

**Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής**

**Πρόγραμμα Μεταπτυχιακών Σπουδών**

**«Προηγμένα Συστήματα Πληροφορικής»**

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Εφαρμογή καταγραφής χωροχρονικού αποτυπώματος Memory Tracker application and Web platform</b>
Όνοματεπώνυμο Φοιτητή	<b>Λουκά Αικατερίνη</b>
Πατρώνυμο	<b>Χάνι</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/ 14047</b>
Επιβλέπων	<b>Ευθύμιος Αλέπης, Επίκουρος Καθηγητής</b>

Ημερομηνία Παράδοσης: **Οκτώβριος 2017**

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

## Περιεχόμενα

Ευρετήριο Εικόνων .....	4
Πρόλογος .....	5
Abstract .....	6
Ευχαριστίες.....	7
1 Εισαγωγή .....	8
1.1 Java .....	8
1.2 Android .....	8
1.3 Google Maps.....	8
1.4 Google Maps API .....	9
1.5 Η γλώσσα προγραμματισμού PHP .....	9
1.6 Η μορφοποίηση JSON.....	9
2 Παρουσίαση και χρήση εφαρμογής.....	10
2.1 Εφαρμογή MemoryTracker .....	10
2.1.1 Είσοδος και εγγραφή στην εφαρμογή .....	10
2.1.2 Αρχική Οθόνη .....	12
2.1.3 New Checkin .....	14
2.1.4 List of Memories.....	19
2.1.5 My Memories On Map .....	22
3 Παρουσίαση και χρήση site .....	24
3.1 Web εφαρμογή MemoryTracker.....	24
3.1.1 Είσοδος και εγγραφή στην εφαρμογή .....	24
3.1.2 Αλλαγή κωδικού χρήστη.....	26
3.1.3 New Check-in.....	27
3.1.4 List of Memories.....	30
3.1.5 Memories on Map .....	32
3.1.6 Εμφάνιση Check-in .....	33
4 Αρχιτεκτονική Συστήματος.....	35
4.1 MemoryTracker εφαρμογή Android .....	36
4.2 MemoryTracker Web εφαρμογή.....	45
5 Συμπεράσματα και Μελλοντικές Επεκτάσεις .....	55
Βιβλιογραφία .....	56

## Ευρετήριο Εικόνων

1 Είσοδος ή εγγραφή στην εφαρμογή .....	10
2 Εγγραφή στην εφαρμογή.....	11
3 Αρχική Οθόνη.....	12
4 Logout.....	13
5 Αποθήκευση τοποθεσίας .....	14
6 Εισαγωγή Σημειώσεων .....	15
7 Εισαγωγή εικόνας.....	16
8 Εισαγωγή εικόνας.....	17
9 Εισαγωγή βίντεο.....	18
10 List of Memories .....	19
11 Εμφάνιση check in.....	20
12 Video Full Screen .....	21
13 Memories On Map.....	22
14 Εύρεση Check In από τον χάρτη.....	23
15 Εισαγωγή χρήστη στην Web εφαρμογή.....	24
16 Νέος χρήστης στην Web εφαρμογή.....	25
17 Αρχική οθόνη Web εφαρμογής.....	26
18 Αλλαγή κωδικού πρόσβασης Web εφαρμογής .....	26
19 Αποθήκευση τοποθεσίας Web εφαρμογής .....	27
20 Αποθήκευση σημειώσεων Web εφαρμογή .....	28
21 Αποθήκευση εικόνας Web εφαρμογή .....	28
22 Αποθήκευση βίντεο Web εφαρμογή .....	29
23 Διαγραφή Check-in Web εφαρμογή .....	31
24 Τελευταία σελίδα λίστας.....	31
25 Χάρτης με τις τοποθεσίες Web εφαρμογή .....	32
26 Εμφάνιση διεύθυνσης Web εφαρμογή .....	32
27 Εμφάνιση Check-in Web εφαρμογής .....	33
28 Εμφάνιση βίντεο Web εφαρμογής .....	34
29 Δυνατότητα αποθήκευσης βίντεο Web εφαρμογής.....	34
30 Firebase Storage .....	37

## Πρόλογος

Στόχος της παρούσας διπλωματικής εργασίας είναι να χρησιμοποιήσει τις δυνατότητες που προσφέρουν οι έξυπνες κινητές συσκευές και συγκεκριμένα το λειτουργικό σύστημα Android, για την ανάπτυξη μιας εφαρμογής για κινητές συσκευές. Η εφαρμογή αυτή σκοπό έχει να καταγράφει το χωροχρονικό αποτύπωμα του χρήστη εν κινήσει. Η εφαρμογή αυτή διαθέτει διάφορες λειτουργίες που εξυπηρετούν τον στόχο αυτό, δηλαδή η καταγραφή μπορεί να συνδυάζεται με πολυμεσικά δεδομένα εν κινήσει χρηστών, όπως είναι η σύνδεση γεωγραφικών συντεταγμένων με ήχο-βίντεο-κείμενο ή η αναπαραγωγή αντίστοιχων ιστορικών αρχείων. Πέρα από την εφαρμογή για Android δημιουργήθηκε επιπλέον μια διαδικτυακή εφαρμογή για δημιουργία και επεξεργασία λογαριασμών χρηστών καθώς και μια βάση δεδομένων για μόνιμη αποθήκευση δεδομένων που αφορούν τους χρήστες και ενέργειές τους. Στην εργασία που ακολουθεί αρχικά υπάρχει μια σύντομη εισαγωγή όπου γίνεται μια πρώτη περιγραφή της εφαρμογής. Στο τρίτο κεφάλαιο γίνεται η παρουσίαση και χρήση εφαρμογής. Ακολούθως αναλύεται λεπτομερώς η διαδικασία υλοποίησης όλων των κομματιών του συστήματος. Ομοίως, στο τέτατο κεφάλαιο γίνεται η παρουσίαση και χρήση του site. Στο πέμπτο κεφάλαιο αναλύεται η αρχιτεκτονική του συστήματος.

## Abstract

The aim of this diploma is to use the capabilities offered by smart mobile devices, specifically the Android operating system, to develop a mobile application. This application is intended to record the user's place and time footprint on-the-go. This application has several functions for this purpose, so recording can be combined with multimedia data for users on-the-go, such as linking geographic coordinates with audio-video-text or playing back corresponding historical files. In addition to the Android app, an online application for creating and editing user's accounts was created, as well as a database for storing user's data and their actions. In the diploma that follows first there is a brief introduction with the description of the application that is made. The third chapter introduces the usage of the application. The procedure for the implementation of all parts of the system is then analyzed in detail. Similarly, the fourth chapter introduces the usage of the site. In Chapter Five, the architecture of the system is analyzed.

## Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή μου κο Ευθύμιο Αλέπη, για την βοήθεια του, την υποστήριξη του και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οδηγίες αλλά και κατευθύνσεις του, συνέβαλαν στο μέγιστο, για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Επιπλέον, θα ήθελα να ευχαριστήσω το Πανεπιστήμιο Πειραιά που με επέλεξε δίνοντας μου την ευκαιρία να συμμετάσχω στο μεταπτυχιακό πρόγραμμα και να αποκτήσω γνώσεις που θα με βοηθήσουν στην μετέπειτα επαγγελματική μου πορεία.

Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου και τους φίλους μου, που με στήριξαν ηθικά και ψυχολογικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο Πανεπιστήμιο Πειραιά αλλά και κατά τη δύσκολη αυτή περίοδο ολοκλήρωσης της διπλωματικής μου εργασίας.

## 1 Εισαγωγή

Σε αυτή την εργασία θα γίνει η υλοποίηση μιας εφαρμογής, η οποία θα μπορεί να καταγράφει το χωροχρονικό αποτύπωμα ενός χρήστη, το οποίο μπορεί να συνδυάζεται με πολυμεσικά δεδομένα εν κινήσει χρηστών ή να αναπαράγει αντίστοιχα ιστορικά αρχεία. Επιπρόσθετα, θα υλοποιηθεί web εφαρμογή η οποία θα κατασκευάζει σε χάρτη τέτοια αρχεία, τα οποία θα μπορεί επίσης να αναπαράγει. Οι δύο εφαρμογές θα μπορούν να ανταλλάξουν τα αρχεία μεταξύ τους.

### 1.1 Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι ότι τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα δηλαδή προσφέρει η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας.

### 1.2 Android

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας αλλά και πολλές άλλες συσκευές τελευταίας τεχνολογίας, όπως tablets, τηλεοράσεις και συστήματα πλοήγησης αυτοκινήτων (GPS). Είναι βασισμένο στον πυρήνα του λειτουργικού συστήματος Linux. Είναι σήμερα ένα από τα πιο δημοφιλή, αν όχι το δημοφιλέστερο, λειτουργικά συστήματα «έξυπνων συσκευών», γνωστών και ως «smart devices». Έχει βέβαια αρκετούς ανταγωνιστές στην αγορά, όπως για παράδειγμα το λειτουργικό σύστημα iOS της Apple και το Windows Mobile της Microsoft.

### 1.3 Google Maps

Η υπηρεσία Google Maps είναι μια διαδικτυακή υπηρεσία χαρτογράφησης που παρέχεται από την Google και τρέχει σε Desktop και Smartphones. Η υπηρεσία επίσης προσφέρει ένα ευρύ φάσμα εφαρμογών όπως δορυφορικές εικόνες, οδικούς χάρτες, φωτογραφίες οδών καθώς και λειτουργίες όπως σχεδιασμό διαδρομής και πλοήγησης. Επίσης υποστηρίζει τη χρήση χαρτών από ιστοσελίδες τρίτων μέσω του Google Maps API και πολλά άλλα όπως εγγραφή τοποθεσίας επιχείρησης και οργανισμών στους χάρτες της. Η βάση δεδομένων δεν ενημερώνεται σε πραγματικό χρόνο αλλά γίνεται συνεχής αναβάθμιση των εικόνων και καμία εικόνα δεν είναι παλαιότερη των τριών χρόνων. Η εφαρμογή Google Maps για κινητά είναι η πιο δημοφιλής και



ευρέως διαδεδομένη εφαρμογή για smartphones με ένα εντυπωσιακό ποσοστό της τάξεως του 54% των χρηστών smartphones να το χρησιμοποιούν τουλάχιστον μία φορά και πιο συγκεκριμένα κατά τον μήνα Αύγουστο.

## 1.4 Google Maps API

Πρόκειται για μια δωρεάν υπηρεσία η οποία επί του παρόντος δεν περιέχει διαφημίσεις, αλλά, η Google περιέχει στους όρους χρήσης ότι διατηρεί το δικαίωμα να προβάλλουν διαφημίσεις στο μέλλον. Με τη χρήση του Google Maps API, είναι δυνατόν να ενσωματώσετε το Google Maps σε μια ιστοσελίδα τρίτου, επί του οποίου σε συγκεκριμένο χώρο συγκεκριμένα δεδομένα μπορούν να επικαλύπτονται. Πάνω από 1,000,000 ιστοσελίδες χρησιμοποιούν το Google Maps API, καθιστώντας αυτή ως την πιο ευρέως χρησιμοποιημένη διαδικτυακή εφαρμογή ανάπτυξης API. Το Google Maps API είναι δωρεάν για εμπορική χρήση, με την προϋπόθεση ότι η περιοχή στην οποία χρησιμοποιείται είναι προσβάσιμη στο κοινό και δεν χρεώνει για την πρόσβαση, και δεν έχουν περισσότερες από 25000 επισκέψεις στο χάρτη σε μια ημέρα. Για τα Sites που δεν πληρούν αυτές τις απαιτήσεις μπορούν να αγοράσουν το Google Maps API για επιχειρήσεις.

## 1.5 Η γλώσσα προγραμματισμού PHP

Η γλώσσα προγραμματισμού PHP είναι μια server-side scripting γλώσσα, δηλαδή βοηθάει στην επικοινωνία του server με τον client. Η PHP αναπτύχθηκε το 1994 και μέχρι σήμερα θεωρείται μία από τις πιο πολυχρησιμοποιημένες side-scripting γλώσσες προγραμματισμού.

## 1.6 Η μορφοποίηση JSON

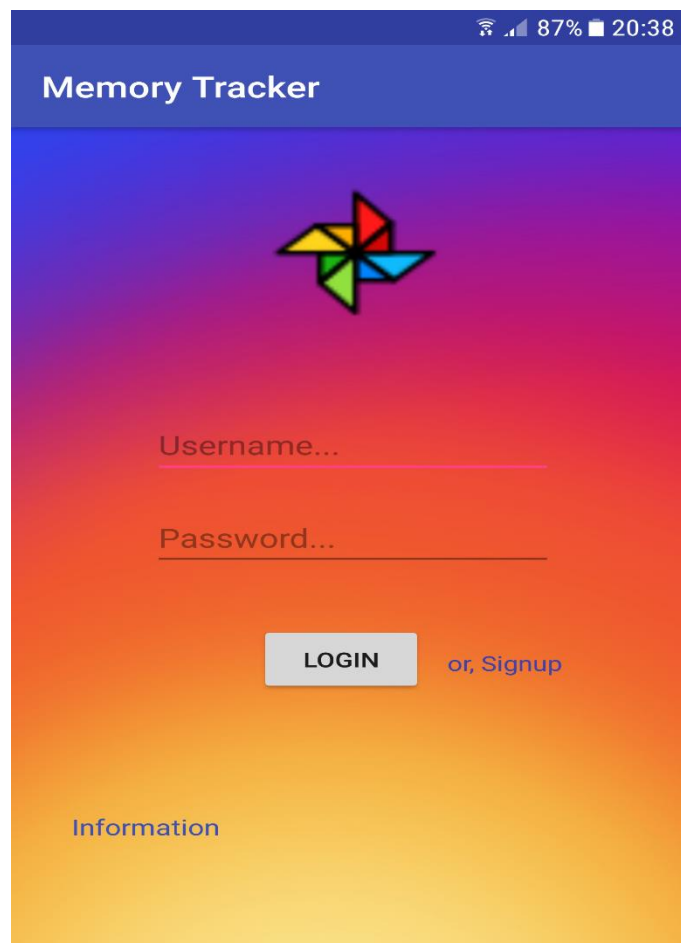
Τα PHP scripts εμφανίζουν τα αποτελέσματα στην μορφή JSON. Το JSON, ονομασία που προέρχεται από τον αγγλικό όρο JavaScript Object Notation, είναι ένας τρόπος απεικόνισης αντικειμένων που αποτελούνται από ζευγάρια κλειδιού- τιμής (key-value pairs) και είναι εύκολα αναγνώσιμος από τον άνθρωπο.

## 2 Παρουσίαση και χρήση εφαρμογής

### 2.1 Εφαρμογή MemoryTracker

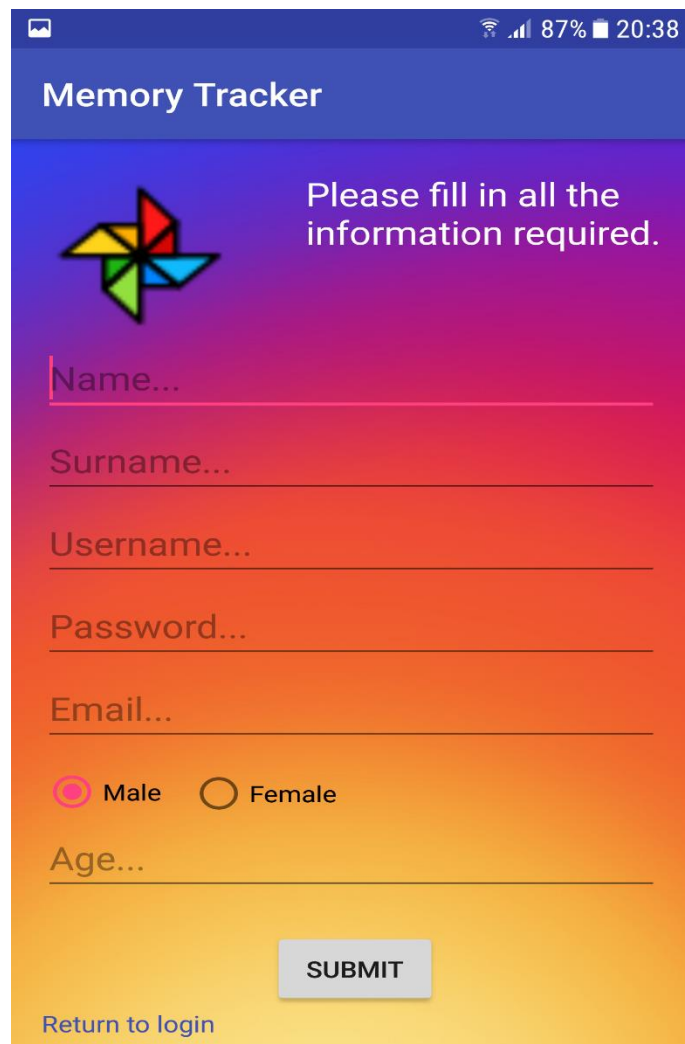
#### 2.1.1 Είσοδος και εγγραφή στην εφαρμογή

- Όταν ανοίγει η εφαρμογή, εμφανίζεται η πρώτη οθόνη, στην οποία ο χρήστης πρέπει να κάνει είσοδο ή εγγραφή στην εφαρμογή, όπως φαίνεται στην παρακάτω εικόνα. Εάν ο χρήστης έχει πραγματοποιήσει εγγραφή τότε επιλέγει «Login» από την εικόνα 1. Στην συνέχεια ο χρήστης καλείται να δώσει το όνομα χρήστη, τον προσωπικό του κωδικό και να πατήσει «Login» προκειμένου να συνδεθεί στην εφαρμογή.



1 Είσοδος ή εγγραφή στην εφαρμογή

- Εάν ο χρήστης δεν έχει κάνει εγγραφή στην εφαρμογή, τότε θα πρέπει να επιλέξει την επιλογή «Signup» (εικόνα 1). Προκειμένου να δημιουργηθεί ο λογαριασμός για την εφαρμογή MemoryTracker, ο χρήστης θα πρέπει να συμπληρώσει όλα τα απαραίτητα πεδία και να επιλέξει «Register» (εικόνα 2). Πατώντας την επιλογή «Return to login», ο χρήστης επιστρέφει στην αρχική οθόνη (εικόνα 1).



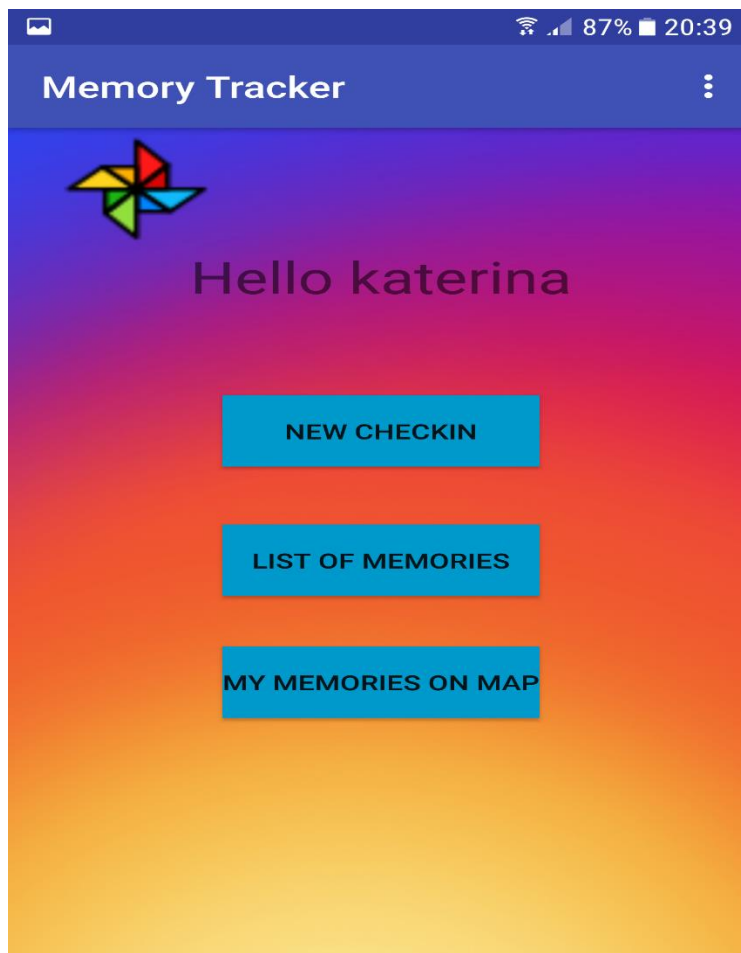
The screenshot shows a mobile application interface for 'Memory Tracker'. At the top, there is a status bar with a signal icon, 87% battery, and the time 20:38. Below the status bar is a blue header with the text 'Memory Tracker'. The main content area has a purple-to-orange gradient background. On the left, there is a colorful pinwheel logo. To the right of the logo, the text reads 'Please fill in all the information required.' Below this, there are several input fields: 'Name...', 'Surname...', 'Username...', 'Password...', 'Email...', and 'Age...'. There are also two radio buttons for gender selection, labeled 'Male' (selected) and 'Female'. At the bottom center, there is a grey 'SUBMIT' button. In the bottom left corner, there is a link that says 'Return to login'.

2 Εγγραφή στην εφαρμογή

### 2.1.2 Αρχική Οθόνη

Όταν ο χρήστης συνδεθεί επιτυχώς στο σύστημα τότε έχει διαθέσιμες όλες τις λειτουργίες της MemoryTracker εφαρμογής.

Η αρχική οθόνη της εφαρμογής φαίνεται στην παρακάτω εικόνα.

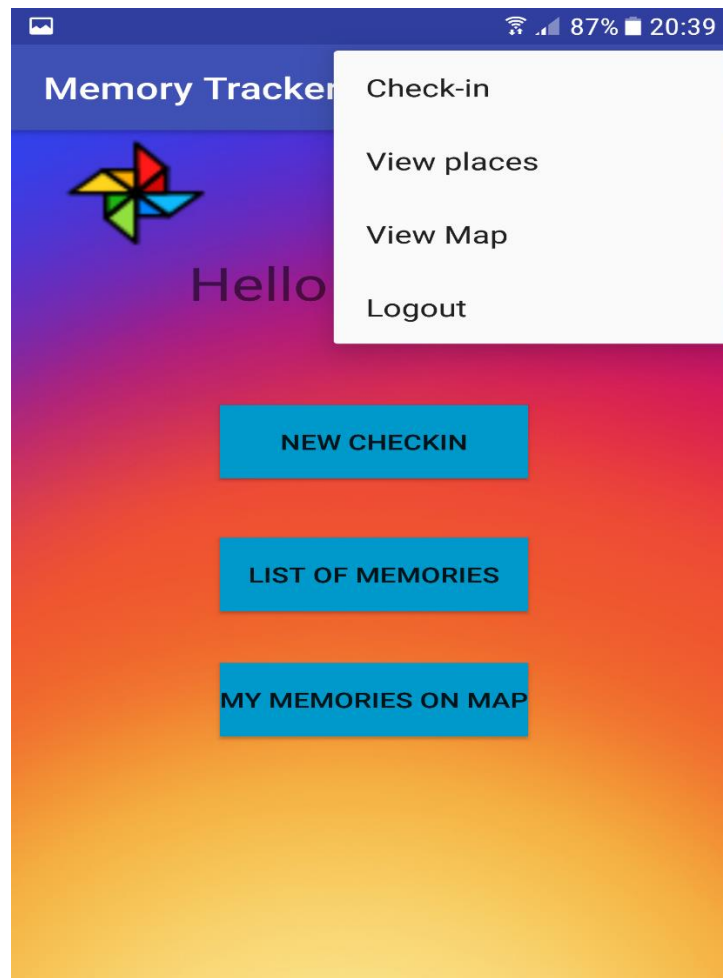


3 Αρχική Οθόνη

Ο χρήστης εισέρχεται στην βασική οθόνη της εφαρμογής στην οποία έχει 3 βασικές επιλογές.

- ❖ Η πρώτη επιλογή είναι να κάνει νέο Check in πατώντας το κουμπί <<**NEW CHECKIN**>>.
- ❖ Η επόμενη επιλογή είναι να δει τα μέρη που έχει επισκεφθεί σε μία λίστα, μαζί με τις αντίστοιχες αποθηκευμένες πληροφορίες, πατώντας το κουμπί <<**LIST OF MEMORIES**>>.
- ❖ Η τελευταία επιλογή είναι να δει ο χρήστης τα check in του, αναρτημένα στον χάρτη με βάση την τοποθεσία, πατώντας το κουμπί «**MY MEMORIES ON MAP**».

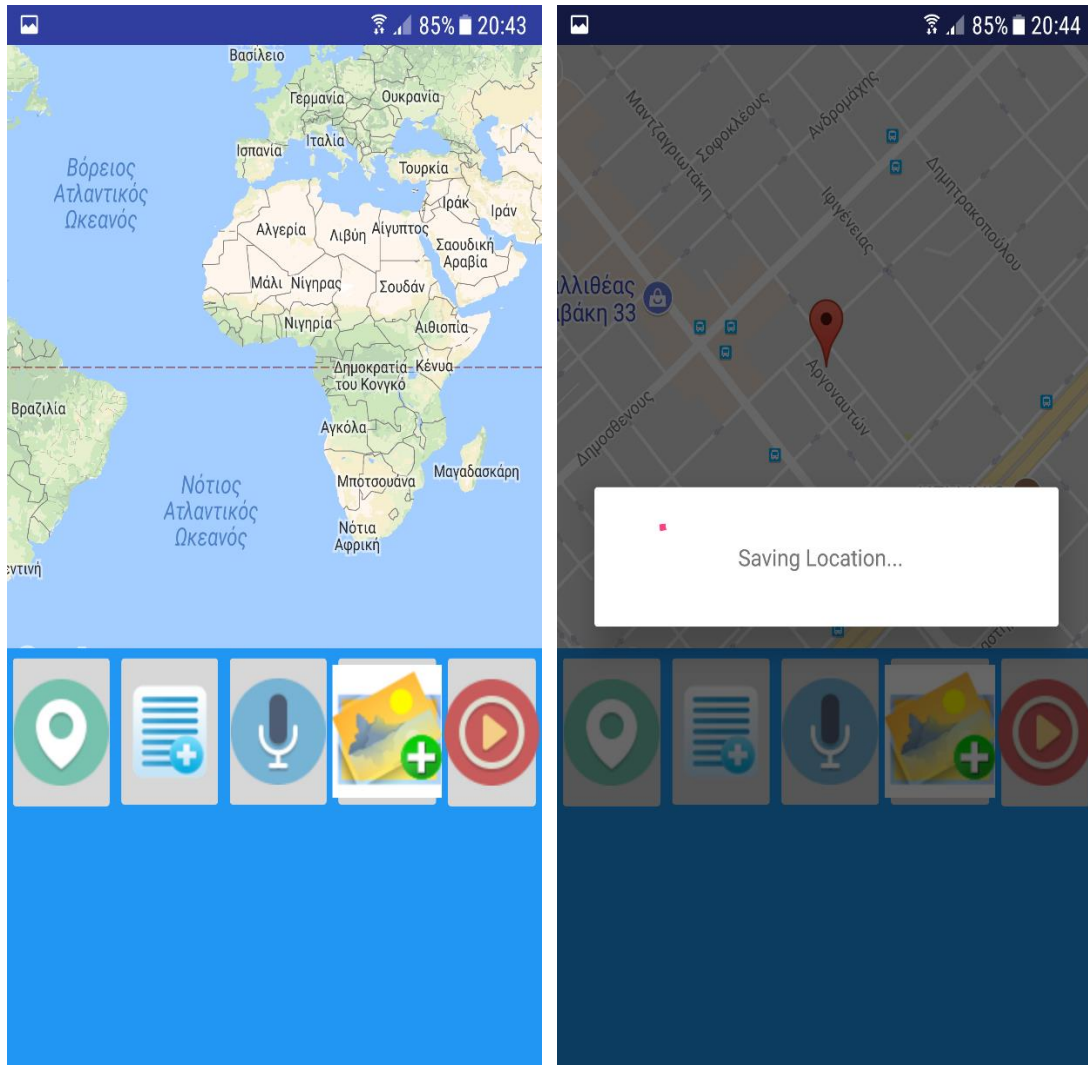
Επίσης υπάρχει και μενού στα δεξιά στο οποίο έχει ακριβώς τις ίδιες δυνατότητες, προσθέτοντας την επιλογή του <<**LOGOUT**>> σε περίπτωση που ο χρήστης θέλει να αποσυνδεθεί ή να μπει με καινούριο λογαριασμό.



4 Logout

## 2.1.3 New Checkin

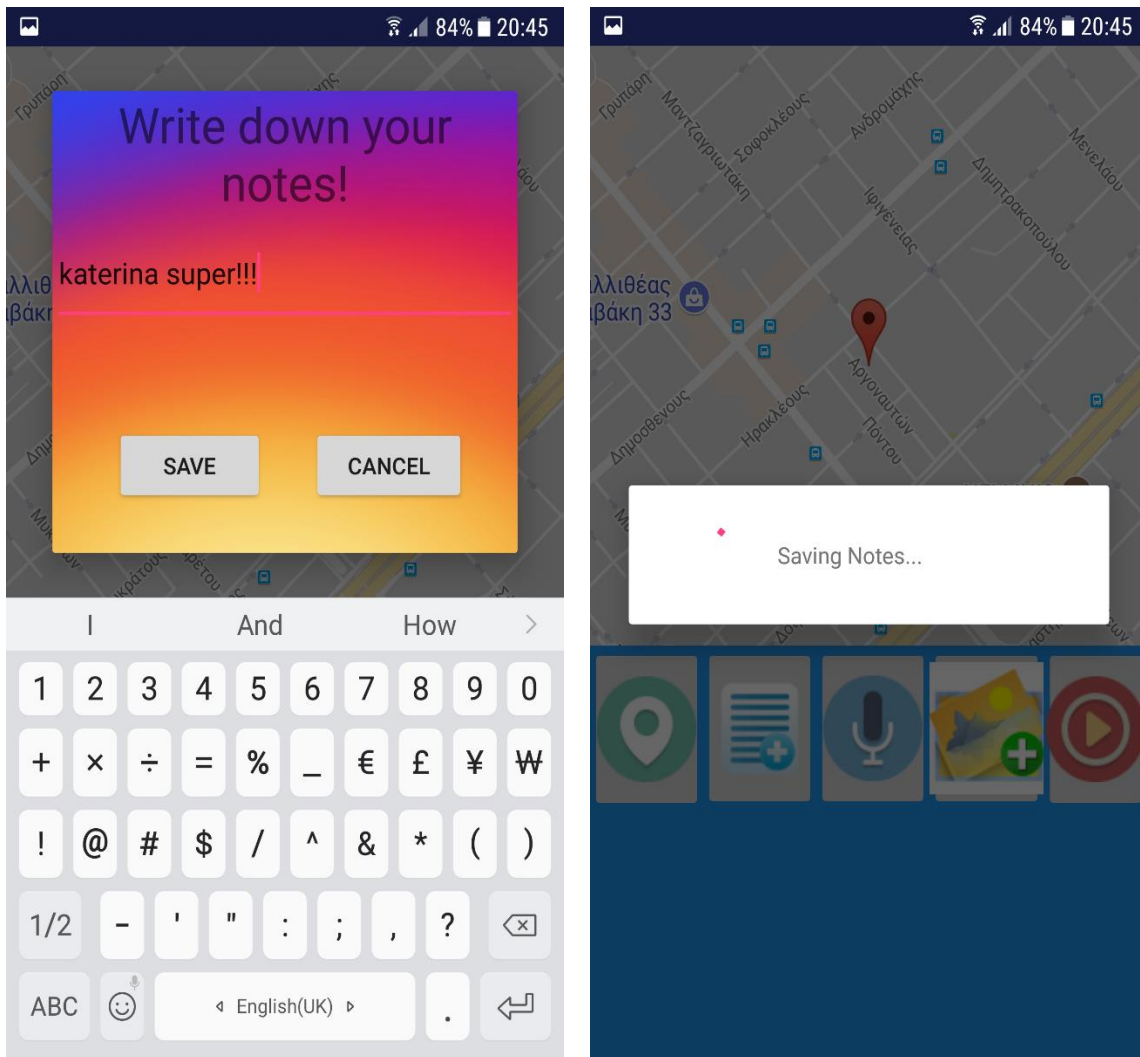
- ✓ Αρχικά ο χρήστης πρέπει να πατήσει το κουμπί της τοποθεσίας, έτσι ώστε να αποθηκευτεί. Με την αποθήκευση τοποθεσίας, ενεργοποιούνται και οι υπόλοιπες δυνατότητες αποθήκευσης.



5 Αποθήκευση τοποθεσίας

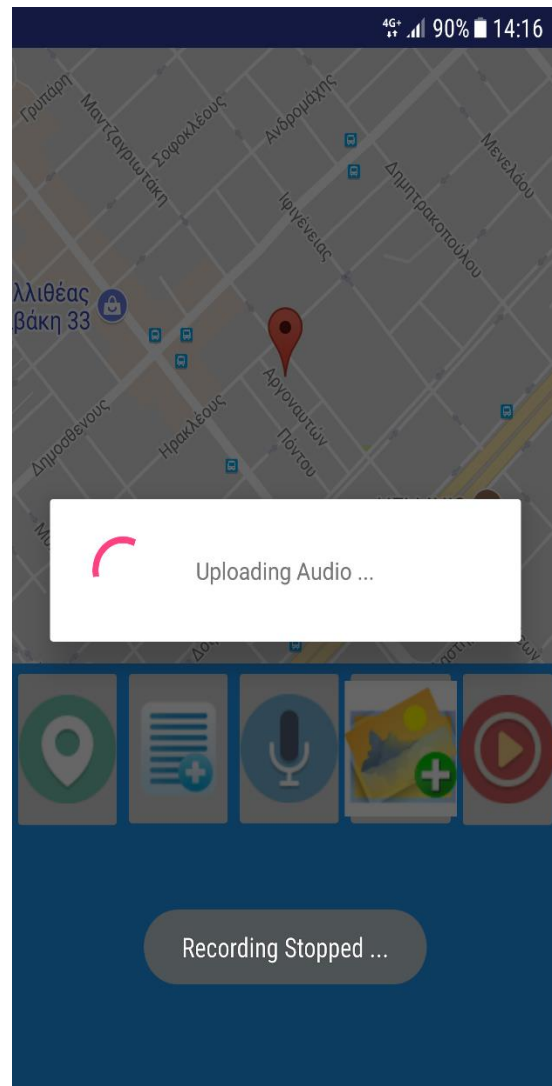
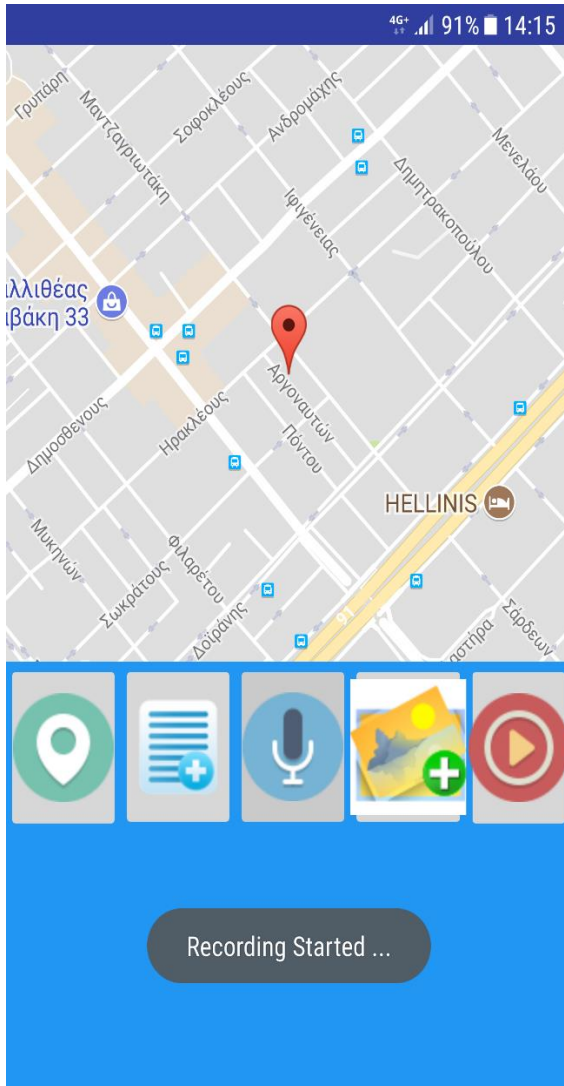
Έπειτα ο χρήστης έχει τη δυνατότητα προαιρετικής επιλογής αποθήκευσης σημειώσεων, ήχου, εικόνας και βίντεο.

✓ Εισαγωγή σημειώσεων



6 Εισαγωγή Σημειώσεων

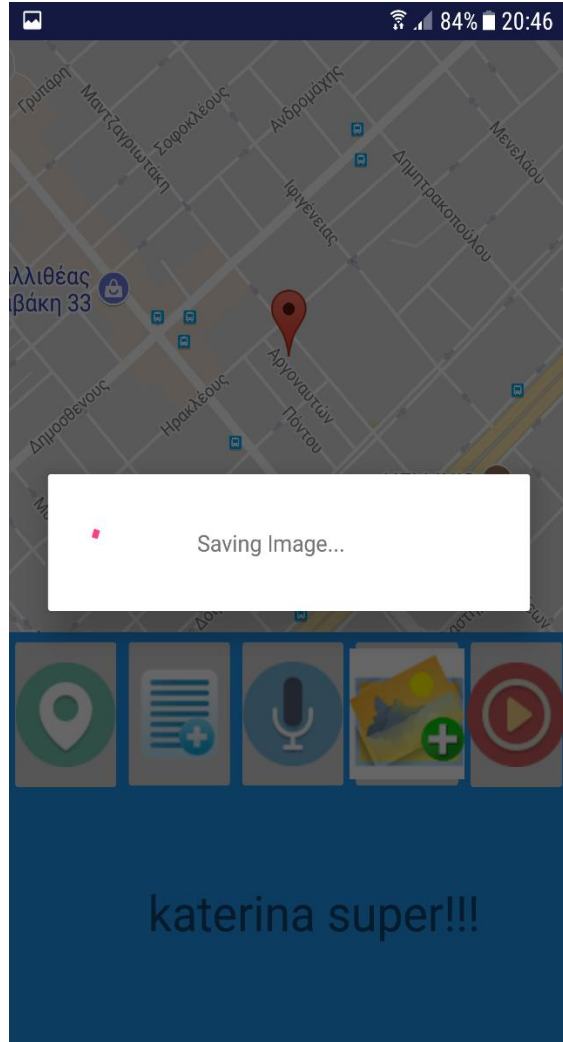
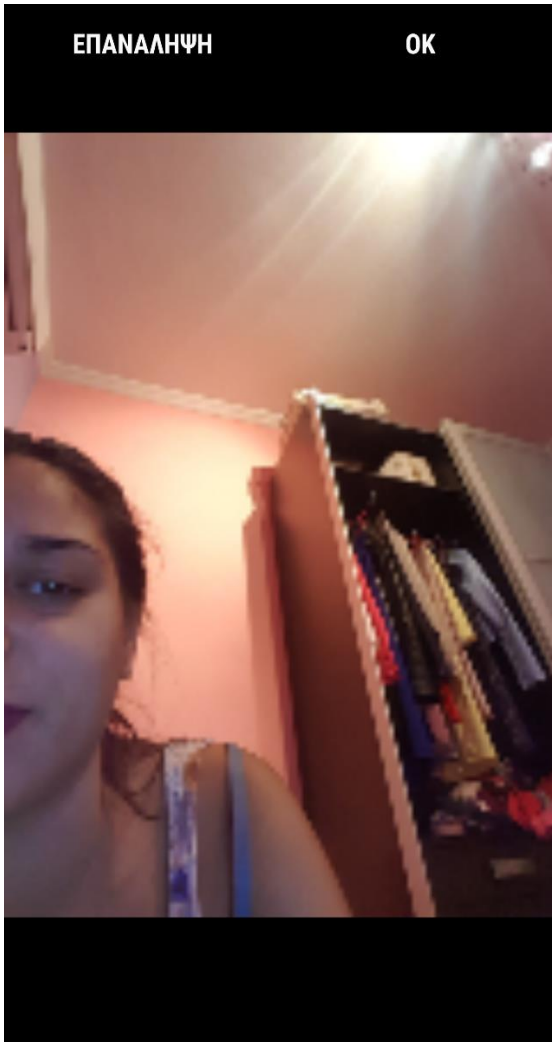
- ✓ Ο χρήστης πρέπει να πατάει συνεχόμενα το κουμπί της ηχογράφησης όσο ηχογραφεί. Αφήνοντας το κουμπί, επιτυγχάνεται η αποθήκευση της ηχοφράγησης.



7 Εισαγωγή εικόνας

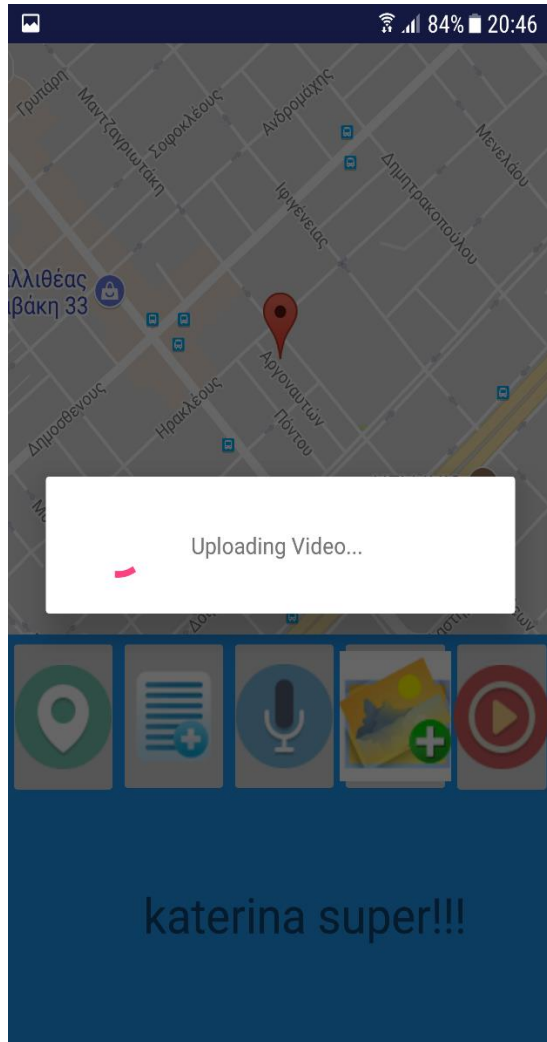
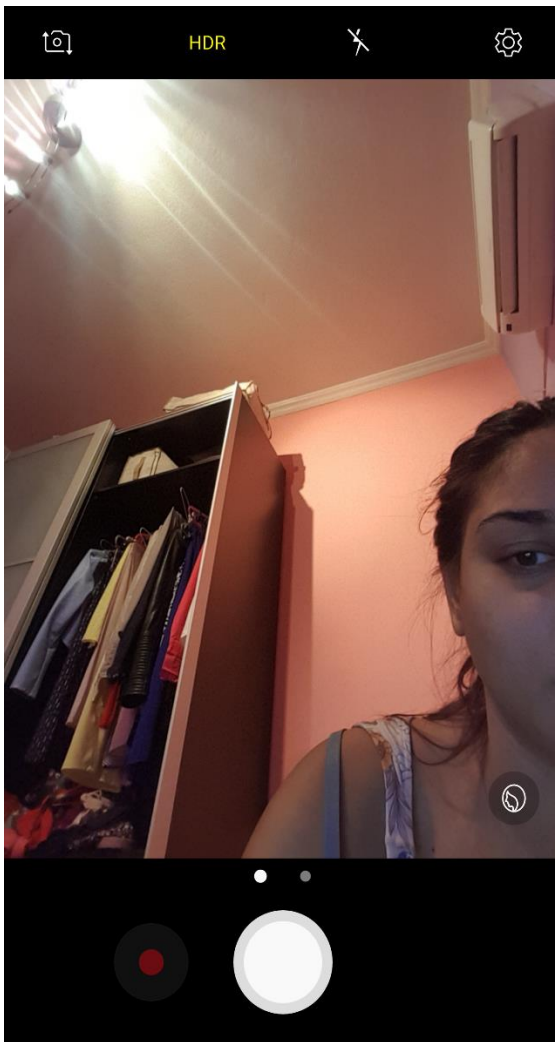


- ✓ Εισαγωγή εικόνας



8 Εισαγωγή εικόνας

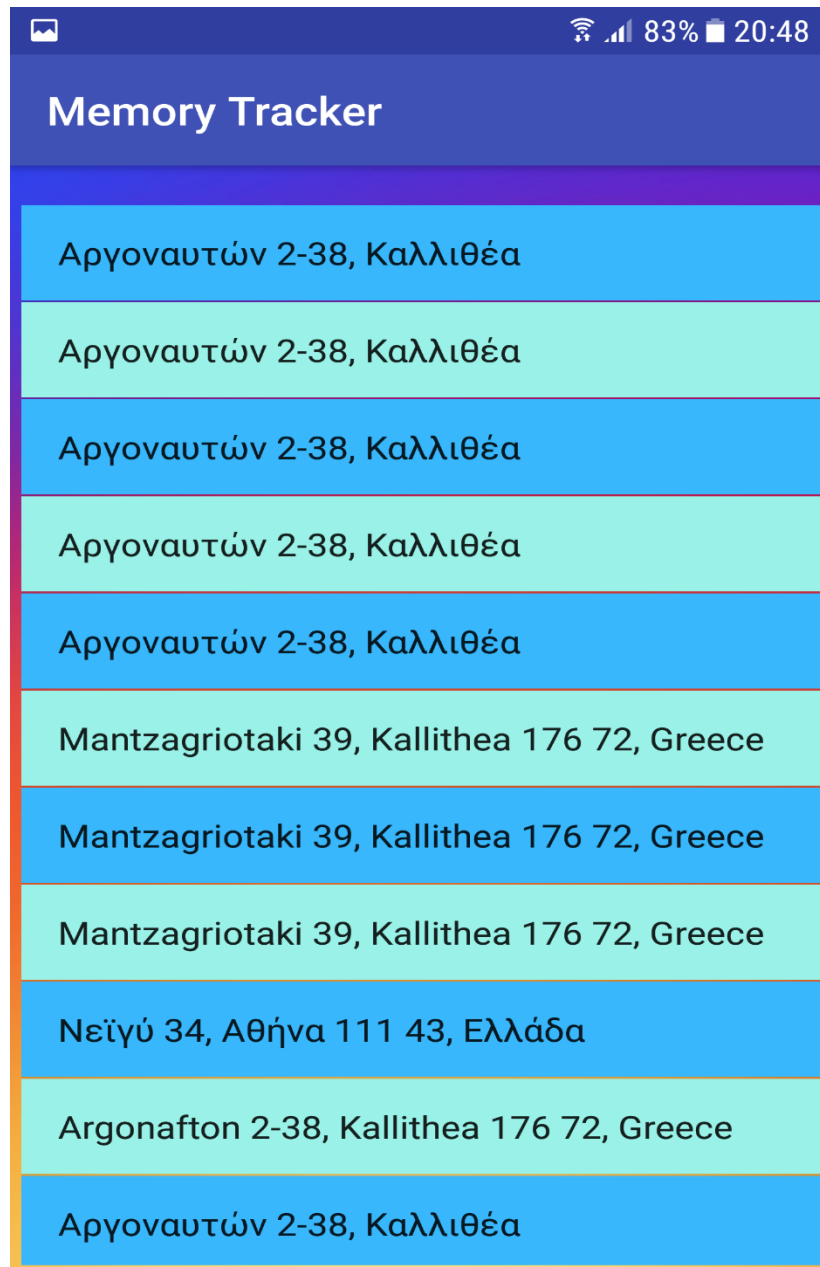
✓ Εισαγωγή βίντεο



9 Εισαγωγή βίντεο

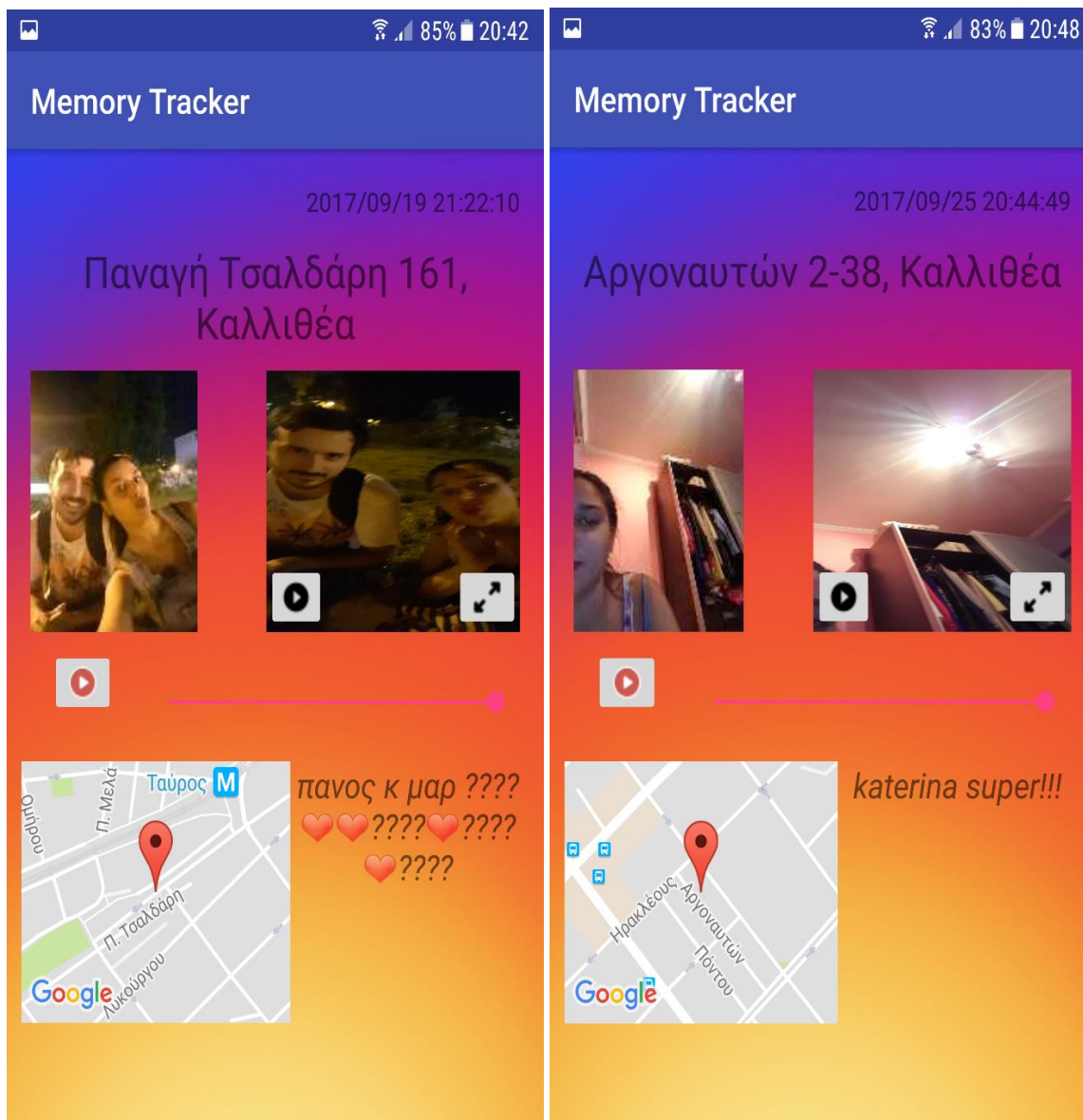
### 2.1.4 List of Memories

- Πατώντας το κουμπί «List of Memories», ο κάθε χρήστης έχει τη δυνατότητα να δει σε μία λίστα όλες τις τοποθεσίες που έχει επισκεφθεί απο την πιο πρόσφατη προς την παλαιότερη. Ο χρήστης μπορεί να πατήσει πάνω σε οποιαδήποτε τοποθεσία θέλει και να δει όλα τα αποθηκευμένα δεδομένα.



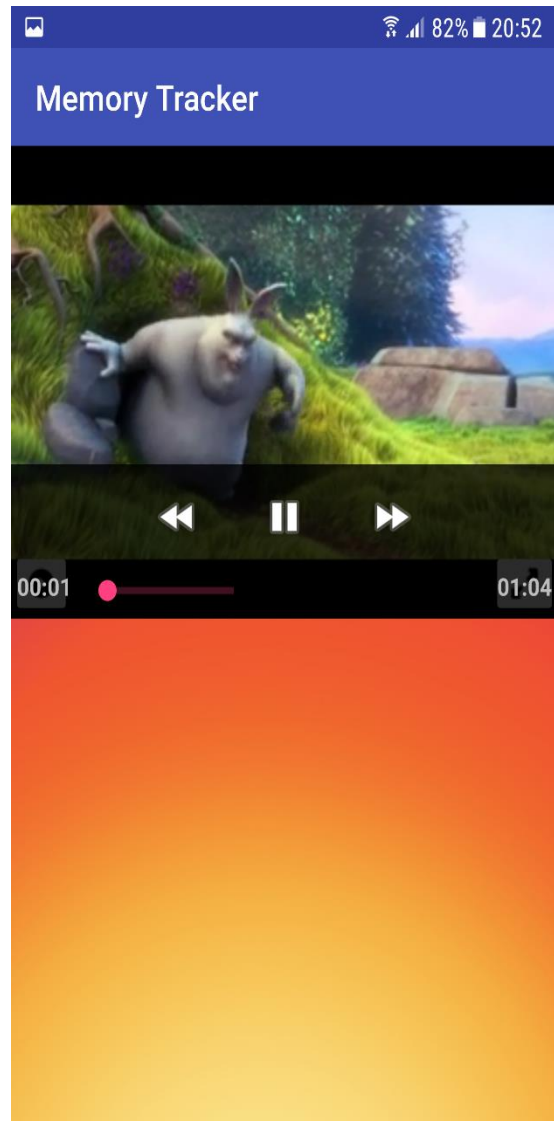
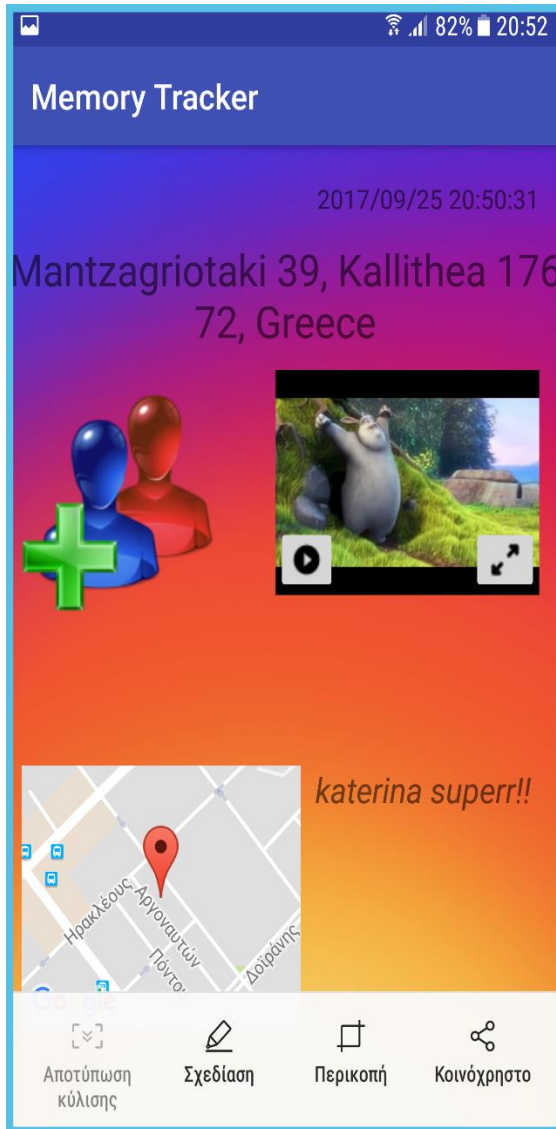
10 List of Memories

- ✓ Πατώντας πάνω σε κάποια τοποθεσία, ο χρήστης μπορεί να δει την εικόνα και το βίντεο που έχει αποθηκεύσει. Επίσης πάνω δεξιά φαίνεται η ημερομηνία η οποία έκανε το Check in του. Παρακάτω φαίνεται η διεύθυνση, καθώς και η τοποθεσία στον χάρτη. Δίπλα στο χάρτη εμφανίζονται η σημειώσεις που έχει γράψει ο χρήστης και στη μέση φαίνεται η ηχογράφιση.



11 Εμφάνιση check in

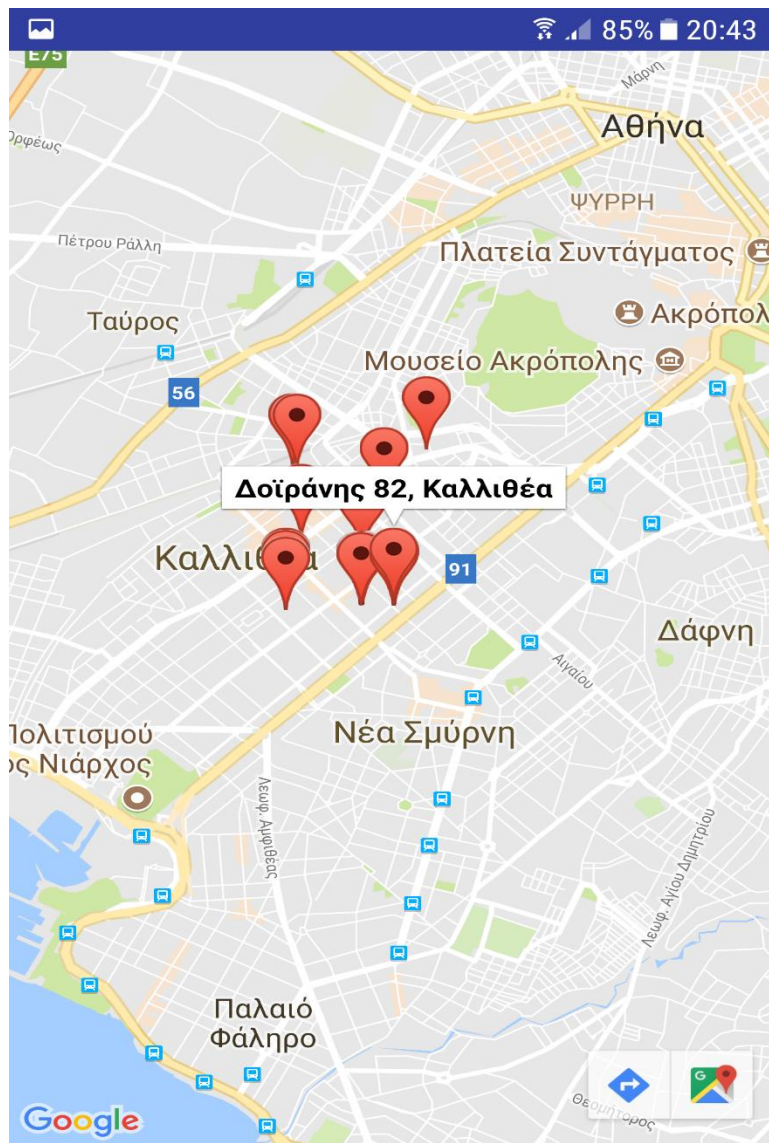
- ✓ Επίσης υπάρχει δυνατότητα Full Screen για προβολή του βίντεο πατώντας το κουμπί της μεγιστοποίησης.



12 Video Full Screen

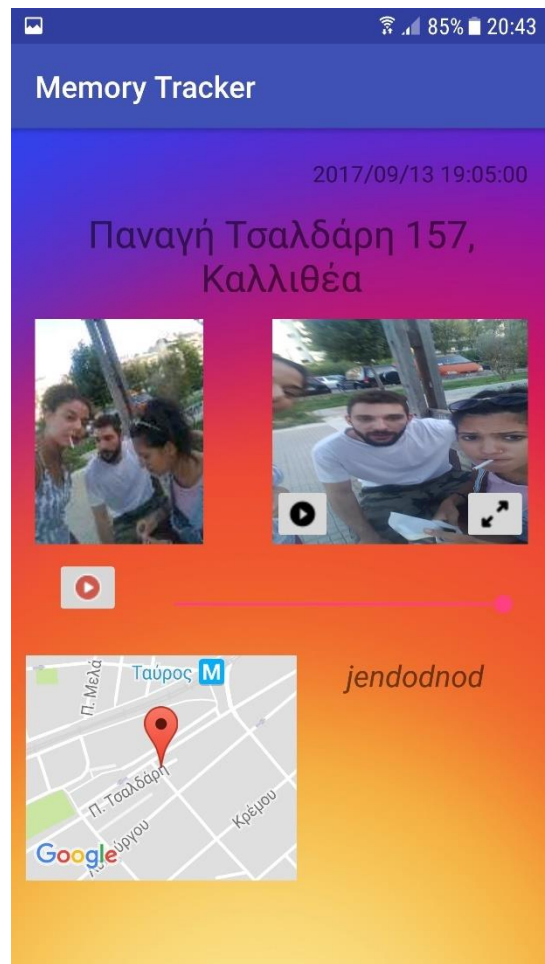
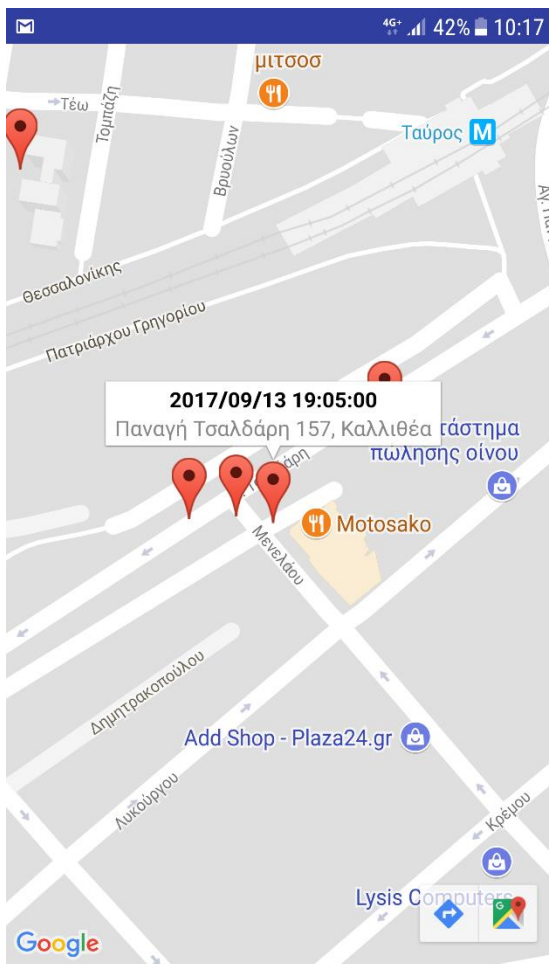
### 2.1.5 My Memories On Map

- Με την επιλογή «My Memories on Map», ο χρήστης βλέπει όλα τα μέρη τα οποία έχει επισκεφθεί απεικονισμένα στον χάρτη, σε μορφή δεικτών.



13 Memories On Map

- ✓ Πατώντας πάνω σε έναν δείκτη, εμφανίζεται η διεύθυνση της συγκεκριμένης τοποθεσίας. Αν ο χρήστης θέλει να δει το Checkin του, τότε μπορεί να πατήσει πάνω στην διεύθυνση. Τότε ο χρήστης μεταφέρεται στα αποθηκευμένα στοιχεία της επιλεγμένης τοποθεσίας.



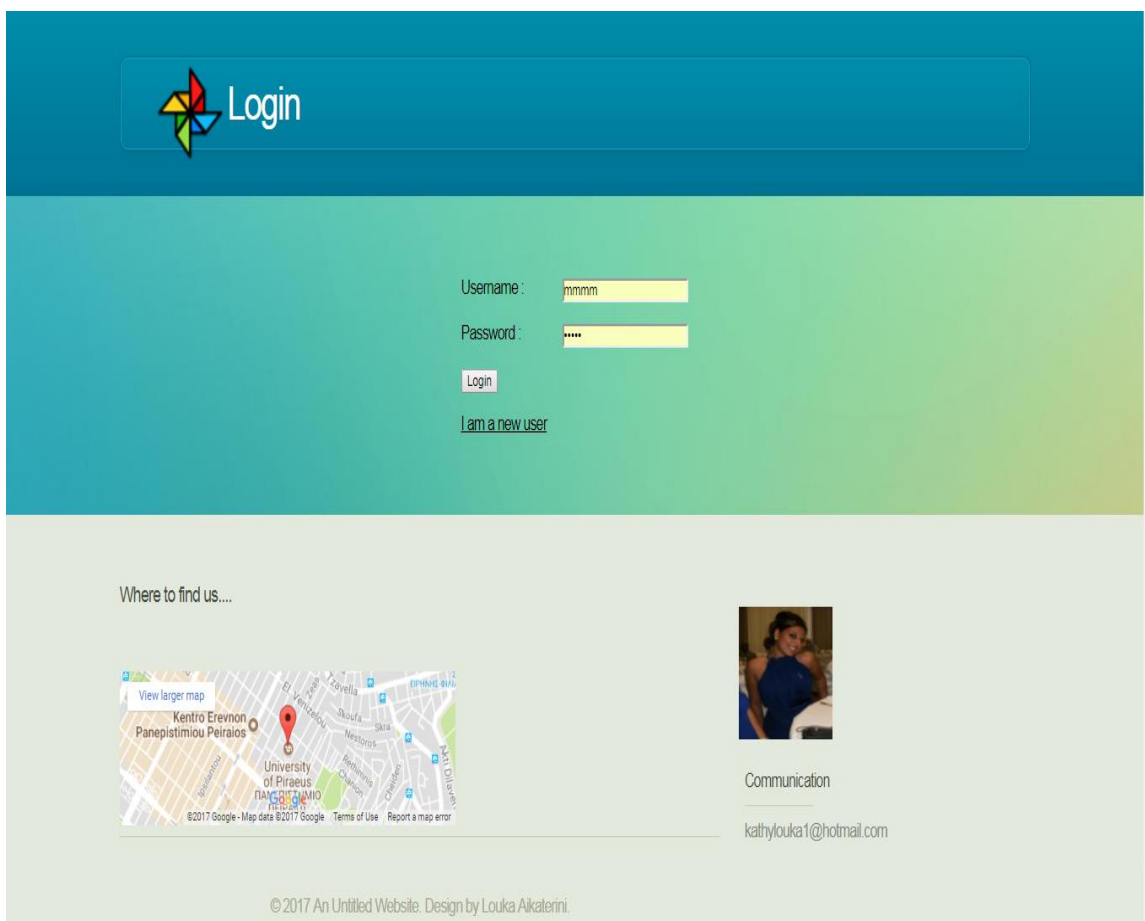
14 Εύρεση Check In από τον χάρτη

### 3 Παρουσίαση και χρήση site

#### 3.1 Web εφαρμογή MemoryTracker

##### 3.1.1 Είσοδος και εγγραφή στην εφαρμογή

- ✓ Είσοδος στην Web εφαρμογή χρησιμοποιώντας το συνθηματικό και τον κωδικό.



15 Εισαγωγή χρήστη στην Web εφαρμογή



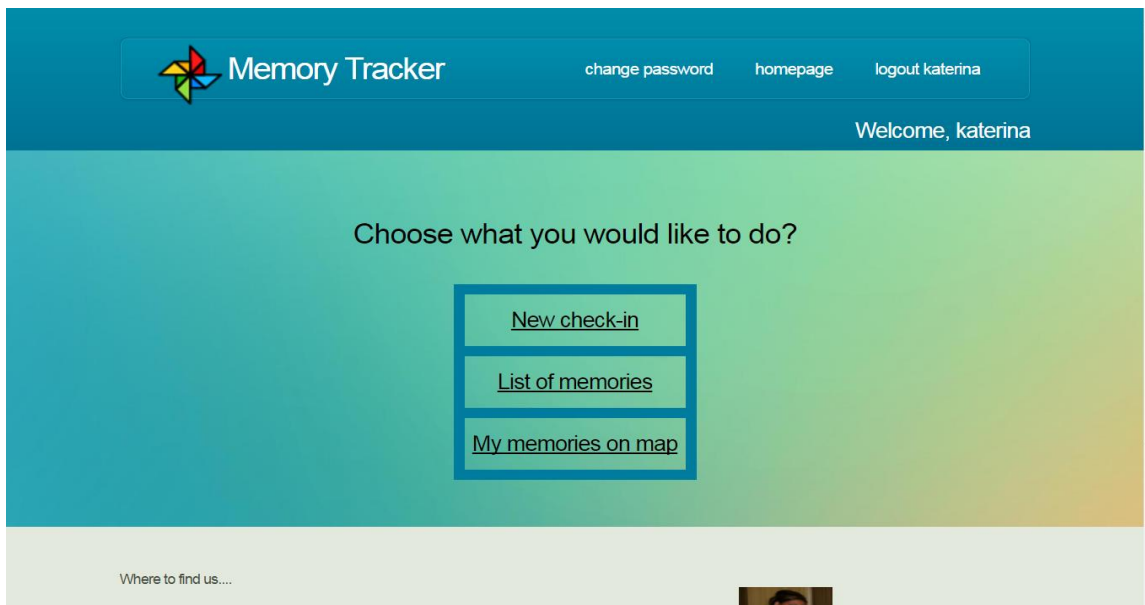
- ✓ Εάν ο χρήστης δεν έχει λογαριασμό, τότε μπορεί να δημιουργήσει έναν πατώντας το «I am a new user». Στην παρακάτω εικόνα απεικονίζεται η δημιουργία νέου χρήστη. Ο χρήστης πρέπει να συμπληρώσει όλα τα απαραίτητα στοιχεία και ο κωδικός του να περιέχει τουλάχιστον ένα ειδικό χαρακτήρα. Στη συνέχεια επιστρέφει στην αρχική οθόνη του «Login» και εισάγει τον όνομα χρήστη και τον κωδικό, έτσι ώστε να έχει πρόσβαση στην Ιστοσελίδα.

The screenshot shows a web application interface for creating a new user. The header is blue with a logo and the text 'New User' and a 'homepage' link. The main content area is green and contains a registration form with the following fields: Name, Surname, Username (with 'mmmm' entered), Password (with '....' entered), E-mail, Gender (radio buttons for Female and Male), and Age. A 'Submit' button is at the bottom of the form, and a 'Go to homepage' link is below it. The footer contains a map and a small profile picture.

16 Νέος χρήστης στην Web εφαρμογή

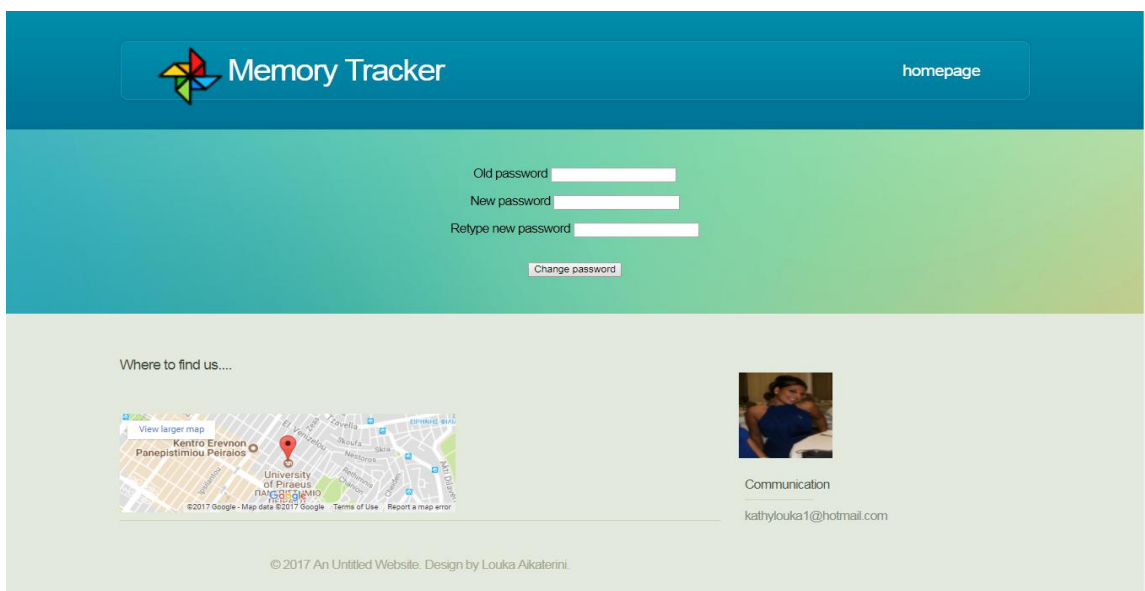
### 3.1.2 Αλλαγή κωδικού χρήστη

- ✓ Αρχική οθόνη Web εφαρμογής. Ο χρήστης μπορεί να αλλάξει τον κωδικό, πατώντας πάνω στο «Change Password».



17 Αρχική οθόνη Web εφαρμογής

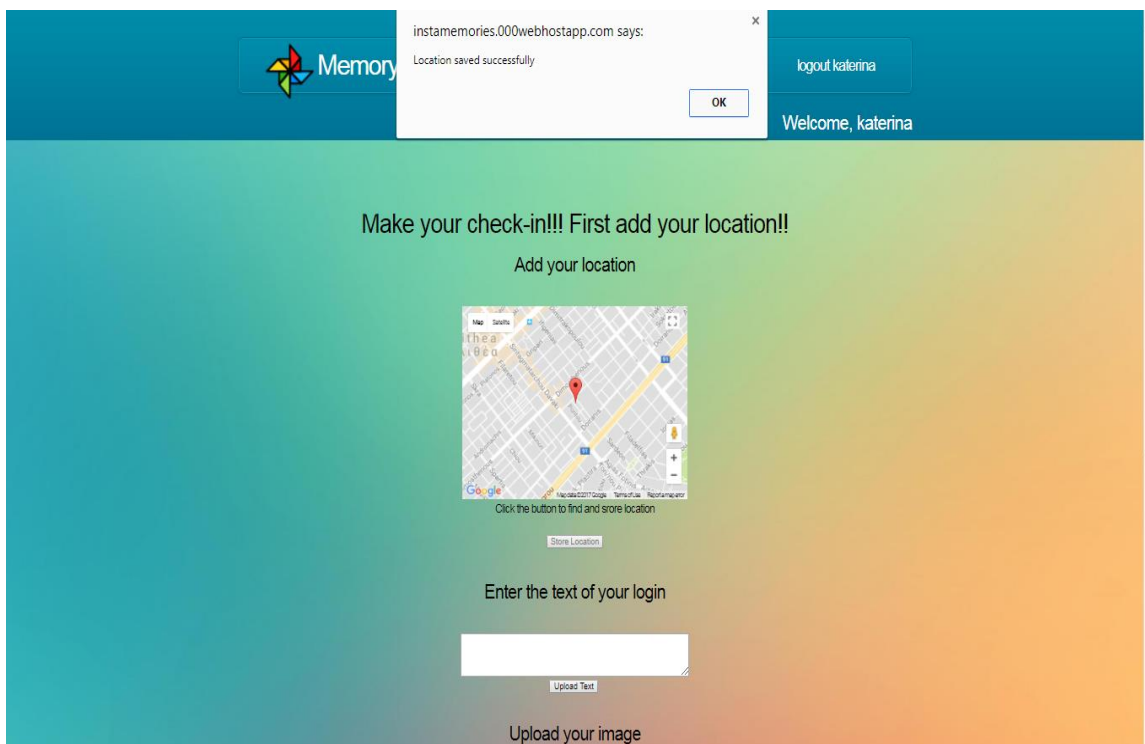
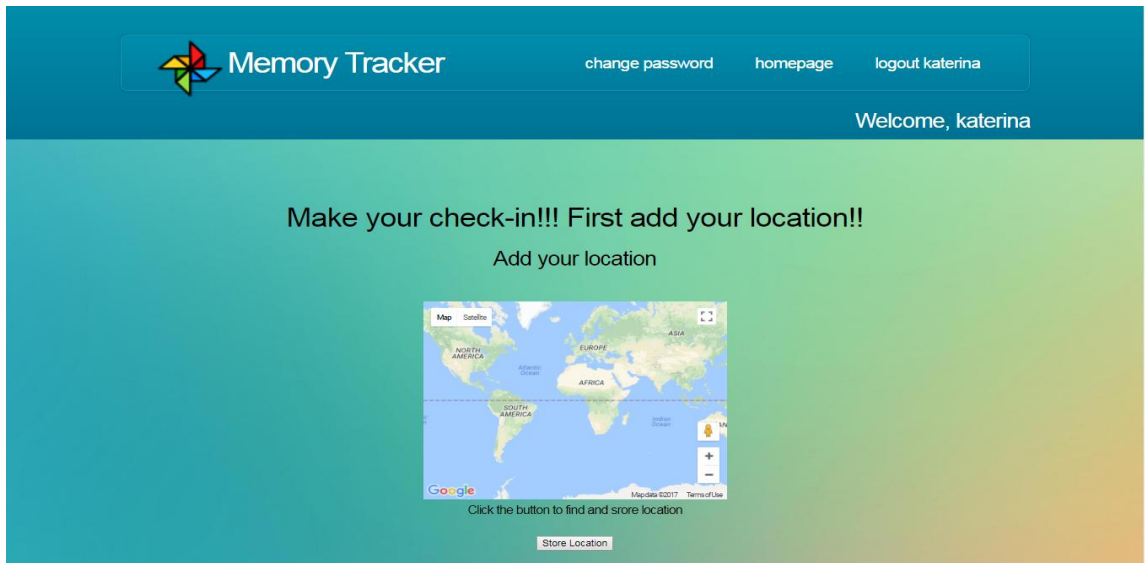
- ✓ Αλλαγή κωδικού πρόσβασης. Ο χρήστης πρέπει να εισάγει τον παλιό κωδικό πρόσβασης στο πεδίο «Old Password». Στο δεύτερο πεδίο πρέπει να εισάγει το νέο κωδικό και να τον επαναλάβει στο τρίτο πεδίο επίσης.



18 Αλλαγή κωδικού πρόσβασης Web εφαρμογής

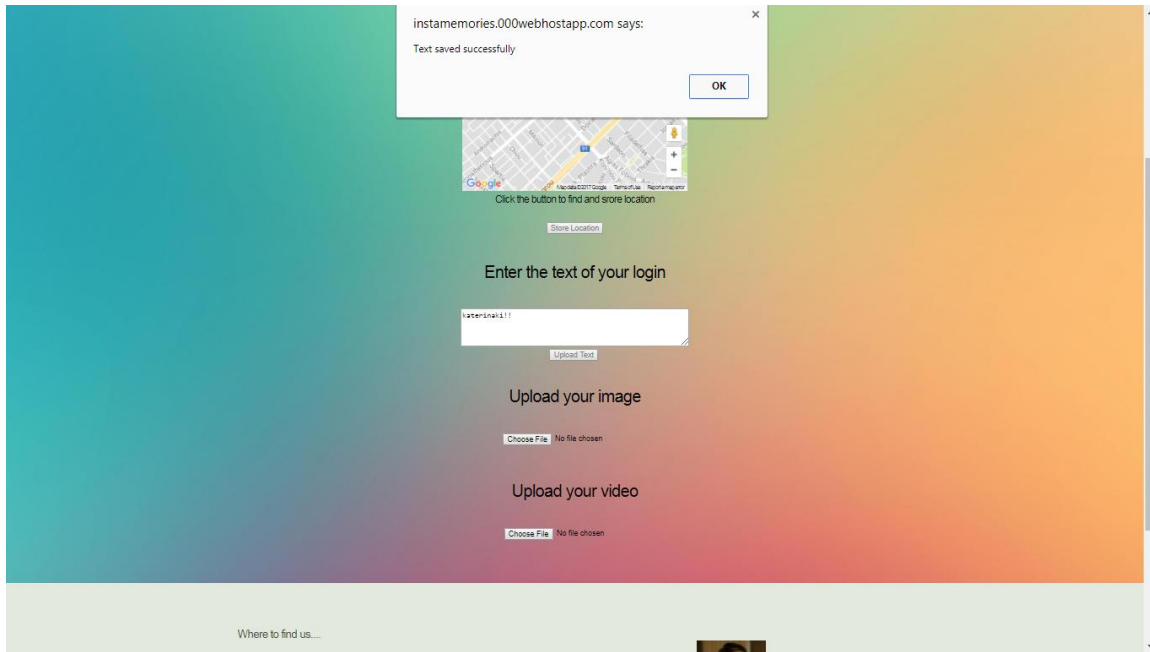
## 3.1.3 New Check-in

- ✓ New Check-in. Αρχικά ο χρήστης πρέπει να πατήσει το κουμπί «Store Location» έτσι ώστε να αποθηκευτεί η διεύθυνση στην οποία βρίσκεται ο υπολογιστής του χρήστη. Έπειτα ενεργοποιείται η αποθήκευση σημειώσεων, εικόνας και βίντεο.

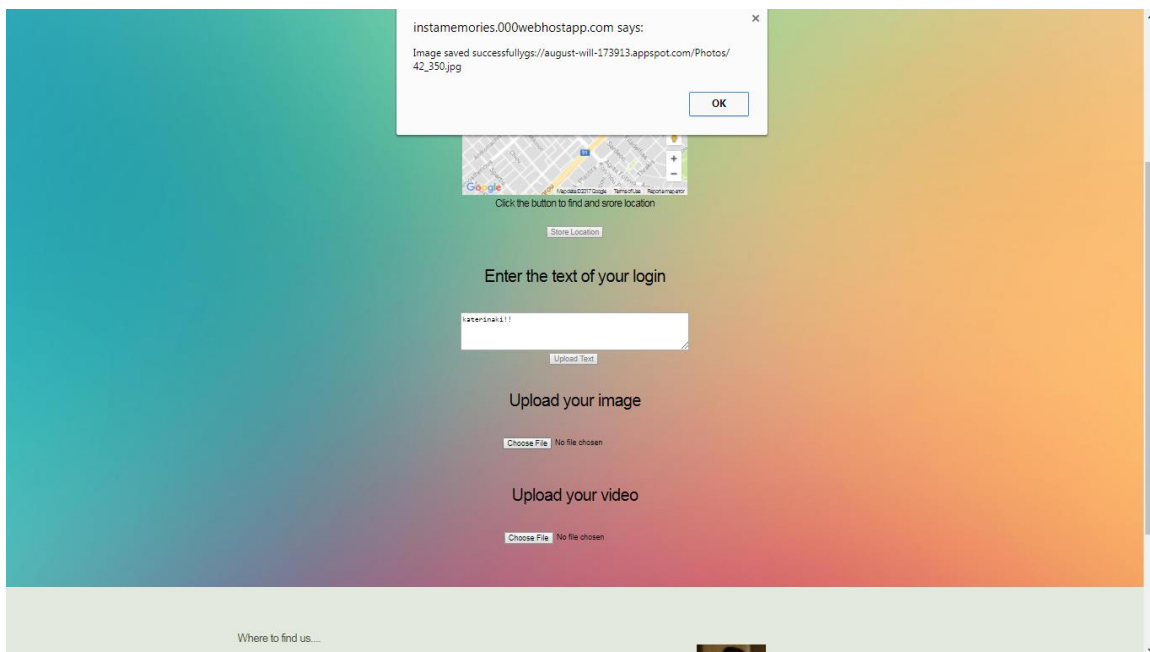


19 Αποθήκευση τοποθεσίας Web εφαρμογής

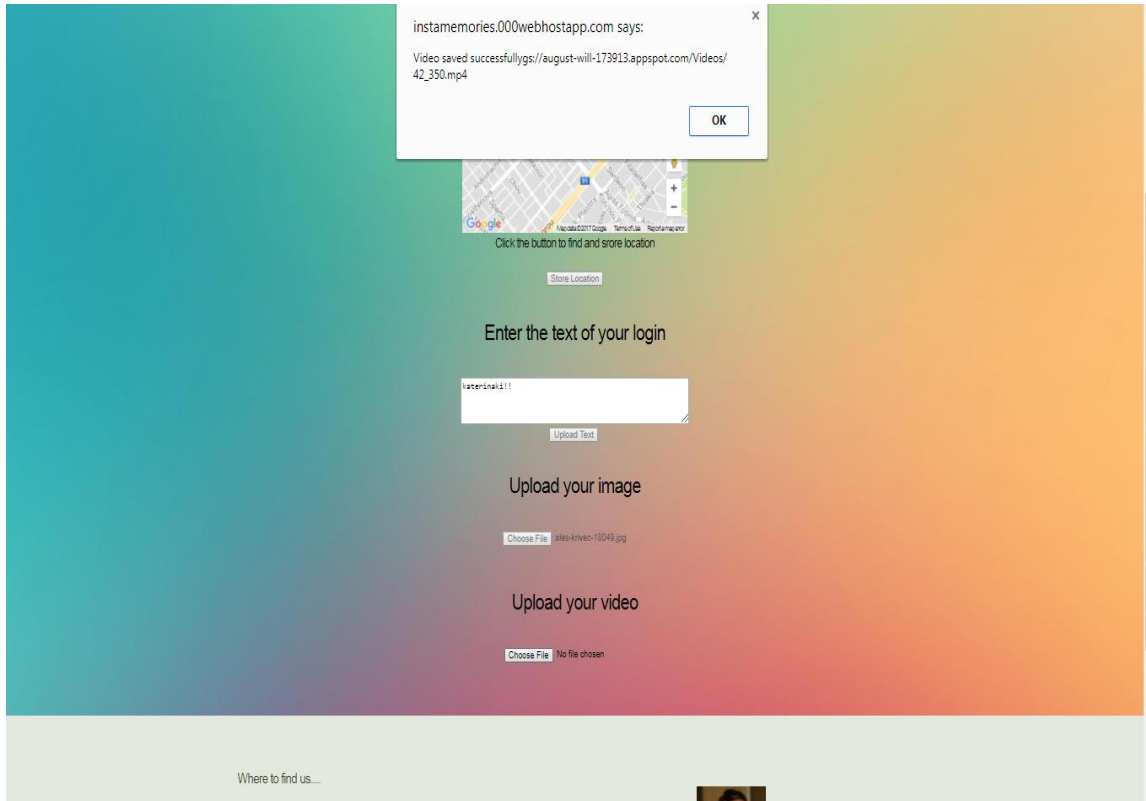
- ✓ Έπειτα ο χρήστης μπορεί να αποθηκεύσει προαιρετικά σημειώσεις, εικόνα και βίντεο για το Check-in του.



20 Αποθήκευση σημειώσεων Web εφαρμογή



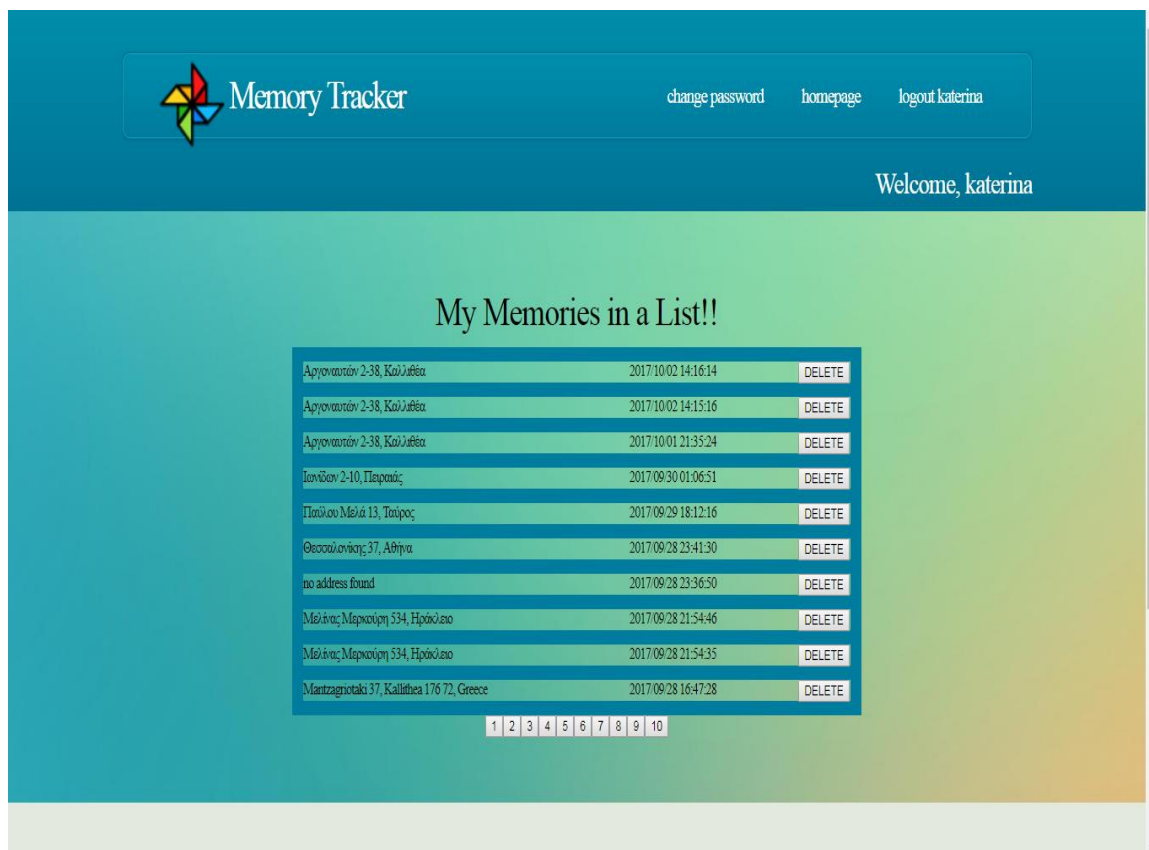
21 Αποθήκευση εικόνας Web εφαρμογή



22 Αποθήκευση βίντεο Web εφαρμογή

### 3.1.4 List of Memories

- ✓ Ο χρήστης, πατώντας το «List of Memories» της αρχικής οθόνης, μπορεί να δει όλα τα checkin τα οποία έχει κάνει από το πιο πρόσφατο στο πιο παλιό, σε μία λίστα. Στην πρώτη στήλη εμφανίζεται η διεύθυνση της τοποθεσίας, στην δεύτερη η ημερομηνία, και στην Τρίτη το delete κουμπί με το οποίο ο χρήστης μπορεί να διαγράψει οριστικά το checkin του. Τέλος, κάτω εμφανίζονται οι σελίδες στις οποίες ο χρήστης μπορεί να αναζητήσει αυτό που ψάχνει σύμφωνα με την ημερομηνία καθώς και με την διεύθυνση.

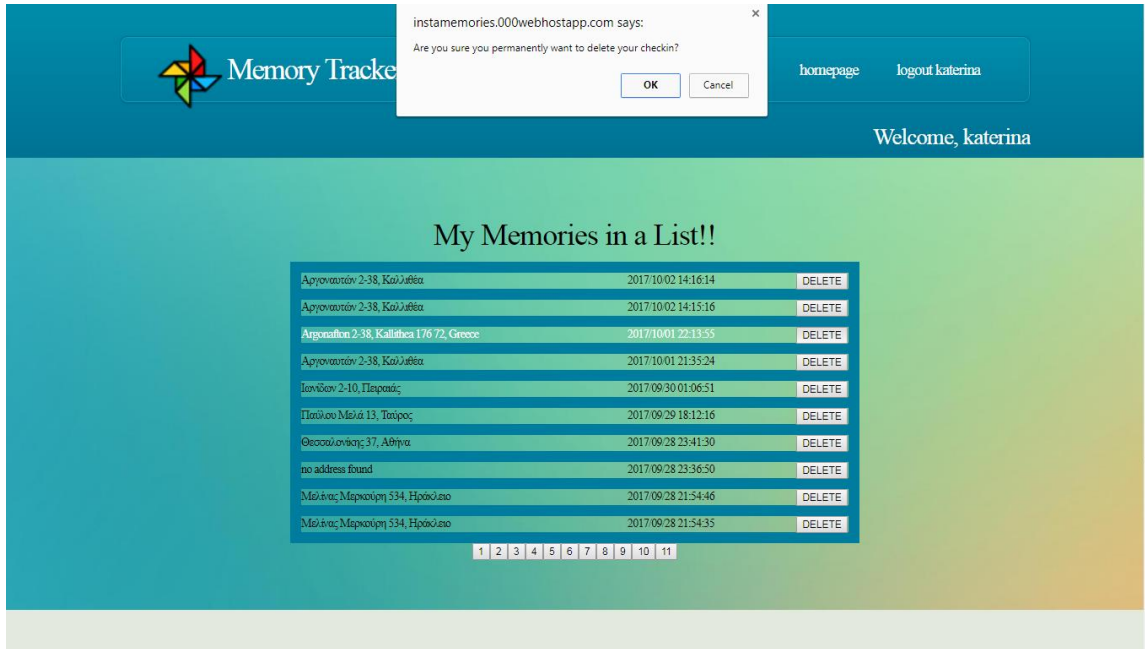


The screenshot shows the Memory Tracker application interface. At the top, there is a navigation bar with the logo, the text "Memory Tracker", and links for "change password", "homepage", and "logout katerina". Below the navigation bar, it says "Welcome, katerina". The main content area is titled "My Memories in a List!!" and displays a table of memories. Each row contains the location, the date and time, and a "DELETE" button. At the bottom of the table, there is a pagination control showing numbers 1 through 10.

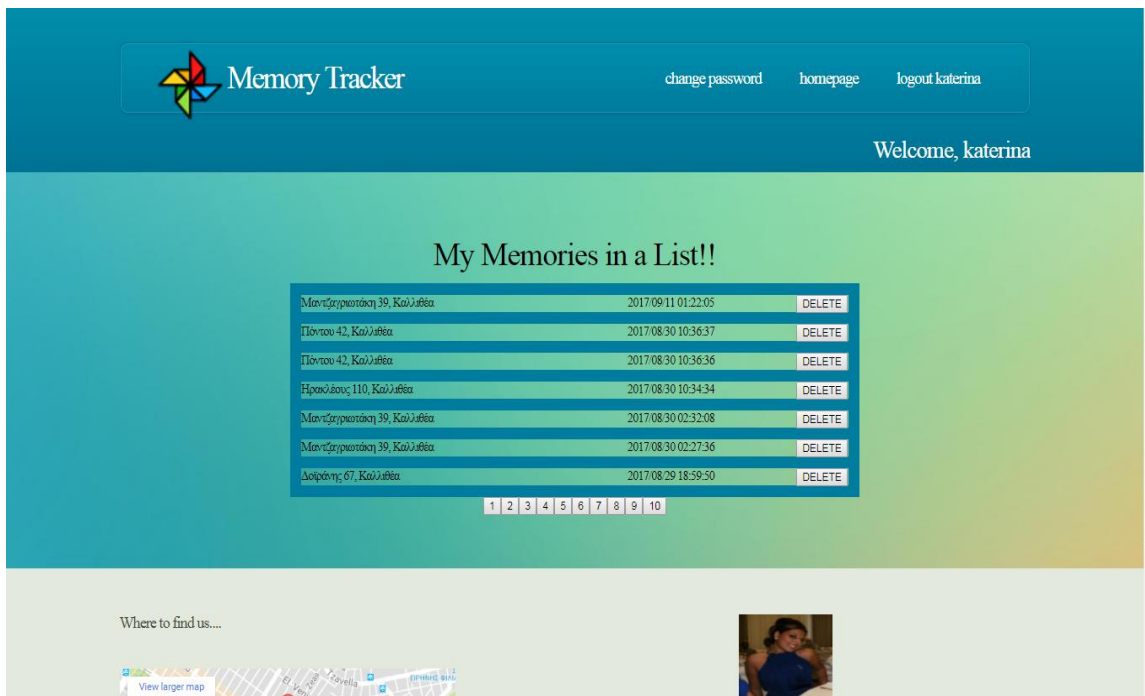
Location	Date and Time	Action
Αργονατών 2-38, Καλλιθέα	2017/10/02 14:16:14	DELETE
Αργονατών 2-38, Καλλιθέα	2017/10/02 14:15:16	DELETE
Αργονατών 2-38, Καλλιθέα	2017/10/01 21:35:24	DELETE
Ιονίων 2-10, Πειραιάς	2017/09/30 01:06:51	DELETE
Παύλου Μελά 13, Τεόρας	2017/09/29 18:12:16	DELETE
Θεσσαλονίκης 37, Αθήνα	2017/09/28 23:41:30	DELETE
no address found	2017/09/28 23:36:50	DELETE
Μελάτης Μαρκαρή 534, Ηράκλειο	2017/09/28 21:54:46	DELETE
Μελάτης Μαρκαρή 534, Ηράκλειο	2017/09/28 21:54:35	DELETE
Mantzagniotaki 37, Kallithea 176 72, Greece	2017/09/28 16:47:28	DELETE

1 2 3 4 5 6 7 8 9 10

✓ Διαγραφή Check-in



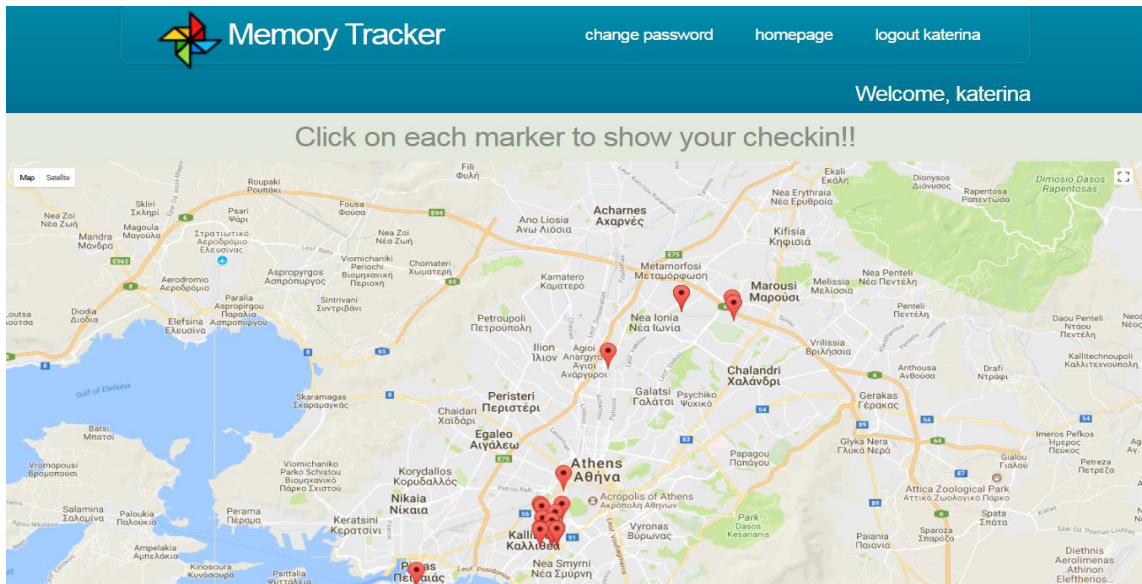
23 Διαγραφή Check-in Web εφαρμογή



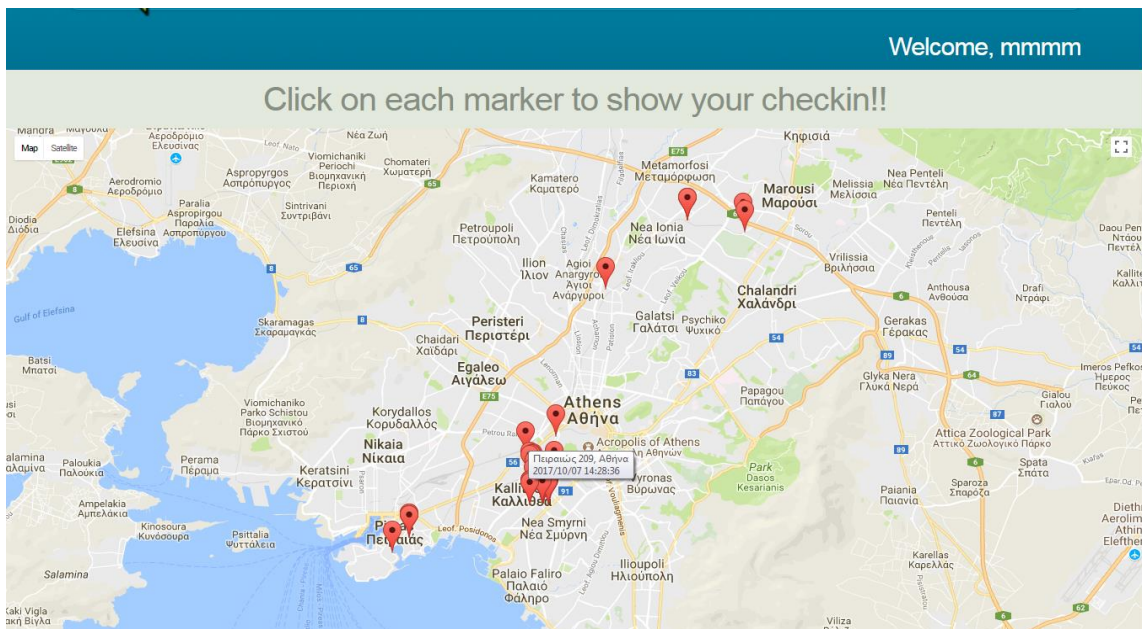
24 Τελευταία σελίδα λίστας

3.1.5 Memories on Map

- ✓ Πατώντας πάνω στην επιλογή «Memories on Map», ο χρήστης μπορεί να δει όλα τα μέρη που έχει επισκεφθεί κεντραρισμένα στον χάρτη. Βάζοντας τον κέρσορα πάνω σε έναν δείκτη, εμφανίζεται η διεύθυνση. Πατώντας πάνω στον δείκτη, ο χρήστης μπορεί να μπει να δει όλα τα αποθηκευμένα δεδομένα του check-in του.



25 Χάρτης με τις τοποθεσίες Web εφαρμογή

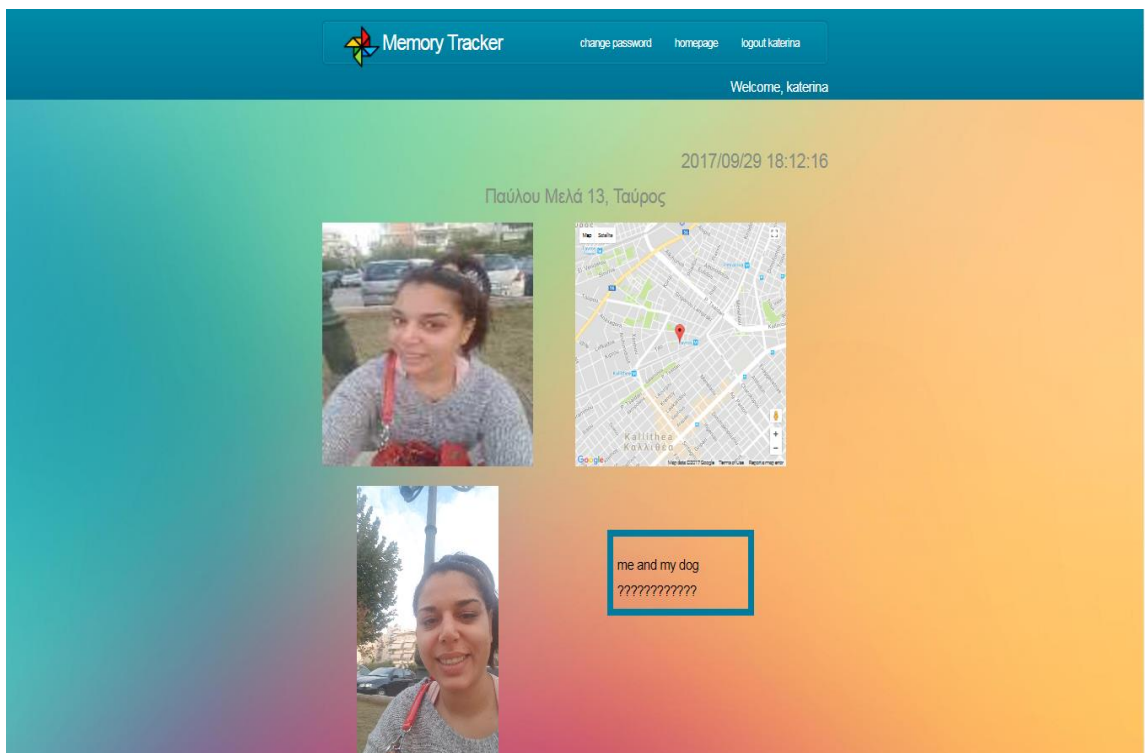


26 Εμφάνιση διεύθυνσης Web εφαρμογή



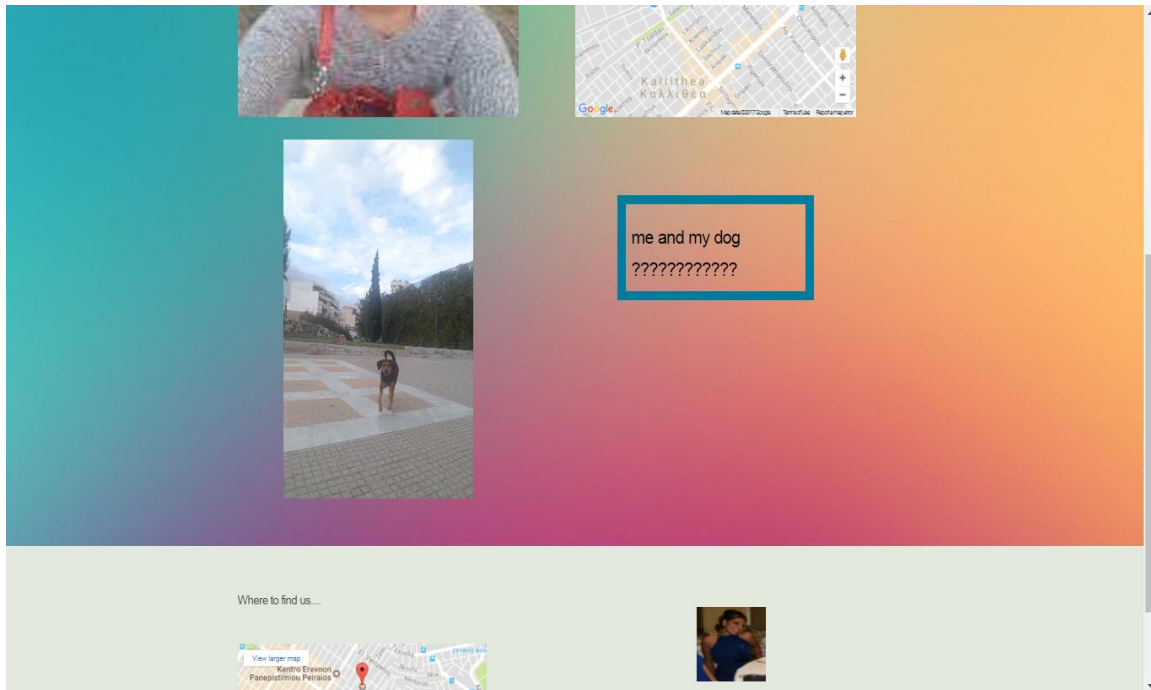
### 3.1.6 Εμφάνιση Check-in

- ✓ Ο χρήστης μπορεί να δει το επιλεγμένο checkin, το οποίο έχει κάνει είτε από την εφαρμογή Android, είτε από την Web εφαρμογή με δύο τρόπους.
  - Είτε από τον χάρτη βρίσκοντας το σημείο το οποίο ήταν και πατώντας πάνω στον δείκτη.
  - Είτε από την λίστα αναζητώντας το στις σελίδες με βάση την ημερομηνία και την διεύθυνση.
  
- ✓ Βρίσκοντάς το, εμφανίζεται η ημερομηνία, η διεύθυνση, η φωτογραφία, το βίντεο, οι σημειώσεις καθώς και η διεύθυνση στον χάρτη. Σε περίπτωση που δεν υπάρχει εικόνα, η εφαρμογή εμφανίζει μια random εικόνα.

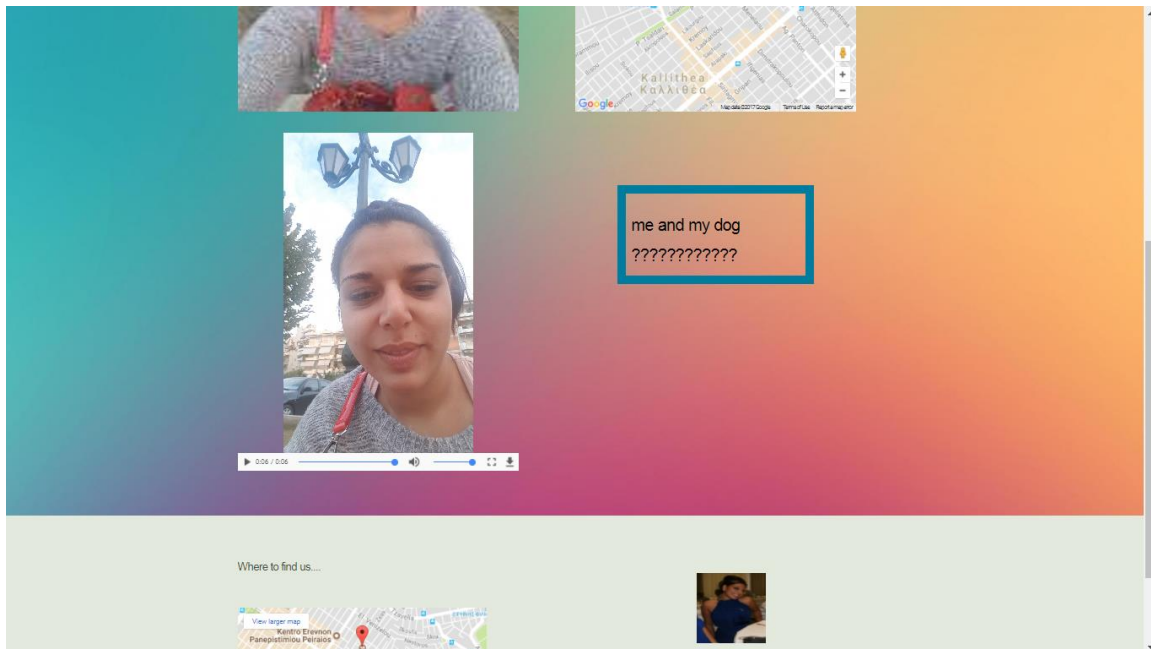


27 Εμφάνιση Check-in Web εφαρμογής

- ✓ Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει στον υπολογιστή την εικόνα και το βίντεο.



28 Εμφάνιση βίντεο Web εφαρμογής



29 Δυνατότητα αποθήκευσης βίντεο Web εφαρμογής

## 4 Αρχιτεκτονική Συστήματος

Σε αυτή την ενότητα θα γίνει η ανάλυση της αρχιτεκτονικής της εφαρμογής . Θα παρουσιαστεί ο τρόπος με τον οποίο υλοποιήθηκε η εφαρμογή, ποια εργαλεία χρησιμοποιήθηκαν, ποια γλώσσα προγραμματισμού, ποια βάση δεδομένων καθώς και μεθοδολογίες ανάπτυξης κώδικα.

Οι δύο εφαρμογές χρησιμοποιούν την ίδια βάση δεδομένων, καθώς και τον ίδιο server, έτσι ώστε να επιτυγχάνεται η επικοινωνία μεταξύ τους.

Βάση δεδομένων :

```
CREATE TABLE `audios` (  
  `id` int(255) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `location_id` int(255) NOT NULL,  
  `audio` varchar(255) COLLATE utf8_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
CREATE TABLE `markers` (  
  `id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `name` varchar(60) CHARACTER SET utf8 NOT NULL,  
  `address` varchar(80) COLLATE utf8_unicode_ci NOT NULL,  
  `lat` float(10,6) NOT NULL,  
  `lng` float(10,6) NOT NULL,  
  `my_date` text COLLATE utf8_unicode_ci NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
CREATE TABLE `notes` (  
  `id` int(255) NOT NULL,  
  `user_id` int(100) NOT NULL,  
  `location_id` int(255) NOT NULL,  
  `notes` text COLLATE utf8_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

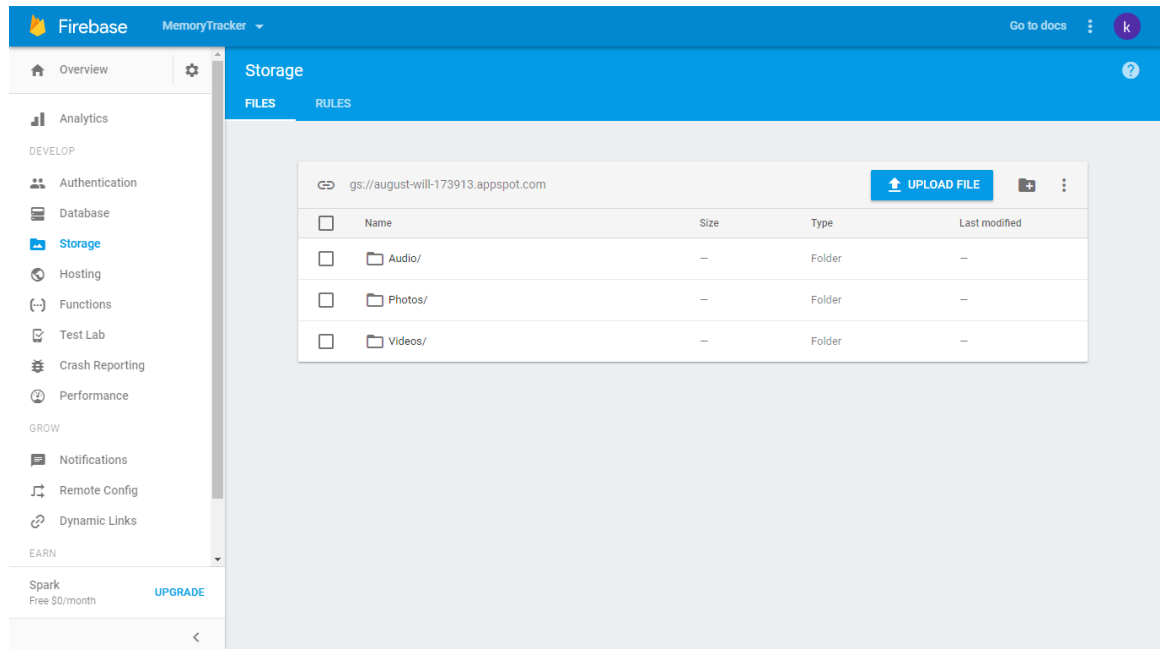
```
CREATE TABLE `photos` (  
  `id` int(255) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `location_id` int(255) NOT NULL,  
  `photo` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `surname` varchar(255) NOT NULL,  
  `username` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `gender` varchar(255) NOT NULL,  
  `age` int(255) NOT NULL,  
  `last_location` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `videos` (  
  `id` int(255) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `location_id` int(255) NOT NULL,  
  `video` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Παρακάτω θα αναλύσουμε με περισσότερη λεπτομέρεια σχετικά με το τι έχει υλοποιηθεί σε κάθε κομμάτι ξεχωριστά:

#### 4.1 MemoryTracker εφαρμογή Android

- ✚ Η εφαρμογή του κινητού έχει δημιουργηθεί στο Android Studio 2.3.3. Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android.
- ✚ Οι γλώσσες προγραμματισμού που χρησιμοποιούνται είναι Java, XML και PHP για σύνδεση με τη βάση.
- ✚ Χρησιμοποιήθηκε το Google Maps της Google play για την εμφάνιση χάρτη καθώς και για την εύρεση τοποθεσίας των χρηστών.
- ✚ Η εφαρμογή χρειάζεται permissions για την χρήση τοποθεσίας, για το μικρόφωνο καθώς και για τον αποθηκευτικό χώρο.
- ✚ Οι δοκιμές πραγματοποιήθηκαν στο Samsung Galaxy S7.
- ✚ Ο server ο οποίος χρησιμοποιήθηκε είναι ο [www.000webhost.com](http://www.000webhost.com), ο οποίος είναι δωρεάν.
- ✚ Η αποθήκευση των εικόνων, των βίντεο καθώς και των ηχογραφήσεων, πραγματοποιούνται στο Firebase Storage.
- ✚ Η αποθήκευση Firebase παρέχει ασφαλείς λήψεις και λήψεις αρχείων για εφαρμογές Firebase, ανεξάρτητα από την ποιότητα του δικτύου. Ο προγραμματιστής μπορεί να το χρησιμοποιήσει για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργεί ο χρήστης. Η αποθήκευση Firebase υποστηρίζεται από το Google Cloud Storage.



30 Firebase Storage

Παρακάτω παρατίθεται ο κώδικας:

- ❖ Για την αποθήκευση δεδομένων στην βάση, έχουμε την κλάση `BackgroundWorker.java`, η οποία χρησιμοποιεί `URLConnection`.

```
String type = params[0];
String login_url =
"http://instamemories.000webhostapp.com/insta_memories/login_app.php";
String register_url =
"http://instamemories.000webhostapp.com/insta_memories/new_user_app.php";
if (type.equals("submit")) {
    try {
        String username = params[1];
        String password = params[2];
        URL url = new URL(login_url);
        HttpURLConnection httpURLConnection =
        (HttpURLConnection)url.openConnection();
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setDoInput(true);
        OutputStream outputStream = httpURLConnection.getOutputStream();
        BufferedWriter bufferedWriter = new BufferedWriter(new
        OutputStreamWriter(outputStream, "UTF-8"));
        String post_data = URLEncoder.encode("username", "UTF-
        8")+"="+URLEncoder.encode(username, "UTF-8")+"&"
        +URLEncoder.encode("password", "UTF-
        8")+"="+URLEncoder.encode(password, "UTF-8");
        bufferedWriter.write(post_data);
```

```

        bufferedWriter.flush();
        bufferedWriter.close();
        outputStream.close();
        InputStream inputStream = httpURLConnection.getInputStream();
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "ISO-8859-1"));
        String result = "";
        String line="";
        while((line = bufferedReader.readLine()) != null){
            result += line;
        }
        String test_result = showJSON(result);
        Log.i("JsonResult", result);
        bufferedReader.close();
        inputStream.close();
        httpURLConnection.disconnect();
        return test_result;

    }catch (MalformedURLException e){
        e.printStackTrace();
    }catch (IOException e){
        e.printStackTrace();
    }
}

```

- ❖ Τα δεδομένα αποθηκεύονται στην κλάση ως `public static` έτσι ώστε να μπορούν να χρησιμοποιούνται και από τις υπόλοιπες κλάσεις.

```

public static String id;
public static String message;
public static JSONArray result;
public static String message_notes;
public static String message_result;

```

- ❖ Στην κλάση `ListOfMemories.java` επιστρέφονται τα δεδομένα με μορφή JSON array.

```

private String showJSON(String response){
    try {
        JSONObject jsonObject = new JSONObject(response);
        result = jsonObject.getJSONArray("server_response");
        JSONObject collegeData = result.getJSONObject(0);
        message = collegeData.getString("message");

    } catch (JSONException e) {
        e.printStackTrace();
    }

    return message;
}

```

- ❖ Για την λειτουργία του check-in, χρησιμοποιήθηκε η κλάση **GPSTracker**, για την εύρεση τοποθεσίας.

```
gps = new GPSTracker(CheckIn.this);
if(gps.canGetLocation()){

    Geocoder geocoder = new Geocoder(getApplicationContext(),
    Locale.getDefault());
    String address = "no address found";

    latitude = gps.getLatitude();
    longitude = gps.getLongitude();

    try {
        List<Address> listAddresses = geocoder.getFromLocation(latitude,
        longitude, 1);

        if (listAddresses != null && listAddresses.size() > 0) {
            if (listAddresses.get(0).getThoroughfare() != null) {
                if (listAddresses.get(0).getSubThoroughfare() != null) {
                    address = listAddresses.get(0).getThoroughfare();
                }
                address += " " + listAddresses.get(0).getSubThoroughfare();
            }
            if (listAddresses.get(0).getLocality() != null){
                address += ", " + listAddresses.get(0).getLocality();
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- ❖ Για την αποθήκευση της ηχογράφησης, χρησιμοποιήθηκε το **MediaRecorder**.

```
private void startRecording() {
    try {
        mRecorder = new MediaRecorder();
        mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        mRecorder.setOutputFile(mFileName);
        mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);

        mRecorder.prepare();
    } catch (IOException e) {
        Log.e(LOG_TAG, "prepare() failed");
    }

    mRecorder.start();
}
private void stopRecording() {
    try {
        mRecorder.stop();
        mRecorder.release();
        mRecorder = null;
    } catch (Exception e) {
        Log.e(LOG_TAG, "failed");
    }
}}
```

- ❖ Η κλάση `CustomAdapter.java` χρησιμοποιείται για να εμφανίζονται δύο διαφορετικά χρώματα στην λίστα.

```
public View getView(int position, View convertView, ViewGroup parent) {
    View row = super.getView(position, convertView, parent);

    if (position % 2 == 1) {
        row.setBackgroundColor(0xFF99F1E7);
    }
    else if (position % 2 == 0) {
        row.setBackgroundColor(0xFF38B7FC);
    }
    return row;
}
```

- ❖ Προκειμένου να επιτευχθεί η αποθήκευση, χρησιμοποιήθηκε ένας Handler με `delayTime = 3000`.

```
private Runnable mLaunchTask1 = new Runnable() {
    public void run() {
        mProgress.dismiss();
        if (BackgroundWorkerSaveFiles.message_notes.matches("Your notes has
        been stored")) {
            addNotesImageButton.setEnabled(false);
            textViewMyNotes.setVisibility(View.VISIBLE);
            textViewMyNotes.setText(textNotes_save);
            Toast.makeText(CheckIn.this,
            BackgroundWorkerSaveFiles.message_notes, Toast.LENGTH_SHORT).show();
        } else{
            Toast.makeText(CheckIn.this,
            BackgroundWorkerSaveFiles.message_notes, Toast.LENGTH_SHORT).show();
        }
    }
};
```

```
btnSave.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        textNotes_save = editText.getText().toString();
        dialog.dismiss();
        mProgress.setMessage("Saving Notes...");
        mProgress.show();
        backgroundWorkerSaveFiles.execute(type1, BackgroundWorker.id,
        BackgroundWorkerSaveFiles.id, textNotes_save);
        mHandler1.postDelayed(mLaunchTask1, myDelayTime);
    }
});
```



- ❖ Η αποθήκευση εικόνας, ήχου και βίντεο, πραγματοποιείται στο **Firebase Storage**.

```
private StorageReference mStorage;
StorageReference filepath = mStorage.child("Audio").child(BackgroundWorker.id +
"_" + BackgroundWorkerSaveFiles.id + ".mp3");
Uri uri = Uri.fromFile(new File(mFileName));

filepath.putFile(uri).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

        mProgress.setMessage("Saving Audio...");
        mProgress.show();
        backgroundWorkerSaveFiles.execute(type2, BackgroundWorker.id,
BackgroundWorkerSaveFiles.id, all_in_one);
        mHandler2.postDelayed(mLaunchTask2, myDelayTime);
    }
});
```

- ❖ Στο **FocusOnEachMemory**, η μεγέθυνση του βίντεο γίνεται με τον παρακάτω κώδικα.

```
fullScreen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showNotesTextView.setVisibility(View.INVISIBLE);
        showAddressTextView.setVisibility(View.INVISIBLE);
        showDateTextView.setVisibility(View.INVISIBLE);
        showImageView.setVisibility(View.INVISIBLE);
        myMap.setVisibility(View.INVISIBLE);
        seekBar.setVisibility(View.INVISIBLE);
        buttonPlay.setVisibility(View.INVISIBLE);
        showVideoView.setLayoutParams(new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.MATCH_PARENT,
RelativeLayout.LayoutParams.MATCH_PARENT));
    }
});
```

- ❖ Χρησιμοποιείται επίσης `getIntent().getExtras()` για να σταλούν τα δεδομένα είτε από το `MapsActivity`, είτε από το `ListOfMemoriesActivity`.

```
Bundle bundle = getIntent().getExtras();
JSONArray jsonArr_last = new JSONArray();
if (bundle.getString("activity") == "map") {
    jsonArr_last = MapActivity.jsonArr;
}else{
    jsonArr_last = ListOfMemories.jsonArr;
}
```

- ❖ Εμφάνιση εικόνας.

```
Glide.with(this /* context */)
    .using(new FirebaseImageLoader())
    .load(pathReference)
    .error(R.drawable.add_image)
    .into(showImageView);
```

- ❖ Εμφάνιση βίντεο.

```
private void fetchVideoUrlFromFirebase() {
    final FirebaseStorage storage = FirebaseStorage.getInstance();
    StorageReference storageRef = storage.getReferenceFromUrl("gs://august-will-173913.appspot.com");
    StorageReference pathReference_audio =
    storageRef.child("Videos/"+user_id+"_"+id+".mp4");
    pathReference_audio.getDownloadUrl().addOnSuccessListener(new
    OnSuccessListener<Uri>() {
        @Override
        public void onSuccess(final Uri uri) {
            try {
                showVideoView.setVisibility(View.VISIBLE);
                playVideo.setVisibility(View.VISIBLE);
                fullScreen.setVisibility(View.VISIBLE);
                playVideo.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        final String url = uri.toString();
                        Uri uri_video = Uri.parse(url);
                        showVideoView.setVideoURI(uri_video);
                        showVideoView.setMediaController(mediaController);
                        mediaController.setAnchorView(showVideoView);
                        showVideoView.start();
                    }
                });
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

```

        });

    } catch (Exception e) {
        e.printStackTrace();
    }
}

})

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.i("error", e.getMessage());
    }
});
}

```

### ❖ Εμφάνιση ListOfMemories

```

if (jArray != null) {
    for (int i=0;i<jArray.length();i++){
        JSONObject jsnobject = new
JSONObject(jArray.getString(i));
        places.add(jsnobject.getString("address"));
    }
    mProgress.dismiss();

    CustomAdapter myAdapter=new
CustomAdapter(ListOfMemories.this,android.R.layout.simple_list_item_1,places);
    listView.setAdapter(myAdapter);

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

            Intent intent = new Intent(getApplicationContext(),
FocunOnEachMemory.class);

            intent.putExtra("clicked_id", position);
            intent.putExtra("activity","list");
            startActivity(intent);
        }
    });
}

```

## ❖ Εμφάνιση δεικτών στον χάρτη

```
JSONObject rec = result.getJSONObject(0);
String array_result = rec.getString("res");
JSONArray jsonArr = new JSONArray(array_result);
LatLngBounds.Builder builder = new LatLngBounds.Builder();
mProgress.dismiss();
for (int i = 0 ; i < jsonArr.length(); i++)
{
    final JSONObject jsonObj = jsonArr.getJSONObject(i);
    HashMap<String, String> location_ = new HashMap<String, String>();
    id = jsonObj.getString("id");
    user_id = jsonObj.getString("user_id");
    name = jsonObj.getString("name");
    address = jsonObj.getString("address");
    lat1 = jsonObj.getString("lat");
    lng1 = jsonObj.getString("lng");
    my_date = jsonObj.getString("my_date");
    Double lat = Double.parseDouble(lat1);
    Double lng = Double.parseDouble(lng1);
    LatLng userLocationTest = new LatLng(lat, lng);
    marker = mMap.addMarker(new MarkerOptions()
        .position(userLocationTest)
        .title(my_date)
        .snippet(address));
    builder.include(marker.getPosition());

    markers.add(marker);
    mMap.addMarker(new
MarkerOptions().position(userLocationTest).title(address));
    mMap.setOnInfoWindowClickListener(new GoogleMap.OnInfoWindowClickListener()
    {
        @Override
        public void onInfoWindowClick(Marker marker) {
            onMarkerClick(marker);
        }
    });
}
} LatLngBounds bounds = builder.build();
int padding = 0; // offset from edges of the map in pixels
CameraUpdate cu = CameraUpdateFactory.newLatLngBounds(bounds, padding);
mMap.animateCamera(cu);
```

## 4.2 MemoryTracker Web εφαρμογή

- ✦ Η εφαρμογή του κινητού έχει δημιουργηθεί στο Sublime Text. Ένας εκλεπτυσμένος επεξεργαστής κειμένου για κώδικα, σήμανση και πεζογραφία.
- ✦ Οι γλώσσες προγραμματισμού που χρησιμοποιούνται είναι PHP, HTML, CSS, JAVASCRIPT, AJAX και PHP.
- ✦ Ο server ο οποίος χρησιμοποιήθηκε είναι ο [www.000webhost.com](http://www.000webhost.com), ο οποίος είναι δωρεάν.
- ✦ Η αποθήκευση των εικόνων, καθώς και των βίντεο πραγματοποιούνται στο Firebase Storage.

### Παρακάτω παρουσιάζεται ο κώδικας:

- ❖ **Η λίστα εμφανίζεται με Javascript. Εμφανίζονται 10 δεδομένα κάθε φορά.**

```
function get_array(clicked_id){  
    var my_table = document.getElementById('my_table');  
  
    $("#my_table tr").remove();  
    // var clicked_id = clicked_id;  
    var result = <?= $json; ?>;  
    var my_mode = result.length % 10;  
    var my_div = Math.ceil(result.length / 10);  
    if (clicked_id == my_div-1){  
        var s = 0;  
        var first = 10 * clicked_id;  
        var last = 10 * clicked_id + my_mode-1;  
        for (var i = first; i < last+1; i++) {  
  
            var row = my_table.insertRow(s);  
            row.className= "format";  
            row.id = result[i]['id'];  
            row.onmouseover = function() {ChangeBackgroundColor(this)};  
            row.onmouseout = function() {RestoreBackgroundColor(this)};
```

```
// row.onclick = function() {click_row(this.id)};

var cell1 = row.insertCell(0);
var cell2 = row.insertCell(1);
var cell3 = row.insertCell(2);

row.cells[0].id = result[i]['id'];
row.cells[1].id = result[i]['id'];
row.cells[0].onclick = function() {click_row(this.id)};
row.cells[1].onclick = function() {click_row(this.id)};
cell2.style.textAlign = "center";

cell1.innerHTML = result[i]['address'];
cell2.innerHTML = result[i]['my_date'];
cell3.innerHTML = "<button id="+result[i]['id'] +
onclick='delete_function(this.id);'>DELETE </button>";

    s++;
}

}else{

    var first = 10 * clicked_id;
    var last = 10 * clicked_id + 9;

    if (result != null) {

        var s = 0;
        for (var i = first; i < last+1; i++) {

            var row = my_table.insertRow(s);
            row.className= "format";
            row.id = result[i]['id'];
            row.onmouseover = function() {ChangeBackgroundColor(this)};
```

```
        row.onmouseout = function() {RestoreBackgroundColor(this)};
        // row.onclick = function() {click_row(this.id)};
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);

        row.cells[0].id = result[i]['id'];
        row.cells[1].id = result[i]['id'];
        row.cells[0].onclick = function() {click_row(this.id)};
        row.cells[1].onclick = function() {click_row(this.id)};

        cell2.style.textAlign = "center";

        cell1.innerHTML = result[i]['address'];
        cell2.innerHTML = result[i]['my_date'];
        cell3.innerHTML = "<button id="+result[i]['id'] +"
onclick='delete_function(this.id);'>DELETE </button>";

        s++;

    }
}
}
```

**❖ Η διαγραφή πραγματοποιείται με ajax.**

```
function delete_function(clicked_id){
    var result = confirm("Are you sure you permanently want to delete your checkin?");
    if (result) {
        var my_click = clicked_id;
        $.ajax({
            type: "POST",
            url: 'delete.php',
            data: {location_id: my_click},
            success: function(data){
                alert("Checkin is deleted permanently!!");
            }
        });
        location.href = 'list_of_memories.php';
    }
    location.href = 'list_of_memories.php';
}
```

**❖ Τα δεδομένα επιστρέφονται σε μορφή json.**

```
<?php

require("connect.php");

if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

if(!isset($_SESSION['name'])){
    header("location: login.php");
}
```



```
$user_id = $_SESSION['id'];  
$dbh = conn();  
$query = "SELECT * FROM markers WHERE user_id = :user_id";  
$stmt=$dbh->prepare($query);  
$stmt->bindParam(':user_id', $user_id);  
$stmt->execute();  
  
$res = $stmt->fetchAll(PDO::FETCH_ASSOC);  
$json=json_encode($res);  
  
?>
```

❖ **Για την εύρεση της τοποθεσίας από τον υπολογιστή χρησιμοποιείται η Geocoder.**

```
function myMap(latitude, longitude) {  
    var address = "";  
    var geocoder = new google.maps.Geocoder();  
    var latitude = latitude;  
    var longitude = longitude;  
    var latLng = new google.maps.LatLng(latitude,longitude);  
    geocoder.geocode({  
        latLng: latLng  
    },  
    function(responses)  
    {  
        if (responses && responses.length > 0)  
        {  
            address = responses[0].formatted_address;  
        }  
    }  
}
```

```

Date.prototype.today = function () {
    return this.getFullYear()
    +"/"+(((this.getMonth()+1) < 10)?"0":"" ) + (this.getMonth()+1)
    +"/"+ ((this.getDate() < 10)?"0":"" ) + this.getDate();
}

Date.prototype.timeNow = function () {
    return ((this.getHours() < 10)?"0":"" ) + this.getHours() +":"+
((this.getMinutes() < 10)?"0":"" ) + this.getMinutes() +":"+ ((this.getSeconds() < 10)?"0":"" ) +
this.getSeconds();
}

var myDate = new Date().today() + " " + new Date().timeNow();
var mapOptions = {
    center: new google.maps.LatLng(latitude, longitude),
    zoom: 15
}
var map = new google.maps.Map(document.getElementById("map"),
mapOptions);

var marker = new google.maps.Marker({
    position: latLng,
    map: map
});
$.ajax({
    type: "POST",
    url: 'store_location.php',
    data: {address: address, lat: latitude, lng: longitude, my_date: myDate},
    success: function(data){
        alert('Location saved successfully');
    }
});
}
}
);
}

```

### ❖ Η αποθήκευση εικόνας και βίντεο πραγματοποιείται στο Firebase.

```

<script src="https://www.gstatic.com/firebasejs/4.3.1/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AlzaSyDYSIrv7CflmQOO1_yp0QINeYzgFAKAOJM",
    authDomain: "august-will-173913.firebaseio.com",
    databaseURL: "https://august-will-173913.firebaseio.com",
    projectId: "august-will-173913",
    storageBucket: "august-will-173913.appspot.com",
    messagingSenderId: "356782305500"
  };
  firebase.initializeApp(config);
  //Get elements
  var uploaderImage = document.getElementById('uploaderImage');
  var fileButtonImage = document.getElementById('fileButtonImage');
  var uploaderVideo = document.getElementById('uploaderVideo');
  var fileButtonVideo = document.getElementById('fileButtonVideo');
  //Listen for file selection
  fileButtonImage.addEventListener('change', function(e){

    var file = e.target.files[0];
    if (file == ""){
      alert('Please select file');
    }else{
      fileName = file.name;
      var extension = fileName.substring(fileName.indexOf('.') + 1).toLowerCase();
      if (extension == "jpg" || extension == "png" || extension == "jpeg" || extension == "gif") {

        var storageRef = firebase.storage().ref('Photos/'+ <?php echo $_SESSION['id']; ?> +
        '_' + <?php

```

```
        var task = storageRef.put(file);
        if (task) {
            document.getElementById("fileButtonImage").disabled = true;
            alert('Image saved successfully' + storageRef);
        }
    }else{
        alert('Please upload file having extensions .jpeg/.jpg/.png/.gif only.');
```

```
    }}
});
fileButtonVideo.addEventListener('change', function(e){
    var file = e.target.files[0];
    if (file == ""){
        alert('Please select file');
```

```
    }else{
        fileName = file.name;
        var extension = fileName.substr(fileName.indexOf('.') + 1).toLowerCase();
        if (extension == "mp4" || extension == "3gpp" || extension == "3gp") {
            var storageRef = firebase.storage().ref('Videos/'+ <?php echo $_SESSION['id']; ?> +
'_ ' + <?php
            echo $save_location_id; ?>+ '.mp4');
```

```
            var task = storageRef.put(file);
            if (task) {
                document.getElementById("fileButtonVideo").disabled = true;
                alert('Video saved successfully'+ storageRef);
            }
        }else{
            alert('Please upload file having extensions .mp4/.3gpp/.3gp only.');
```

```
        }}
    });
</script>
```

**❖ Εμφάνιση τοποθεσιών στον χάρτη.**

```
<script>
function myMap(latitude, longitude) {
    var mapOptions = {
        center: new google.maps.LatLng(latitude, longitude),
        zoom: 15
    }
    var map = new google.maps.Map(document.getElementById("map"), mapOptions);
}

$( document ).ready(function() {
    // var current page = "<?php //echo $res; ?>";
    var mapOptions = {
        center: new google.maps.LatLng(0.0, 0.0),
        zoom: 2
    }
    var map = new google.maps.Map(document.getElementById("map"), mapOptions);
    $.ajax({
        url : 'get_values.php',
        type : 'POST',
        dataType : 'json',
        success : function (result) {
            var bounds = new google.maps.LatLngBounds();
            var i;
            if (result != null){
                for (i = 0; i < result.length; i++) {
                    var longitude = parseFloat(result[i]['lng']);
                    var latitude = parseFloat(result[i]['lat']);
                    var location_id = result[i]['id'];
                    var myLatLng = new google.maps.LatLng(latitude, longitude);
```

```
var marker = new google.maps.Marker({
    position: {lat: latitude, lng: longitude},
    map: map,
    title: result[i]['address'] + '\n' + result[i]['my_date']
});
marker.set("id", result[i]['id']);
google.maps.event.addListener(marker, 'click', (function(marker, i) {
return function() {
    $.ajax({
        type: "POST",
        url: 'test1.php',
        data: {location_id: marker.get("id")},
        success: function(data){
            // alert(data);
        }
    });
    location.href = 'focus_on_each_memory.php';
})(marker, i));
    bounds.extend(myLatLng);
} map.fitBounds(bounds);
},
error : function () {
    alert("error");
}
}); });
var x = document.getElementById("demo");
function showPosition(position) {
    myMap(position.coords.latitude, position.coords.longitude);
}
</script>
```

## 5 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Οι μελλοντικές επεκτάσεις που θα μπορούσαν να γίνουν στο σύστημά μας είναι πολλές. Παρακάτω θα αναφερθούν ενδεικτικά κάποιες από τις επεκτάσεις της εφαρμογής:

- Σύνδεση της εφαρμογής με social media, όπως είναι το facebook και το Instagram.
- Αποστολή μηνυμάτων μεταξύ χρηστών.
- Αποθήκευση πολλαπλών δεδομένων στο κάθε checkin του χρήστη.
- Δυνατότητα αλλαγής των αποθηκευμένων δεδομένων του χρήστη.

## Βιβλιογραφία

- <https://firebase.google.com/docs/reference/js/firebase.storage>
- <https://developers.google.com/maps/>
- [https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)
- <https://stackoverflow.com/questions/409999/getting-the-location-from-an-ip-address>
- <https://files.000webhost.com/>
- [https://www.w3schools.com/js/js\\_json\\_arrays.asp](https://www.w3schools.com/js/js_json_arrays.asp)
- <https://developer.android.com/reference/android/widget/ArrayAdapter.html>
- <https://developer.android.com/reference/android/os/Handler.html>
- <https://stackoverflow.com/questions/15609306/convert-string-to-json-array>
- <https://developer.android.com/reference/java/net/URLConnection.html>
- <https://developer.android.com/guide/topics/media/mediarecorder.html>
- <https://stackoverflow.com/questions/20210565/android-location-manager-get-gps-location-if-no-gps-then-get-to-network-provid>