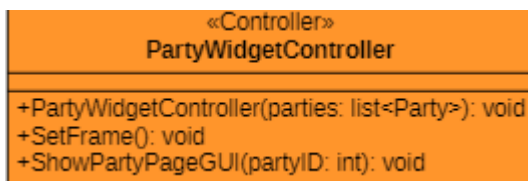




Τέλος περιλαμβάνει την λειτουργικότητα του χάρτη για την ενημερώσει τον τοποθετήσεων σε συνεργασία με το Google Maps Proxy.



#### 4.1.3.1 Ορισμού - Principal Component Controller PartyWidgetController



Ο ελεγκτής PartyWidgetController περιέχει τις μεθόδους για την εμφάνιση των πάρτι.

Μέθοδοι της κλάσης :

- PartyWidgetController(parties: list<Party>): Constructor της κλάσης που έχει ως όρισμα μια λίστα από πάρτι.
- SetFrame(): Η μέθοδος αυτή τοποθετεί το ένα πάρτι κάτω από το άλλο.
- ShowPartyPageUI(partyID:int): Η μέθοδος αυτή χρησιμοποιείται όταν ο χρήστης επιλέγει ένα προτεινόμενο πάρτι και μεταβαίνει στο PartyPage όπου λέει τις λεπτομέρειες του επιλεγμένου πάρτι.



## Boundary PartyWidget

«Boundary» PartyWidget
-lblTittle: label -lblTheme: label -lblLocation: label -lblDateTime: label -lblParticipants: label -iconUser: icon -lblUserName: label -lblRating: label -lblCapacity: label -iconStickfigure: icon -iconStar: icon -partyID: int +btnDirect: button +isDirectable: boolean +PartyWidget(Party): void +BtnDirect(): void

Η κλάση αυτή περιέχει τα βασικά στοιχεία του κάθε προτεινόμενου πάρτι στην οθόνη του χρήστη.

Χαρακτηριστικά της κλάσης :

- lblTittle: Ένα αλφαριθμητικό με τον τίτλο του πάρτι.
- lblTheme: Ένα αλφαριθμητικό με το θέμα του πάρτι.
- lblLocation: Ένα αλφαριθμητικό με την τοποθεσία του πάρτι.
- lblDateTime: Ένα αλφαριθμητικό με την ημερομηνία και ώρα διεξαγωγής του πάρτι.
- lblParticipants: Ένα αλφαριθμητικό με τους συμμετέχοντες χρήστες.
- iconUser: Είναι το εικονίδιο του χρήστη.
- lblUserName: Ένα αλφαριθμητικό με το όνομα του χρήστη π.χ. “KehagiasFan”.
- lblRating: Ένα αλφαριθμητικό με την βαθμολογία του χρήστη.
- lblCapacity: Ένα αλφαριθμητικό με τους συμμετέχοντες χρήστες σε σύγκριση με τους μέγιστους χρήστες.
- iconStickfigure: Είναι ένα εικονίδιο με το σχήμα ενός ανθρώπου.
- iconStar: Είναι ένα εικονίδιο με το σχήμα ενός αστεριού.
- partyID: Είναι ο μοναδικός αριθμός που αντιστοιχεί στο συγκεκριμένο πάρτι.
- btnDirect: Είναι ένα Button ανακατεύθυνσης.
- isDirectable: Είναι μια μεταβλητή ψευδούς/αληθούς για την εκπλήρωση της μετάβασης του χρήστη.

Μέθοδοι της κλάσης :

- PartyWidget(Party): Constructor της κλάσης που έχει ως όρισμα ένα πάρτι για την δημιουργία του αντικειμένου.
- BtnDirect(): Η μέθοδος αυτή ανακατευθύνει τον χρήστη στην επιλογή ενός πάρτι.

**Controller BarMenuController**

«Controller» BarMenuController
+BarMenuController(): void +ShowHomePageGUI(): void +ShowHostPageGUI(): void +ShowHistoryPageGUI(): void +ShowFriendsPageGUI(): void +ShowBarMenuGUI(): void

Ο ελεγκτής BarMenuController περιέχει την μέθοδο για την εμφάνιση του μενού της εφαρμογής και τις μεθόδους μετάβασης.

Μέθοδοι της κλάσης :

- BarMenuController(): Constructor της κλάσης.
- ShowHomePageGUI(): Η μέθοδος αυτή ανακατευθύνει τον χρήστη στο Home Page.
- ShowHostPageGUI(): Η μέθοδος αυτή ανακατευθύνει τον χρήστη στο Host Page.
- ShowHistoryPageGUI(): Η μέθοδος αυτή ανακατευθύνει τον χρήστη στο History Page.
- ShowFriendsPageGUI(): Η μέθοδος αυτή ανακατευθύνει τον χρήστη στο Friends Page.
- ShowBarMenuGUI(): Η μέθοδος αυτή χρησιμοποιείται για την εμφάνιση του μενού στον χρήστη.

**Boundary BarMenuGUI**

«Boundary» BarMenuGUI
-btnHome: button -btnHost: button -btnHistory: button -btnFriends: button -menuBar: shape
+BarMenuGUI(): void +BtnHome(): void +BtnHost(): void +BtnHistory(): void +BtnFriends(): void

Η κλάση αυτή περιέχει τα στοιχεία του μενού και τις μεθόδους μετάβασης.

Χαρακτηριστικά της κλάσης :

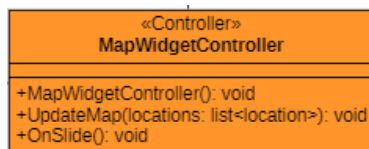
- btnHome: Ένα κουμπί για την μετάβαση στο Home Page.
- btnHost: Ένα κουμπί για την μετάβαση στο Host Page.
- btnHistory: Ένα κουμπί για την μετάβαση στο History Page.
- btnFriends: Ένα κουμπί για την μετάβαση στο Friends Page.
- menuBar: Είναι το περίγραμμα του μενού.



Μέθοδοι της κλάσης:

- BarMenuGUI(): default constructor.
- BtnHome(): συνάρτηση που καλεί την συνάρτηση ShowHomePageGUI() του controller BarMenuController για την μετάβαση του χρήστη στην σελίδα Home.
- BtnHost(): συνάρτηση που καλεί την συνάρτηση ShowHostPageGUI() του controller. BarMenuController για την μετάβαση του χρήστη στην σελίδα Host.
- BtnHistory(): συνάρτηση που καλεί την συνάρτηση ShowHistoryPageGUI() του controller. BarMenuController για την μετάβαση του χρήστη στην σελίδα History.
- BtnFriends(): συνάρτηση που καλεί την συνάρτηση ShowFriendsPageGUI() του controller. BarMenuController για την μετάβαση του χρήστη στην σελίδα Friends.

### Controller MapWidgetController

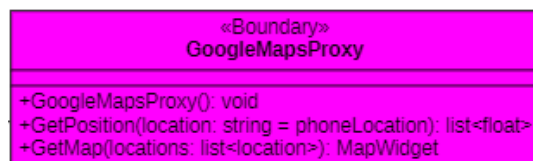


Ο ελεγκτής MapWidgetController περιέχει τις μεθόδους για την διαχείριση του χάρτη με την βοήθεια του GoogleMapsProxy.

Μέθοδοι της κλάσης:

- MapWidgetController(): default constructor.
- UpdateMap(locations : list<location> ): Η συνάρτηση που παίρνει ως όρισμα μια λίστα από σημεία στο χάρτη με τα πάρτι και μετά καλεί την συνάρτηση GetMap(locations: list<location>) και επιστρέφει σημεία στον χάρτη με την τοποθεσία του κάθε πάρτι.
- OnSlide(): Συνάρτηση που χρησιμοποιείται κατά την κίνηση του χάρτη για την εμφάνιση των νέων σημείων.

### Boundary GoogleMapsProxy



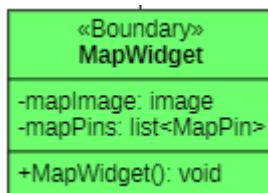
Η κλάση αυτή περιέχει το έτοιμο Google Map API και τις μεθόδους για την διαχείριση των σημείων.



Μέθοδοι της κλάσης:

- MapWidgetController(): default constructor.
- GetPosition(location: string = phoneLocation): Συνάρτηση που παίρνει ως όρισμα την τοποθεσία του κινητού του χρήστη και τα επιστρέφει σε μια λίστα σημείων.
- GetMap(locations:list<location>): Συνάρτηση που παίρνει ως όρισμα μια λίστα από σημεία και τα επιστρέφει στον χάρτη με τα Pin.

### Boundary MapWidget



Η κλάση αυτή περιέχει τον χάρτη

Χαρακτηριστικά της κλάσης :

- mapImage: Είναι η εικόνα που μας δίνει το GoogleMapsProxy για τον χάρτη.
- mapPins: Είναι μια λίστα από σημεία που εμφανίζονται στον χάρτη.

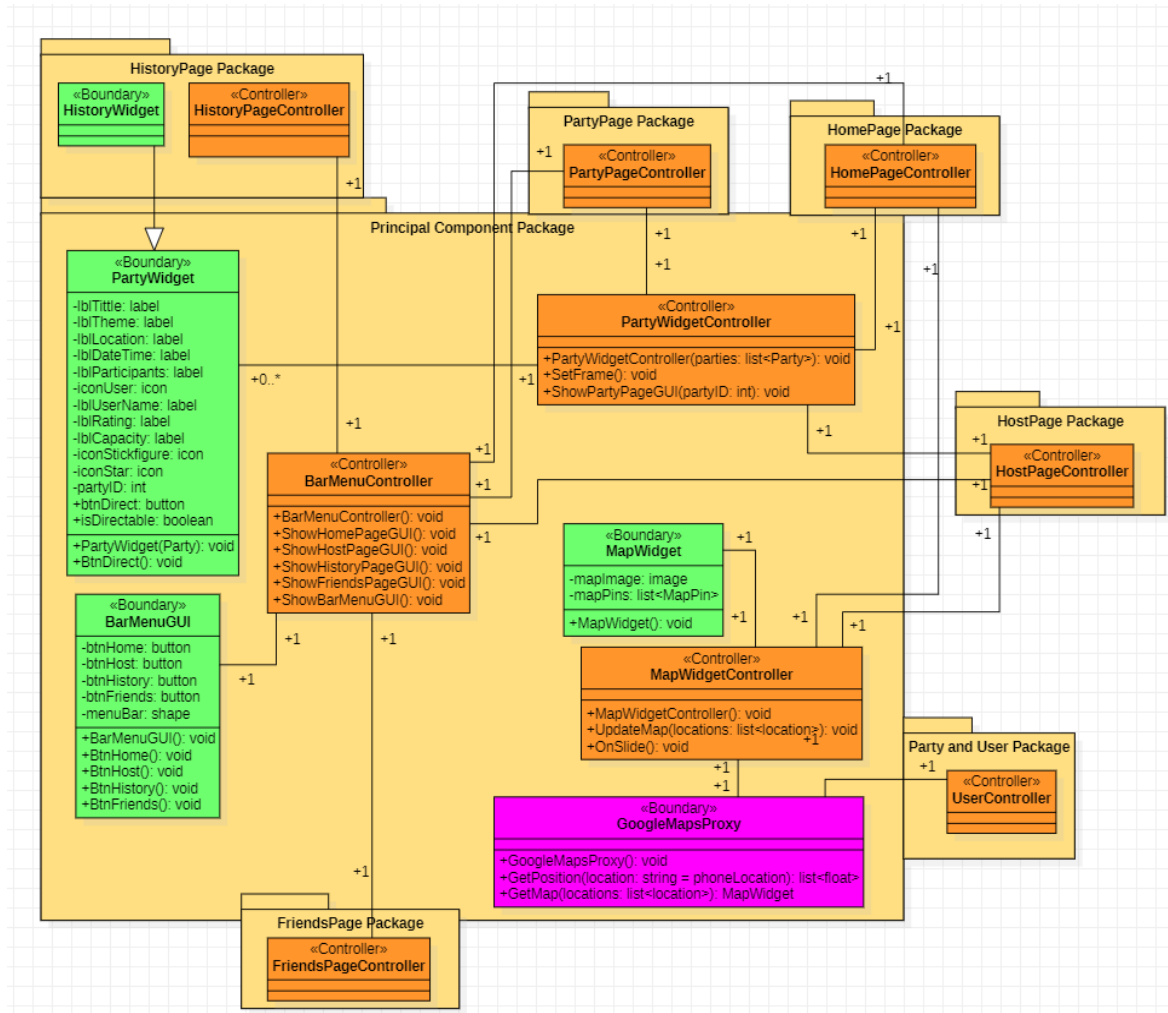
Μέθοδοι της κλάσης:

- MapWidget(): default constructor.



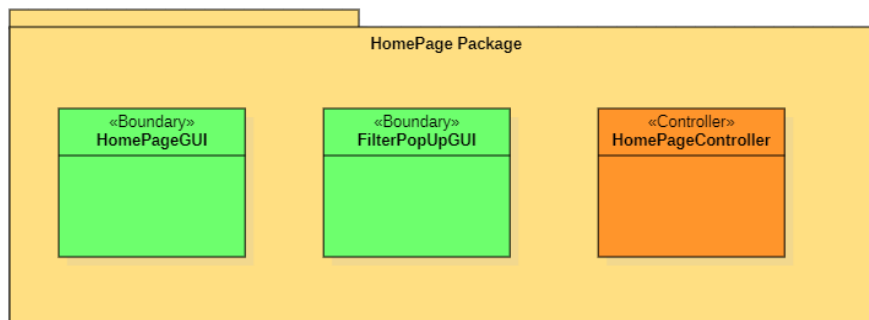


#### 4.1.3.2 Διαγράμματα κλάσεων πακέτου Principal Component



#### 4.1.4 Πακέτο Home Page

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με το “home page” της εφαρμογής. Συγκεκριμένα περιλαμβάνει την κλάση της γραφικής διεπαφής χρήστη του “home page”, επιπλέον περιλαμβάνει τον ελεγκτή αυτής, και τέλος, την γραφική διεπαφή χρήστη του φιλτραρίσματος πάρτι η οποία υλοποιείται σε μορφή “pop-up”. Ο ελεγκτής της γραφικής διεπαφής χρήστη του “home page” αποτελεί επίσης ελεγκτή της γραφικής διεπαφής χρήστη του φιλτραρίσματος.





#### 4.1.4.1 Ορισμοί - Πακέτο HomePage

##### Boundary HomePageGUI

«Boundary» HomePageGUI
+btnDown: button -lblPartyUp: label -btnFilter: button -lblWelcome: label -partyWidgetBackground: shape
+HomePageGUI(): void +BtnFilter(): void +BtnDown(): void

Η κλάση αυτή περιλαμβάνει όλα τα αντικείμενα διεπαφής χρήστη που απαιτούνται για να σχεδιαστεί η σελίδα “home page”.

Ιδιότητες κλάσης:

- btnDown: Είναι το κουμπί που χρησιμοποιείται για την απόκρυψη των πάρτι με σκοπό την πλήρη εμφάνιση του χάρτη.
- lblPartyUp: Είναι η ταμπέλα στο πάνω αριστερό μέρος της οθόνης στο “home page” που έχει σαν κείμενο το όνομα της εφαρμογής.
- btnFilter: Είναι το κουμπί που χρησιμοποιείται για την εμφάνιση της διεπαφής χρήστη του φιλτραρίσματος. Βρίσκεται στο πάνω δεξιό μέρος της οθόνης.
- lblWelcome: Είναι η ταμπέλα που εμφανίζεται στο “home page” πάνω από τα πάρτι και χρησιμοποιείται για να καλωσορίσει τους χρήστες στην εφαρμογή.
- partyWidgetBackground: Αποτελείται από το πλαίσιο πάνω στο οποίο εμφανίζονται τα πάρτι.

Μέθοδοι κλάσης:

- HomePageGUI(): Αποτελεί την μέθοδο δόμησης της σελίδας “home page”.
- BtnFilter(): Αποτελεί την συνάρτηση η οποία εκτελείται όταν πατηθεί το κουμπί btnFilter. Μεταφέρει τον έλεγχο στον ελεγκτή της σελίδας “home page” και συγκεκριμένα καλεί την συνάρτηση αυτού, ShowFilterPageGUI().
- BtnDown(): Αποτελεί την συνάρτηση η οποία εκτελείται όταν πατηθεί το κουμπί btnDown. Μεταφέρει τον έλεγχο στον ελεγκτή της σελίδας “home page” και συγκεκριμένα καλεί την συνάρτηση αυτού, HidePartyWidgets().

**Boundary FilterPopUpGUI**

«Boundary» FilterPopUpGUI
-background: shape -lblFilter: label -ddRating: dropDownMenu -ddDistance: dropDownMenu -ddTheme: dropDownMenu -ddCapacity: dropDownMenu -btnApply: button
+FilterPopUpGUI(): void +SetRating(rating: string): void +GetRating(): float +SetDistance(distance: string): void +GetDistance(): float +SetTheme(theme: string): void +GetTheme(): string +SetCapacity(capacity: string): void +GetCapacity(): int +BtnApply(): void

Η κλάση αυτή αποτελεί την “pop-up” γραφική διεπαφή χρήστη του φιλτραρίσματος των πάρτι.

Ιδιότητες κλάσης:

- background: Αποτελεί το πλαίσιο πάνω στο οποίο εμφανίζονται τα φίλτρα.
- lblFilter: Είναι η ταμπέλα η οποία εμφανίζεται πάνω από τα φίλτρα και έχει σαν κείμενο, όπως υποδηλώνει και το όνομά της, “Filters”.
- ddRating: Είναι το “drop down menu” για την επιλογή του φίλτρου που σχετίζεται με την εμφάνιση των πάρτι, των οποίων οι διοργανωτές έχουν βαθμολογία μεγαλύτερη ή ίση της επιλεχθείσας.
- ddDistance: Είναι το “drop down menu” για την επιλογή του φίλτρου που σχετίζεται με την εμφάνιση των πάρτι, η απόσταση των οποίων από την τρέχουσα τοποθεσία χρήστη δεν είναι μεγαλύτερη της επιλεχθείσας.
- ddTheme: Είναι το “drop down menu” για την επιλογή του φίλτρου που σχετίζεται με την εμφάνιση των πάρτι, των οποίων το θέμα είναι ίδιο με το επιλεχθέν.
- ddCapacity: Είναι το “drop down menu” για την επιλογή του φίλτρου που σχετίζεται με την εμφάνιση των πάρτι, η χωρητικότητα των οποίων είναι μεγαλύτερη ή ίση της επιλεχθείσας.
- btnApply: Είναι το κουμπί που βρίσκεται κάτω από τα προαναφερθέντα φίλτρα και έχει σαν σκοπό την εφαρμογή αυτών.

Μέθοδοι κλάσης:

- FilterPopUpGUI(): Αποτελεί την μέθοδο δόμησης της γραφικής διεπαφής χρήστη του φιλτραρίσματος.
- SetRating(rating: string): Αποτελεί την συνάρτηση η οποία θέτει το κείμενο που θα εμφανίζεται στο “drop down menu” ddRating.
- GetRating(): Αποτελεί την συνάρτηση που επιστρέφει στην έξοδο την ελάχιστη βαθμολογία διοργανωτή που επέλεξε ο χρήστης, σε μορφή τύπου “float”.
- SetDistance(rating: string): Αποτελεί την συνάρτηση η οποία θέτει το κείμενο που θα εμφανίζεται στο “drop down menu” ddDistance.
- GetDistance(): Αποτελεί την συνάρτηση που επιστρέφει στην έξοδο την μέγιστη απόσταση των πάρτι από την τρέχουσα τοποθεσία του, σε μορφή τύπου “float”.
- SetTheme(rating: string): Αποτελεί την συνάρτηση η οποία θέτει το κείμενο που θα εμφανίζεται στο “drop down menu” ddTheme.



- **GetTheme():** Αποτελεί την συνάρτηση που επιστρέφει στην έξοδο το θέμα το οποίο επέλεξε ο χρήστης, σε μορφή τύπου “string”.
- **SetCapacity(rating: string):** Αποτελεί την συνάρτηση η οποία θέτει το κείμενο που θα εμφανίζεται στο “drop down menu” ddCapacity.
- **GetCapacity():** Αποτελεί την συνάρτηση που επιστρέφει στην έξοδο την ελάχιστη χωρητικότητα των πάρτι που θέλει ο χρήστης να εμφανιστούν, σε μορφή τύπου “int”.
- **BtnApply():** Αποτελεί την συνάρτηση που εκτελείται όταν πατηθεί το κουμπί btnApply. Μεταφέρει τον έλεγχο στον ελεγκτή του “HomePage” και συγκεκριμένα καλεί την συνάρτηση αυτού, ViewParties().

### Controller HomeController

«Controller» HomeController
+HomeController(): void +ViewParties(): void -FilterParties(rating: float, distance: float, theme: string, capacity: int): list<Party> +ScrollParties(): void +ShowFilterPageGUI(): void +HidePartyWidgets(): void

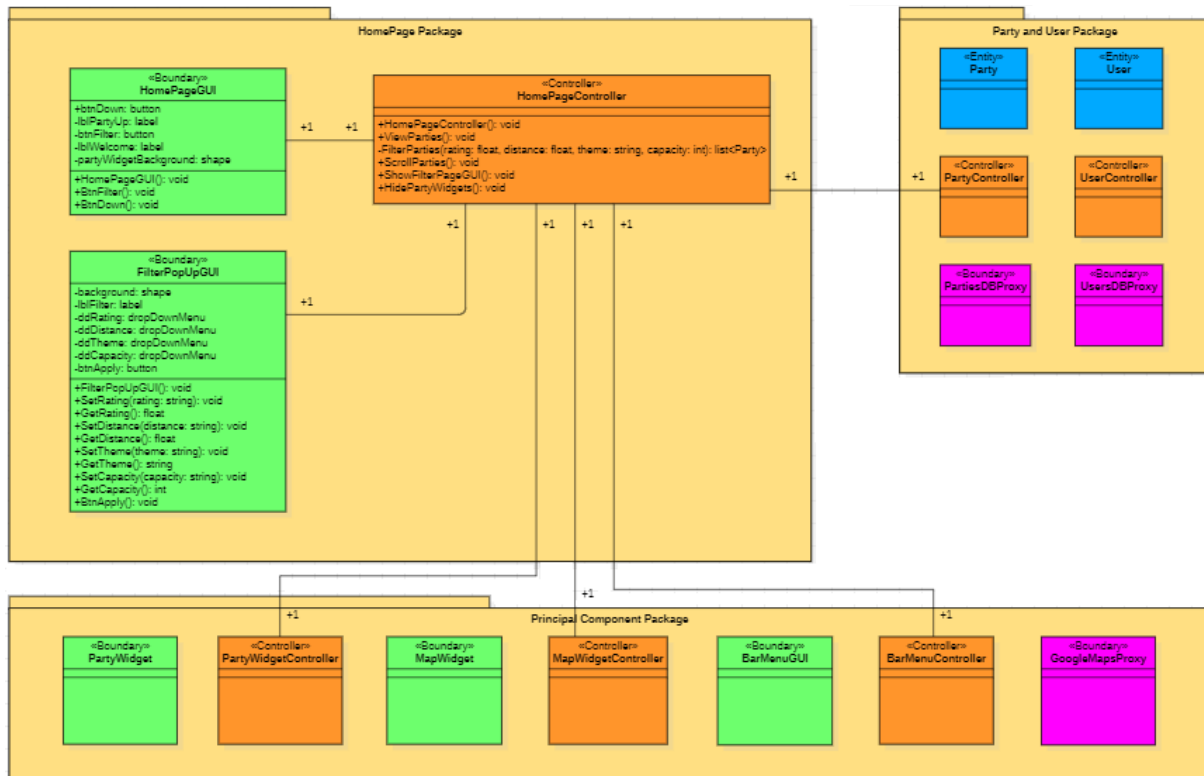
Η κλάση αυτή αποτελεί τον ελεγκτή του “HomePage” όπως και τον ελεγκτή της γραφικής διεπαφής χρήστη του φιλτραρίσματος.

Μέθοδοι κλάσης:

- **HomeController():** Αποτελεί την συνάρτηση δόμησης του ελεγκτή.
- **ViewParties():** Είναι η συνάρτηση η οποία είναι υπεύθυνη για την εμφάνιση των πάρτυ. Εσωτερικά καλεί την “private” συνάρτηση FilterParties(rating: float, distance: float, theme: string, capacity: int) και στην συνέχεια καλεί την συνάρτηση δόμησης της κλάσης PartyWidgetController με όρισμα την λίστα με πάρτι που επιστρέφει η FilterParties.
- **FilterParties(rating: float, distance: float, theme: string, capacity: int):** Αποτελεί την συνάρτηση η οποία είναι υπεύθυνη να επιστρέψει μία λίστα από πάρτι τα οποία ικανοποιούν τα φίλτρα που λαμβάνει σαν ορίσματα. Εσωτερικά καλεί την ομώνυμη συνάρτηση FilterParties(filters: string) του ελεγκτή PartyController η οποία επιστρέφει μία λίστα από αντικείμενα τύπου Party. Αυτή είναι και η λίστα που τελικά θα επιστρέψει η FilterParties του ελεγκτή HomeController.
- **ScrollParties():** Αποτελεί την συνάρτηση η οποία είναι υπεύθυνη για την μετακίνηση των διαθέσιμων πάρτυ στην σελίδα “home page” όταν ο χρήστης κάνει “scroll”.
- **ShowFilterPageGUI():** Αποτελεί την συνάρτηση η οποία είναι υπεύθυνη για την εμφάνιση της “pop-up” γραφικής διεπαφής χρήστη του φιλτραρίσματος.
- **HidePartyWidgets():** Αποτελεί την συνάρτηση η οποία είναι υπεύθυνη για την απόκρυψη των πάρτυ με στόχο την πλήρη εμφάνιση του χάρτη στην σελίδα “home page”.

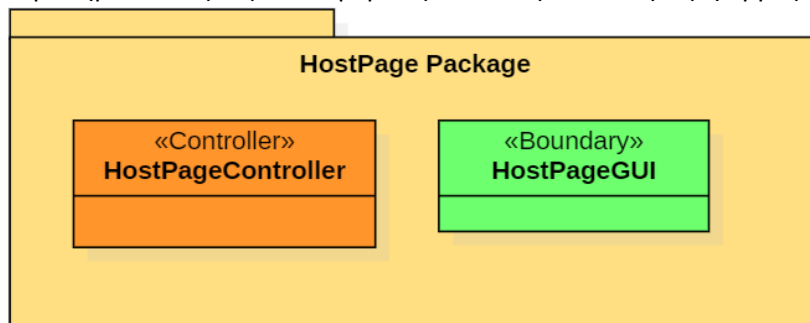


#### 4.1.4.2 Διαγράμματα κλάσεων - Πακέτο HomePage



#### 4.1.5 Πακέτο HostPage

Το πακέτο αυτό αναφέρεται στην δυνατότητα του χρήστη να δημοσιεύσει ένα πάρτι, συμπληρώνοντας τις λεπτομέρειες του στις κατάλληλες φόρμες και έπειτα κάνοντάς το publish.



##### 4.1.5.1 Ορισμού - Πακέτο HostPage

###### ControllerHostPageController



Ο ελεγκτής HostPageController περιέχει την μέθοδο για την δημοσίευση ενός πάρτι.



Μέθοδοι της κλάσης :

- `HostPageController()` : Constructor της κλάσης.
- `Publish(title:string, theme:string, capacity:string, date:string, time:string, address:string)` Η μέθοδος αυτή καλεί την αντίστοιχη μέθοδο του χρήστη για την δημιουργία ενός πάρτι.

### Boundary `HostPageGUI`

«Boundary» <b>HostPageGUI</b>
-btnPublish: button -txtTitle: textBox -ddTheme: dropDownMenu -txtCapacity: textBox -date: Calendar -time: Clock -txtAddress: textBox -lblHostParty: label -lblPreview: label
+HostPageGUI(): void +BtnPublish(): void +GetTitleTxt(): string +SetTitleTxt(txtMessage: string): void +GetAddressTxt(): string +SetAdressTxt(txtAddress: string): void +GetCapacityTxt(): string +SetCapacityTxt(txtCapacity: string): void +GetTheme(): string +SetTheme(theme: string): void +GetCalendarDate(): string +SetCalendar(calendar: string): void +GetTime(): string +SetTime(time: string): void

Η κλάση αυτή αντιπροσωπεύει την σελίδα δημοσίευσης πάρτι

Χαρακτηριστικά της κλάσης:

- `btnPublish`: κουμπί για την δημοσίευση του πάρτι.
- `txtTitle`: textbox για την εισαγωγή του τίτλου του πάρτι.
- `ddTheme`: drop down menu για την εισαγωγή θέματος του πάρτι.
- `txtCapacity`: textbox για την εισαγωγή του μέγιστου αριθμού συμμετεχόντων.
- `date`: εισαγωγή μέσω ημερολογίου της ημερομηνίας διεξαγωγής του πάρτι.
- `time`: εισαγωγή μέσω ρολογιού της ώρας διεξαγωγής του πάρτι.
- `txtAddress`: textBox για την εισαγωγή της διεύθυνσης του πάρτι.
- `lblHostParty`: κείμενο “Host a Party!”.
- `lblPreview`: κείμενο “Preview your post”.

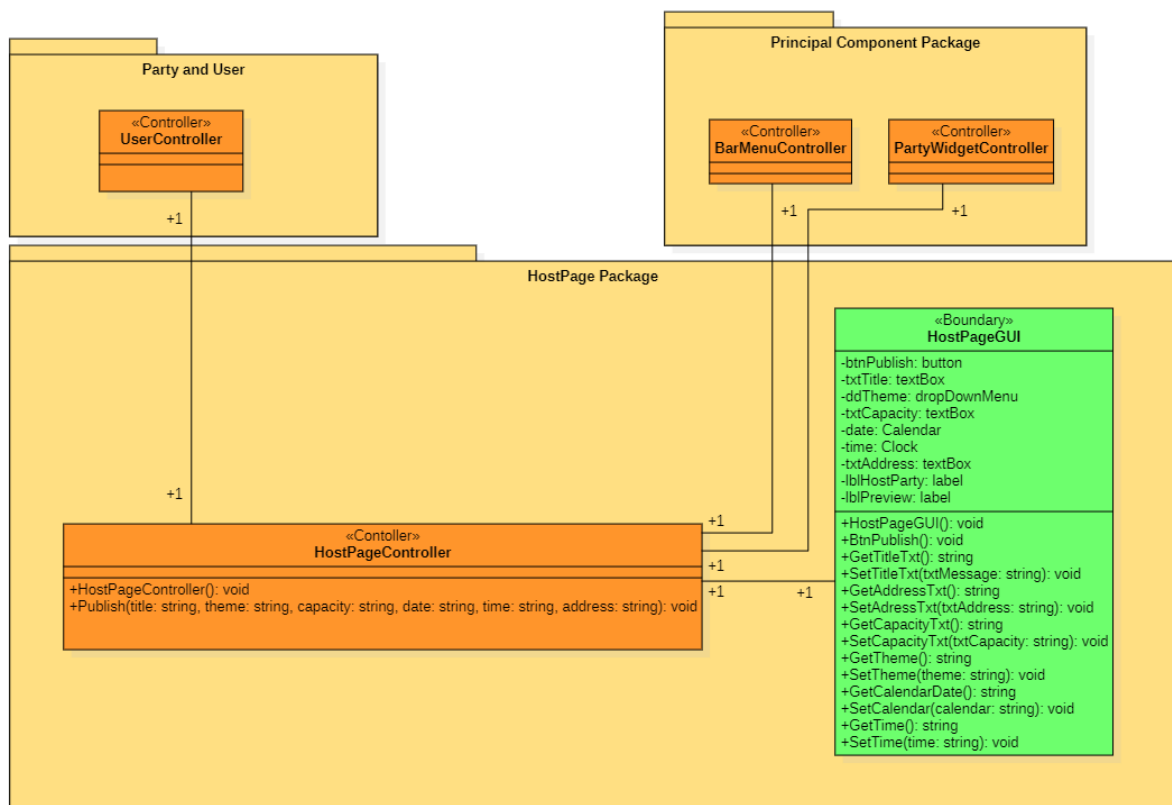
Μέθοδοι της κλάσης:

- `HostPageGUI()`: default constructor.
- `BtnPublish()`: συνάρτηση που καλεί την συνάρτηση `Publish` του controller `HostPageController` για την δημοσίευση του πάρτι.
- `GetTitleTxt()`: επιστρέφει τον τίτλο του πάρτι.



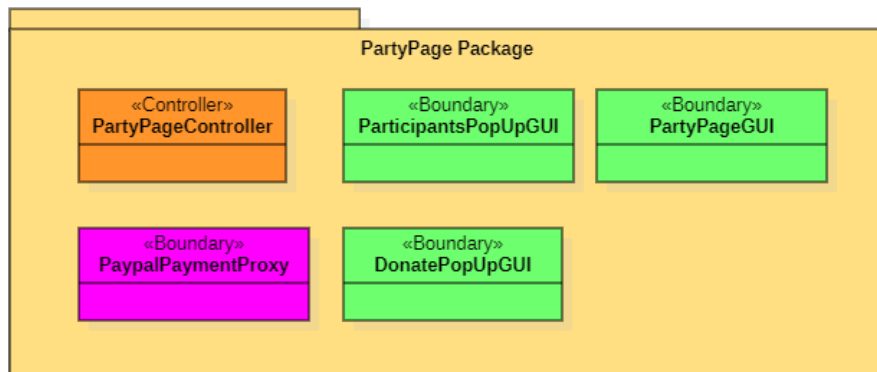
- `SetTitleTxt(txtMessage: string)`: θέτει τον τίτλο του πάρτι στο αρχικό “Title”.
- `GetAddressTxt()`: επιστρέφει την διεύθυνση του πάρτι.
- `SetAddressTxt(txtAddress: string)`: θέτει την διεύθυνση του πάρτι στην αρχική τιμή “Address”.
- `GetCapacity Txt()`: επιστρέφει τον μέγιστο αριθμό συμμετεχόντων.
- `SetCapacity Txt(txtCapacity: string)`: θέτει τον αριθμό συμμετεχόντων στην αρχική τιμή “Capacity”.
- `GetTheme()`: επιστρέφει την επιλογή της θεματολογίας.
- `SetTheme(theme: string)`: θέτει την θεματολογία του πάρτι στην αρχική τιμή Theme.
- `GetCalendarDate()`: επιστρέφει την ημερομηνία διεξαγωγής του πάρτι.
- `SetCalendar(calendar: string)`: θέτει στην ημερομηνία την αρχική τιμή “Date”.
- `GetTime()`: επιστρέφει την ώρα διεξαγωγής του πάρτι.
- `SetTime(time: string)`: θέτει την ώρα στην αρχική τιμή “Time”.

#### 4.1.5.2 Διαγράμματα κλάσεων πακέτου HostPage



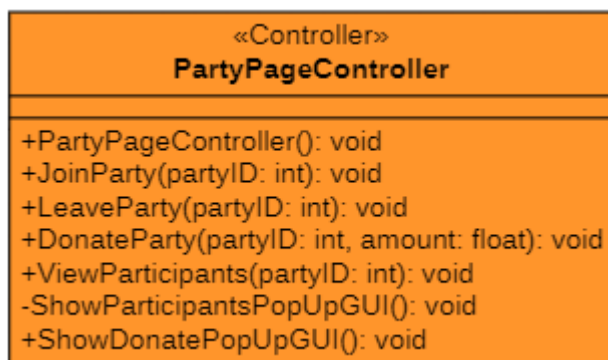


#### 4.1.6 Πακέτο PartyPage



##### 4.1.6.1 Ορισμοί - Πακέτο PartyPage

###### Controller PartyPageController



Η κλάση αυτή αποτελεί τον ελεγκτή που είναι υπεύθυνος για την λειτουργικότητα και εμφάνιση της σελίδας του πάρτι.

Μέθοδοι της Κλάσης:

- PartyPageController(): default constructor.
- JoinParty(partyID : int): η μέθοδος αυτή είναι υπεύθυνη για την συμμετοχή ενός χρήστη σε ένα πάρτι.
- LeaveParty(partyID: int): η μέθοδος αυτή είναι υπεύθυνη για την αποχώρηση ενός χρήστη από ένα πάρτι.
- DonateParty(partyID: int, amount: float): η μέθοδος αυτή υλοποιεί τη οικονομική συνεισφορά σε ένα πάρτι.
- ViewParticipants(partyID:int): μέθοδος για την προβολή των συμμετεχόντων σε ένα πάρτι. Ανατρέχει αρχικά στο database για την εύρεση των participants και έπειτα καλεί την ShowParticipantsPopUpGUI().
- ShowParticipantsPopUpGUI(): μέθοδος υπεύθυνη για την προβολή του παραθύρου με τους συμμετέχοντες ενός πάρτι.
- ShowDonatePopUpGUI(): μέθοδος υπεύθυνη για την προβολή του παραθύρου για την οικονομική συνεισφορά.





### Boundary PartyPageGUI

«Boundary» PartyPageGUI
-btnJoin: button -btnLeave: button -btnDonate: button -btnParticipants: button -lblPartyUp: label
+PartyPageGUI(): void +BtnJoin(): void +BtnLeave(): void +BtnDonate(): void +BtnParticipants(): void

Η κλάση αυτή αντιπροσωπεύει την σελίδα λεπτομερειών ενός πάρτι.

Χαρακτηριστικά της κλάσης:

- btnJoin : κουμπί για την συμμετοχή στο πάρτι.
- btnLeave: κουμπί για την αποχώρηση από πάρτι.
- btnDonate: κουμπί για την οικονομική συνεισφορά σε ένα πάρτι.
- btnParticipants: κουμπί για την προβολή των συμμετεχόντων .
- lblPartyUp: κείμενο “Party Up”.

Μέθοδοι της κλάσης:

- PartyPageGUI(): default constructor.
- BtnJoin(): μέθοδος που καλεί την JoinParty() του controller.
- BtnLeave(): μέθοδος που καλεί την LeaveParty() του controller.
- BtnDonate(): μέθοδος που καλεί την DonateParty() του controller.
- BtnParticipants(): μέθοδος που καλεί την ViewParticipants() του controller.

### Boundary DonatePopUpGUI

«Boundary» DonatePopUpGUI
-txtAmount: textBox -lblDonate: label -lblHelpHost: label -iconDollar: icon -btnPaypal: button -background: shape
+DonatePopUpGUI(): void +GetAmount(): float +BtnPaypal(): void

Η κλάση αυτή αντιπροσωπεύει το pop-up για την οικονομική συνεισφορά σε ένα πάρτι.



Χαρακτηριστικά της κλάσης:

- txtAmount: textbox για την εισαγωγή του χρηματικού ποσού για προσφορά.
- lblDonate: κείμενο "Donation to Host".
- lblHelpHost: κείμενο "Help the party organizer with a donation".
- iconDollar: εικονίδιο δολαρίου στο textbox.
- btnPaypal: κουμπί για την συναλλαγή μέσω PayPal.
- background: το πλαίσιο του pop-up.

Μέθοδοι της κλάσης:

- DonatePopUpGUI(): default constructor.
- GetAmount(): μέθοδος που επιστρέφει το ποσό που θέλει ο χρήστης να συνεισφέρει.
- BtnPaypal(): μέθοδος που καλεί την μέθοδο DonateParty(partyID: int, amount: float).

### Boundary ParticipantsPopUpGUI

«Boundary» ParticipantsPopUpGUI
-background: shape -lblParticipants: label -btnAdd: list<button> -usernames: list<string> -userIcon: list<icon> -userIDs: list<int>
+ParticipantsPopUpGUI(participants: list<User>): void +BtnAdd(userID: int): void

Η κλάση αυτή αντιπροσωπεύει το pop-up για την προβολή των συμμετεχόντων σε ένα πάρτι.

Χαρακτηριστικά της κλάσης:

- background: το πλαίσιο του pop-up.
- lblParticipants: κείμενο "Participants".
- btnAdd: κουμπί για την προσθήκη φίλου.
- usernames: τα ονόματα των χρηστών που συμμετέχουν στο πάρτι.
- userIcon: τα εικονίδια των χρηστών που συμμετέχουν στο πάρτι.
- userIDs: τα ids των χρηστών που συμμετέχουν στο πάρτι.

Μέθοδοι της κλάσης:

- ParticipantsPopUpGUI(participants: list<User>): default constructor.
- BtnAdd(): μέθοδος που καλείται με το πάτημα του κουμπιού btnAdd για την προσθήκη ενός φίλου.

### Boundary PaypalPaymentProxy

«Boundary» PaypalPaymentProxy
+PaypalPaymentProxy(): void +Donate(email: string, partyHostEmail: string, amount: int): void

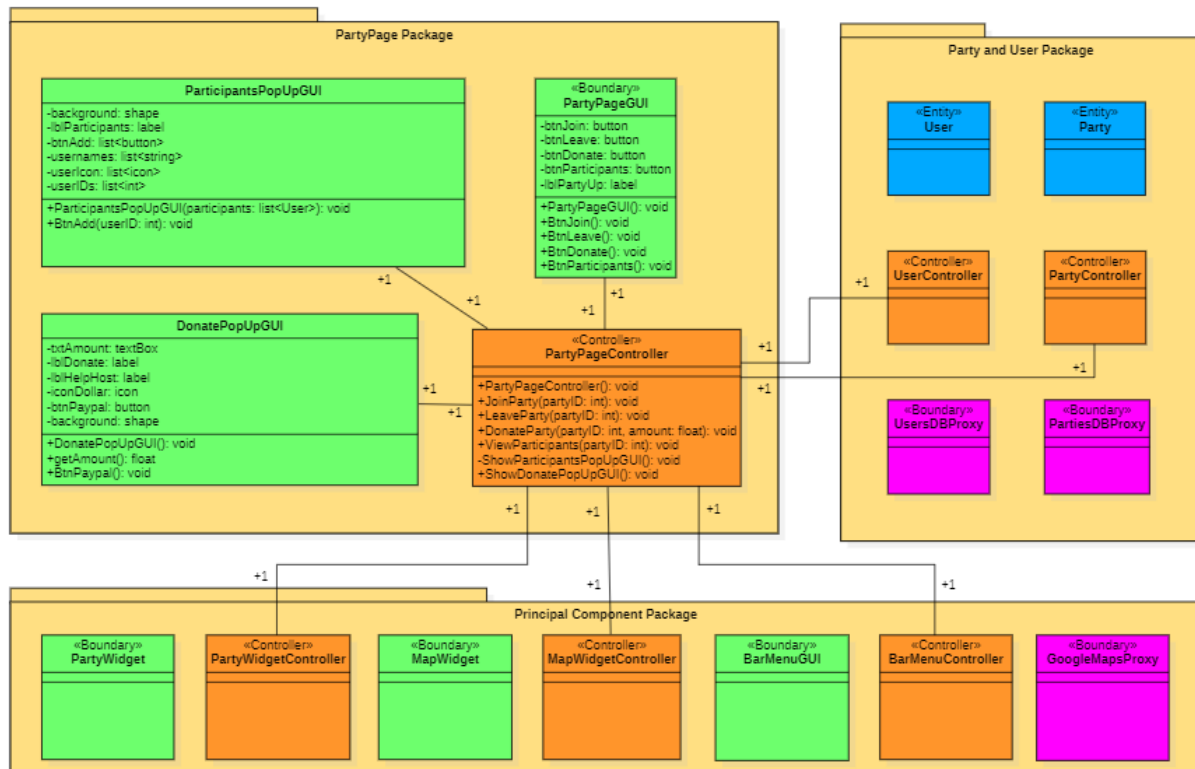
Η κλάση αυτή χρησιμοποιείται για την επικοινωνία με την υπηρεσία πληρωμής PayPal ώστε να υλοποιηθούν οι οικονομικές συνεισφορές σε πάρτι.



Μέθοδοι της κλάσης:

- PayPalPaymentProxy(): default constructor.
- Donate(userEmail:string, partyHostEmail: string, amount:int): μέθοδος για την διεξαγωγή της πληρωμής μέσω του API της PayPal χρησιμοποιώντας το email του χρήστη, το email του διοργανωτή και το ποσό της πληρωμής.

#### 4.1.6.1 Διαγράμματα κλάσεων πακέτου PartyPage

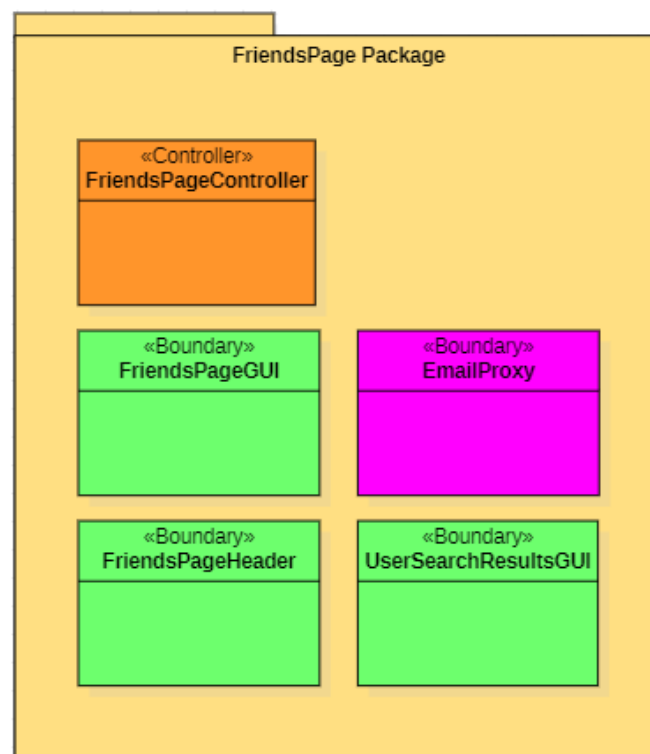




## 4.2 <Πακέτο(α) λεξιλογίου σεναρίων μέσης προτεραιότητας>

### 4.2.1 Πακέτο FriendsPage

Το πακέτο αυτό περιλαμβάνει τις κλάσεις που σχετίζονται με το “friends page” της εφαρμογής. Συγκεκριμένα περιλαμβάνει τις κλάσεις της γραφικής διεπαφής χρήστη της σελίδας φίλων, επιπλέον περιλαμβάνει τον ελεγκτή αυτών, και τέλος, μία “proxy” κλάση που σχετίζεται με την αποστολή email.



#### 4.2.1.1 Ορισμοί - Πακέτο FriendsPage

##### Boundary FriendsPageGUI

«Boundary» FriendsPageGUI
-usernames: list<string> -userIcons: list<icon> -lblFriendsList: label -background: shape
+FriendsPageGUI(friends: list<User>): void

Η κλάση αυτή περιλαμβάνει τα αντικείμενα διεπαφής χρήστη που απαιτούνται για να σχεδιαστεί μία λίστα φίλων.

Ιδιότητες κλάσης:

- usernames: Είναι μία λίστα που έχει σαν στοιχεία τα ονόματα των φίλων του χρήστη.
- userIcons: Είναι μία λίστα που έχει σαν στοιχεία τα εικονίδια (φωτογραφίες) των φίλων του χρήστη.



- `lblFriendsList`: Είναι η ταμπέλα με κείμενο “Friends List”. Ουσιαστικά υποδηλώνει ότι πρόκειται για μία λίστα φίλων.
- `background`: Είναι το πλαίσιο πάνω στο οποίο εμφανίζονται οι φίλοι του χρήστη.

Μέθοδοι κλάσης:

- `FriendsPageGUI(friends: list<User>)`: Αποτελεί την μέθοδο δόμησης της σελίδας “friends page”. Παίρνει σαν όρισμα μία λίστα με στοιχεία τύπου `User`.

### Boundary FilterPopUpGUI

«Boundary» FriendsPageHeader
+lblPartyUp: label +searchBar: textBox +headerBackground: shape
+FriendsPageHeader(): void +SetSearchBarText(text: string): void +GetSearchBarText(): string

Η κλάση αυτή αποτελεί την κεφαλίδα της γραφικής διεπαφής χρήστη της σελίδας φίλων.

Ιδιότητες κλάσης:

- `lblPartyUp`: Είναι η ταμπέλα στο πάνω αριστερά μέρος της σελίδας με κείμενο το όνομα της εφαρμογής.
- `searchBar`: Είναι η μπάρα αναζήτησης στο πάνω μέρος της σελίδας με την οποία ο χρήστης μπορεί να αναζητεί άλλους χρήστες.
- `headerBackground`: Είναι το πλαίσιο πάνω στο οποίο βρίσκονται τα δύο προαναφερθέντα αντικείμενα.

Μέθοδοι κλάσης:

- `FriendsPageHeader()`: Αποτελεί την μέθοδο δόμησης της κεφαλίδας της σελίδας φίλων του χρήστη.
- `SetSearchBarText(text: string)`: Αποτελεί την συνάρτηση η οποία θέτει το κείμενο που θα εμφανίζεται στην μπάρα αναζήτησης `searchBar`.
- `GetSeatchBarText()`: Αποτελεί την συνάρτηση που επιστρέφει στην έξοδο το κείμενο που έγραψε ο χρήστης στην μπάρα αναζήτησης `searchBar`.

### Boundary UserSearchResultsGUI

«Boundary» UserSearchResultsGUI
-lblSearchResults: label -participantsBackground: shape -btnAdd: list<button> -usernames: list<string> -userIcon: list<icon> -userIDs: list<int>
+UserSearchResultsGUI(results: list<User>): void +BtnAdd(userID: int): void



Η κλάση αυτή περιλαμβάνει τα αντικείμενα διεπαφής χρήστη που απαιτούνται για να σχεδιαστεί μία λίστα χρηστών, με την πρόσθετη λειτουργία προσθήκης κάποιου χρήστη στην λίστα φίλων.

Ιδιότητες κλάσης:

- `lblSearchResults`: Είναι η ταμπέλα που υποδηλώνει ότι πρόκειται για το αποτέλεσμα της αναζήτησης χρηστών. Έχει σαν κείμενο “Search Results”.
- `participantsBackground`: Είναι το πλαίσιο πάνω στο οποίο εμφανίζονται οι χρήστες.
- `btnAdd`: Είναι μία λίστα από κουμπιά, ένα για τον κάθε χρήστη, η λειτουργία των οποίων είναι η προσθήκη ενός χρήστη στην λίστα φίλων.
- `usernames`: Είναι μία λίστα που έχει σαν στοιχεία τα ονόματα των χρηστών.
- `userIcons`: Είναι μία λίστα που έχει σαν στοιχεία τα εικονίδια (φωτογραφίες) των χρηστών.
- `userIDs`: Είναι μία λίστα που έχει σαν στοιχεία τα `userID` των χρηστών.

Μέθοδοι κλάσης:

- `UserSearchResultsGUI(results: list<User>)`: Αποτελεί την συνάρτηση δόμησης της γραφικής διεπαφής χρήστη των χρηστών αναζήτησης. Παίρνει σαν όρισμα μία λίστα με στοιχεία τύπου `User`.
- `BtnAdd(userID: int)`: Είναι η συνάρτηση η οποία καλείται όταν ο χρήστης πατήσει οποιοδήποτε από τα κουμπιά `Add` της λίστας `btnAdd`. Έχει σαν σκοπό την προσθήκη του αντίστοιχου χρήστη στην λίστα φίλων. Μεταφέρει τον έλεγχο στον ελεγκτή `FriendsPageController`. Συγκεκριμένα στην συνάρτηση αυτού, `AddFriend(userID: int)`.

### Controller `FriendsPageController`

«Controller» <code>FriendsPageController</code>
<code>+FriendsPageController(): void</code> <code>+ShowFriends(): void</code> <code>+ShowUserSearchResults(): void</code> <code>+AddFriend(userID: int): void</code> <code>+SearchUsers(username: string): void</code>

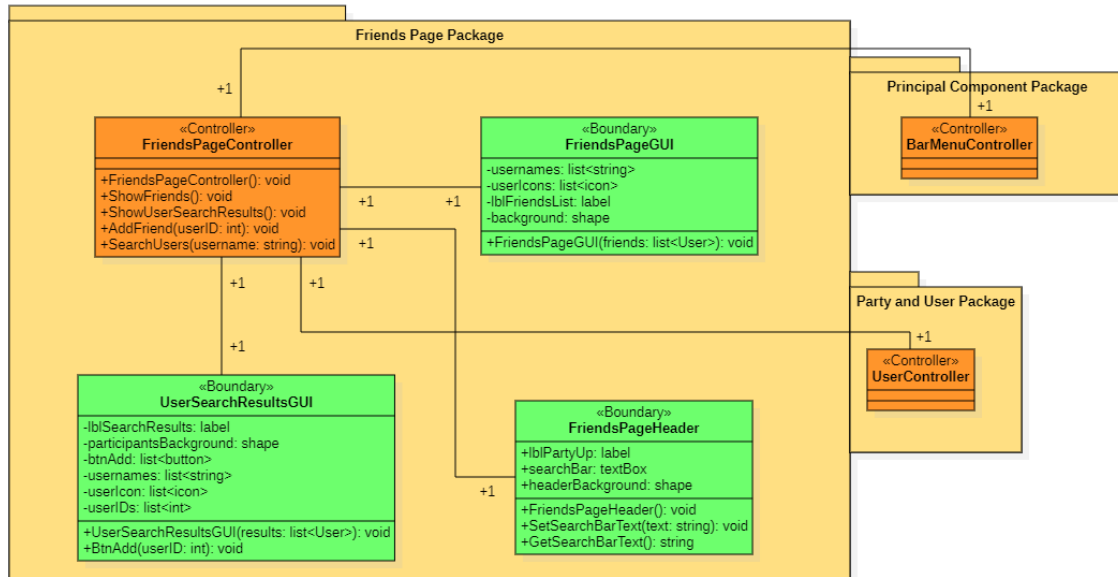
Η κλάση αυτή αποτελεί τον ελεγκτή της σελίδας φίλων του χρήστη.

Μέθοδοι κλάσης:

- `FriendsPageController`: Αποτελεί την συνάρτηση δόμησης του ελεγκτή της σελίδας φίλων του χρήστη.
- `ShowFriends()`: Είναι η συνάρτηση που είναι υπεύθυνη για την εμφάνιση της γραφικής διεπαφής χρήστη των φίλων του χρήστη.
- `ShowUserSearchResults()`: Είναι η συνάρτηση που είναι υπεύθυνη για την εμφάνιση της γραφικής διεπαφής χρήστη των αναζητούμενων χρηστών.
- `AddFriend(userID: int)`: Είναι η συνάρτηση που είναι υπεύθυνη για την προσθήκη του χρήστη με το συγκεκριμένο `userID` στην λίστα φίλων του χρήστη. Μεταφέρει τον έλεγχο στον ελεγκτή `UserController`, και συγκεκριμένα στην ομώνυμη συνάρτηση αυτού, `AddFriend(userID: int)`.
- `SearchUsers(username: string)`: Είναι η συνάρτηση που είναι υπεύθυνη για την αναζήτηση των χρηστών, με όνομα όμοιο με το όρισμα αυτής. Συγκεκριμένα μεταφέρει τον έλεγχο στον ελεγκτή `UserController`, ο οποίος χρησιμοποιεί το όρισμα αυτό για να αντλήσει τους χρήστες από την βάση δεδομένων των χρηστών μέσω της κλάσης `UserDBProxy`.



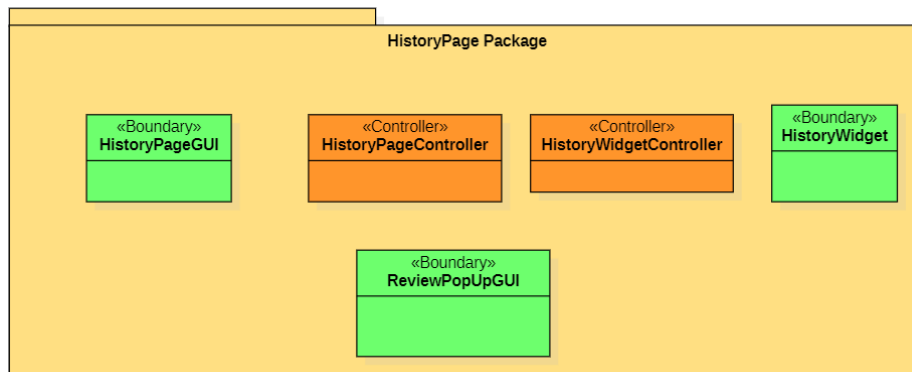
#### 4.2.1.2 Διαγράμματα κλάσεων - Πακέτο FriendsPage



#### 4.3 <Πακέτο(α) λεξιλογίου σεναρίων χαμηλής προτεραιότητας >

##### 4.3.1 Πακέτο History Page

Το πακέτο αυτό αναφέρεται στην δυνατότητα του χρήστη να βλέπει το ιστορικό των συμμετέχων πάρτι και να μπορεί να αξιολογήσει τον διοργανωτή.



##### 4.3.1 Ορισμού - Πακέτο HistoryPage

Controller  
HistoryPageController





Η κλάση αυτή αποτελεί τον ελεγκτή που είναι υπεύθυνος για την λειτουργικότητα και εμφάνιση του ιστορικού πάρτι.

Μέθοδοι της Κλάσης:

- HistoryPageController(): default constructor.
- setFrame(): η μέθοδος αυτή είναι υπεύθυνη για την αναπαράσταση των παλαιών πάρτι καλώντας την συνάρτηση setFrame() της κλάσης HistoryWidgetController.
- viewRecentlyJoinedParties(): η μέθοδος αυτή υλοποιεί την εμφάνιση των παλαιών πάρτι.
- getRecentlyJoinedParties(): η μέθοδος αυτή επιστρέφει μια λίστα από Party που την χρησιμοποιεί η συνάρτηση HistoryWidgetController(parties: list<Party>) για την δημιουργία αντικειμένων τύπου HistoryWidget(Party) με βάση το πλήθος συμμετοχών του χρήστη σε παρελθόντα πάρτι, αυτά περιέχονται στην κλάση HistoryWidgetController.

### Boundary HistoryPageGUI

«Boundary» <b>HistoryPageGUI</b>
+lblPartyHistory: label -background: shape
+HistoryPageGUI(): void

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση της σελίδας του ιστορικού πάρτι.

Χαρακτηριστικά της κλάσης:

- lblPartyHistory: κείμενο “Party History: ”.
- background: αποτελεί το σχήμα και το χρώμα στο παρασκήνιο του HistoryPage.

Μέθοδοι της Κλάσης:

- HistoryPageGUI(): default constructor.

### Controller HistoryWidgetController

«Controller» <b>HistoryWidgetController</b>
+HistoryWidgetController(parties: list<Party>): void +setFrame(): void +showReviewPopUpGUI(partyID: int): void +setRating(partyID: int, rating: int): void

Η κλάση αυτή αποτελεί τον ελεγκτή που είναι υπεύθυνος για τον καθορισμό του κάθε πάρτι που είχε συμμετέχει ο χρήστης καθώς και την λειτουργικότητα της αξιολόγησης του διοργανωτή.

Μέθοδοι της Κλάσης:

- HistoryWidgetController(parties: list<Party>): default constructor έχοντας ως όρισμα μια λίστα από Party.





- `SetFrame()`: η μέθοδος αυτή είναι υπεύθυνη για την αναπαράσταση των παλαιών πάρτι το ένα κάτω από το άλλο.
- `ShowReviewPopUpGUI(partyID:int)`: η μέθοδος αυτή είναι υπεύθυνη για την αναπαράσταση του `PopUpGUI` για την αξιολόγηση του διοργανωτή.
- `SetRating(partyID:int, rating:int)`: η μέθοδος αυτή είναι υπεύθυνη για την λειτουργικότητα της αξιολόγησης του χρήστη προς τον διοργανωτή παίρνοντας ως όρισμα το `partyID` και `rating`.

### Boundary HistoryWidget

<b>«Boundary» HistoryWidget</b>
-btnReview: button
+HistoryWidget(Party): void -BtnReview(partyID: int): void

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση των παλαιών πάρτι στην σελίδα ιστορικό πάρτι (`HistoryPage`)

Χαρακτηριστικά της κλάσης:

- `btnReview`: ένα κουμπί για την εμφάνιση του `ReviewPopUpGUI`.

Μέθοδοι της Κλάσης:

- `HistoryWidget(Party)`: default constructor έχοντας ως όρισμα `Party`.
- `BtnReview(partyID:int)`: η μέθοδος αυτή είναι υπεύθυνη για την αναπαράσταση του `ReviewPopUpGUI`.

### Boundary ReviewPopUpGUI

<b>«Boundary» ReviewPopUpGUI</b>
-stars: simpleRatingBar
+ReviewPopUpGUI(): void +OnClick(rating: int): void

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του μενού αξιολόγησης του διοργανωτή του πάρτι.

Χαρακτηριστικά της κλάσης:

- `stars`: είναι μια μπάρα αξιολόγησης με πέντε αστέρια στην σειρά.

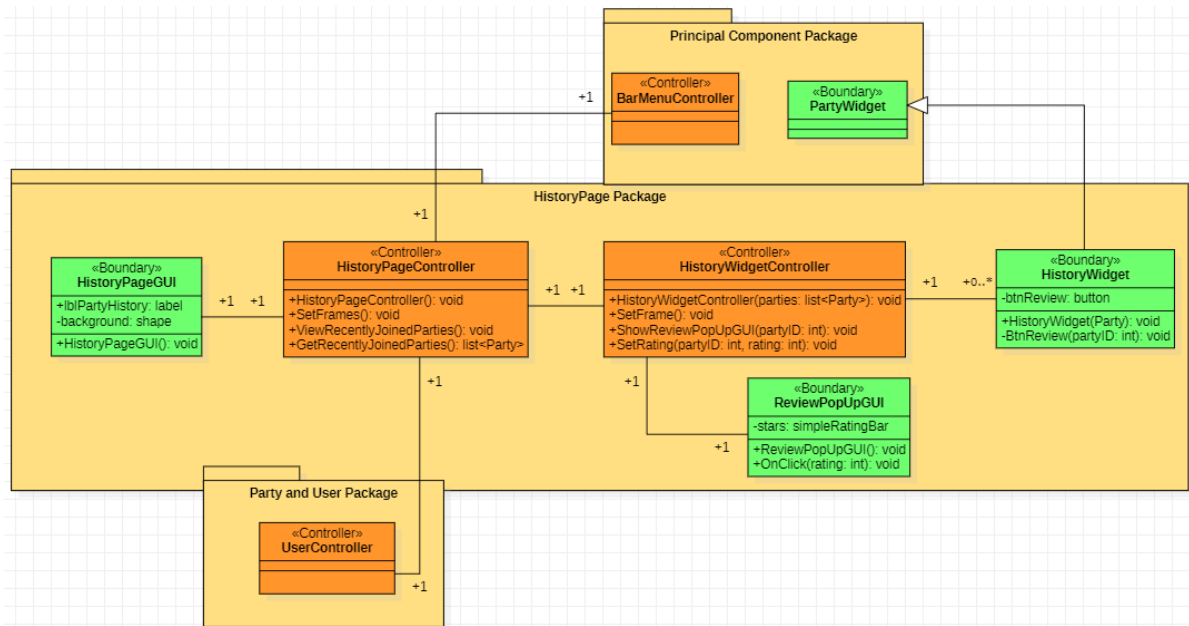
Μέθοδοι της Κλάσης:

- `ReviewPopUpGUI()`: default constructor.



- OnClick(rating:int): η μέθοδος αυτή είναι υπεύθυνη για την αναπαράσταση γεμίζοντας τα επιλεγμένα αστέρια με χρώμα αλλά και την μετάβαση της αξιολόγησης προς τον HistoryWidgetController για την διαχείρισή της.

#### 4.3.2 Διαγράμματα κλάσεων πακέτου HistoryPage





## **Παράρτημα Ι – Γλωσσάριο**

Το σετ των ακρωνυμίων που χρησιμοποιείτε στο έγγραφο

ΟΑ	Ομάδα Εργασίας
ΛΑ-xxxx	Λειτουργική Απαίτηση xxxx
ΜΛΑ-xxxx	Μη Λειτουργική Απαίτηση xxxx
GUI	Graphical User Interface



---

## Παράρτημα II – Ανοιχτά Θέματα

<Μια δυναμική λίστα με ανοιχτά θέματα απαιτήσεων, θα δημιουργηθεί στο παράρτημα Γ>

- Προσβασιμότητα της εφαρμογής σε άτομα με ειδικές ανάγκες , συμφωνία με πρότυπο WCAG 2.0.
- Ο χρήστης να μπορεί να αναφέρει ένα δημοσιευμένο πάρτι ως ψεύτικο ή επικίνδυνο και να υπάρχει ένας διαχειριστής προκειμένου να ελέγχει τέτοιου είδους καταχρήσεις.
- Με μεθόδους αυτοματοποίησης και μηχανικής μάθησης βάσει της δημοτικότητας του χρήστη, ένας χρήστης να μπορεί να έχει έναν συγκεκριμένο μέγιστο αριθμό διοργάνωσης πάρτι.
- Η εφαρμογή να διαθέτει περισσότερες αλληλεπιδράσεις μεταξύ φίλων, όπως συνομιλία και προφίλ.