Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2020-2021

Διδάσκοντες: Ν. Παπασπύρου, Β. Βεσκούκης

Θεματικό πεδίο

Το θεματικό πεδίο της εργασίας είναι το πρόβλημα της διαχείρισης της φόρτισης ηλεκτρικών οχημάτων, η οποία πραγματοποιείται σε σταθμούς που είναι εγκατεστημένοι σε δημόσιους ή ιδιωτικούς χώρους, όπως οι δημοτικοί χώροι στάθμευσης στο δρόμο, οι οργανωμένοι χώροι στάθμευσης (parking), τα πρατήρια καυσίμων, τα ιδιωτικά οικιακά parking, κ.ά. Η επιδιωκόμενη διάδοση των ηλεκτρικών οχημάτων και η (προς το παρόν;) μεγάλη διάρκεια της φόρτισης ενός ηλεκτρικού οχήματος σε σύγκριση με τη διάρκεια ανεφοδιασμού σε υγρά καύσιμα, καθώς και οι ιδιαίτερες απαιτήσεις των φορτιστών σε ηλεκτρική ισχύ, καθιστούν αναγκαίο ένα σχεδιασμό της διαδικασίας της φόρτισης, η οποία δεν μπορεί να είναι πανομοιότυπη με αυτή του ανεφοδιασμού με υγρά καύσιμα.

Παράλληλα, η απελευθέρωση της αγοράς ηλεκτρικής ενέργειας επιτρέπει σε πολλούς παρόχους να διαθέτουν ηλεκτρική ενέργεια για φόρτιση οχημάτων σε ανταγωνιστικές τιμές. Οι πάροχοι αυτοί δυνητικά είναι πολύ περισσότεροι από τους ελάχιστους παρόχους υγρών καυσίμων. Η δυνατότητα αυτή επιτρέπει ο ιδιοκτήτης (ακριβέστερα: ο διαχειριστής, operator) ενός σταθμού φόρτισης να μην συνδέεται υποχρεωτικά με έναν μόνο πάροχο ενέργειας, όπως συμβαίνει στα υγρά καύσιμα. Επίσης, απαιτείται οποιοδήποτε όχημα να μπορεί να φορτιστεί σε οποιοδήποτε σταθμό. Αναγκαία προϋπόθεση για να γίνει αυτό, είναι η προτυποποίηση των πρωτοκόλλων φόρτισης και των ηλεκτρικών συνδέσεων, σχετικά με την οποία μπορείτε να βρείτε κάποιες αναφορές στο χώρο wiki της εργασίας.

Τέλος, μια σημαντική διαφοροποίηση της φόρτισης ηλεκτρικών οχημάτων από τον ανεφοδιασμό υγρών καυσίμων είναι ότι υπηρεσίες φόρτισης ξεκίνησαν να σχεδιάζονται μόνο πρόσφατα, οπότε είναι εξ αρχής εφικτή η μεταφορά δεδομένων που σχετίζονται με τη φόρτιση προς ένα ή περισσότερα πληροφοριακά συστήματα. Σε αυτά έχουν πρόσβαση οι διάφοροι εμπλεκόμενοι (stakeholders), καθένας για τους δικούς του σκοπούς, οι οποίοι μπορούν να εκτείνονται από την έκδοση λογαριασμών (billing) μέχρι τη χάραξη πολιτικής.

Διατύπωση ζητουμένου

Ενα τέτοιο πληροφοριακό σύστημα, στο οποίο υποθέτουμε ότι θα έχουν πρόσβαση διάφοροι εμπλεκόμενοι (stakeholders), είναι το αντικείμενο της εξαμηνιαίας εργασίας του μαθήματος. Η έκταση ενός πραγματικού τέτοιου συστήματος είναι προφανώς απαγορευτικά μεγάλη, οπότε θα ακολουθήσουμε μια προσέγγιση κατά την οποία θα αναγνωρίσουμε τους εμπλεκόμενους και τις λειτουργίες με μια σχετική ευρύτητα, ενώ θα ασχοληθούμε βαθύτερα με ένα μικρό υποσύνολο των λειτουργιών. Η ενασχόλησή μας θα περιλαμβάνει την αναγνώριση και προδιαγραφή των

απαιτήσεων, την αρχιτεκτονική και λεπτομερή σχεδίαση, καθώς και την υλοποίηση επιλεγμένων λειτουργιών, κάποιων κοινών και κάποιων διαφορετικών για κάθε ομάδα. Η προδιαγραφή των απαιτήσεων και η σχεδίαση θα γίνουν με χρήση κατάλληλων προτύπων και εργαλείων (πρότυπα IEEE, UML, git, εργαλεία αυτόματου deployment και testing).

Η αναγνώριση των εμπλεκομένων (stakeholders) και των λειτουργιών που θα περιλαμβάνει το σύστημα θα γίνει με συνεργατικές συζητήσεις (workshops) στο μάθημα. Με την ολοκλήρωση αυτών, κατάλογος με τους εμπλεκόμενους και τις λειτουργίες θα είναι διαθέσιμος στο χώρο Teams και στο wiki του μαθήματος. Από τον κατάλογο αυτό κάποιες λειτουργίες θα χαρακτηριστούν κοινές και θα υλοποιηθούν από όλες τις ομάδες, ενώ κάποιες άλλες θα μπορούν να επιλέγονται από κάθε ομάδα σύμφωνα με τον αριθμό των μελών της.

Θα σας διατεθούν κάποια σύνολα ανοικτών δεδομένων τα οποία σχετίζονται με φόρτιση ηλεκτρικών οχημάτων και τα οποία γίνονται διαθέσιμα από πηγές που θα αναφέρονται κατά περίπτωση. Τα σύνολα αυτά αφορούν:

α) Γεγονότα φόρτισης ηλεκτρικών οχημάτων

β) Θέσεις φόρτισης ηλεκτρικών οχημάτων

γ) Τεχνικά χαρακτηριστικά ηλεκτρικών οχημάτων

Τα δεδομένα αυτά, σε συνδυασμό με τις λειτουργίες που θα υλοποιήσετε, θα καθοδηγήσουν τη σχεδίαση των διαφορετικών όψεων του πληροφοριακού συστήματος που κάθε ομάδα θα προδιαγράψει, σχεδιάσει και κατασκευάσει. Τα τμήματα του πληροφοριακού συστήματος έχουν ως εξής:

1. Ένα υποσύστημα back-end, το οποίο θα υποστηρίζει δυνατότητες διαχείρισης χρηστών (εγγραφή, σύνδεση, αποσύνδεση), προκειμένου αυτοί να αποκτούν πρόσβαση σε δεδομένα μέσω ενός REST API.
2. Ένα υποσύστημα back-end, υπεύθυνο για τη διαχείριση των δεδομένων και την πρόσβαση σε αυτά (εγγραφή, ανάγνωση) μέσω ενός REST API.
3. Μία εφαρμογή CLI (Command Line Interface) για τις λειτουργίες προσπέλασης δεδομένων και διαχείρισης χρηστών. Η εφαρμογή θα λειτουργεί ως client του REST API που παρέχεται από το back-end υποσύστημα, προσφέροντας στο χρήστη της τη δυνατότητα να εκτελεί λειτουργίες δημιουργίας χρήστη, σύνδεσης χρήση, ανάγνωσης από και εγγραφής στα σύνολα δεδομένων.
4. Μία δικτυακή εφαρμογή (Web Application), η οποία θα προσφέρει στο χρήστη δυνατότητες παρουσίασης των δεδομένων (διαγράμματα, οπτικοποίηση σε χάρτη). Η εφαρμογή αυτή θα λειτουργεί σε περιβάλλον Web browser και θα αποτελεί το front-end του συστήματος (δεύτερος client του REST API). Τα δεδομένα που θα παρουσιάζονται θα καθορίζονται από τις λειτουργίες που κάθε ομάδα θα επιλέξει. Ο αριθμός των λειτουργιών σχετίζεται με τον αριθμό των μελών κάθε ομάδας όπως αναφέρεται σε πίνακα παρακάτω.

Θα σας διατεθούν οι προδιαγραφές των REST APIs για τις κοινές λειτουργίες και χρήσιμα σύνολα δεδομένων. Οι αντίστοιχες προδιαγραφές για τις επιλεγόμενες ανά ομάδα λειτουργίες αποτελούν παραδοτέα της εργασίας. Η λίστα και ο χαρακτηρισμός των λειτουργιών (κοινές, επιλεγόμενες) που θα περιλαμβάνονται στις περιπτώσεις χρήσης που θα υλοποιηθούν θα ανακοινωθούν στο Teams και στο Moodle του μαθήματος.

Ομάδες εργασίας

Η εργασία θα γίνει σε **ομάδες των 3, 4, 5 ή 6 ατόμων**, οι οποίες θα υλοποιήσουν τον πλήρη κύκλο ανάπτυξης του συστήματος (ανάλυση απαιτήσεων, σύνταξη προδιαγραφών, σχεδιασμός και αρχιτεκτονική, υλοποίηση και έλεγχοι αποδοχής, εγκατάσταση και λειτουργία).

Για τη συνεργατική διαχείριση των εκδόσεων της τεκμηρίωσης και του πηγαίου κώδικα, είναι υποχρεωτική η χρήση του Github, όπως αναφέρεται και ακολούθως. Η συγκρότηση των ομάδων και οι λογαριασμοί χρηστών των μελών τους στο Github θα πρέπει να μας γνωστοποιηθούν το αργότερο μέχρι τις **15.11.2019**. Θα πρέπει, επίσης, **με τον ιδρυματικό λογαριασμό χρήστη** να εγγραφείτε στο συνεργατικό σύστημα ηλεκτρονικής μάθησης Moodle που χρησιμοποιείται στο μάθημα (courses.pclab.ece.ntua.gr), ώστε να έχετε άμεση πρόσβαση σε επικοινωνίες, ανακοινώσεις, υλικό, κ.ά..

Ελάχιστες κοινές τεχνικές προδιαγραφές

Το σύστημα που θα αναπτύξετε θα πρέπει να υποστηρίζει τα εξής χαρακτηριστικά (ελάχιστες κοινές προδιαγραφές):

1. Το σύστημα θα αποτελείται από back-end και front-end υποσυστήματα που θα υλοποιούν έναν αριθμό περιπτώσεων χρήσης, στις οποίες θα περιλαμβάνονται λειτουργίες από τη λίστα που έχει ανακοινωθεί στο **SE2020-RequirementsStakeholdersFINAL_2.xlsx**. Ο αριθμός των περιπτώσεων χρήσης εξαρτάται από το πλήθος των μελών της ομάδας, όπως αναφέρεται στη συνέχεια.
2. Κάθε υποσύστημα back-end θα παρέχει κατάλληλο REST API για τη διασύνδεσή του με τις υπόλοιπες εφαρμογές, το οποίο προαιρετικά μπορεί να είναι συμβατό με το πρότυπο OpenAPI 3.0. Τα endpoints που θα αναπτύξετε θα προδιαγραφούν λεπτομερώς από τους διδάσκοντες κατά τη διάρκεια υλοποίησης της εργασίας και θα είναι κοινά για όλες τις ομάδες.
3. Η γλώσσα των διεπαφών χρήστη στην εφαρμογή CLI θα είναι η αγγλική. Η γλώσσα των διεπαφών χρήστη για τις άλλες εφαρμογές, όπου απαιτείται, θα είναι η ελληνική ή η αγγλική.
4. Κάθε ομάδα θα πρέπει να κάνει χρήση ενός εργαλείου αυτοματισμού του «χτισίματος» του λογισμικού (build automation), το οποίο στην περίπτωση της Java θα είναι το Gradle, ενώ στην περίπτωση της Javascript θα είναι της επιλογής σας. Στο πλαίσιο του μαθήματος θα γίνει φροντιστήριο / εργαστήριο για το Gradle.
5. Κάθε ομάδα θα συντάξει σενάρια ελέγχου και θα ενσωματώσει την εκτέλεση των αντίστοιχων δοκιμών με αυτόματο τρόπο για τις λειτουργίες του back-end υποσυστήματος. Για το σκοπό αυτό, στην περίπτωση της Java θα χρησιμοποιηθεί το εργαλείο Spock, ενώ στην περίπτωση της Javascript, το εργαλείο θα είναι της επιλογής σας. Στο πλαίσιο του μαθήματος θα γίνει φροντιστήριο / εργαστήριο για το Spock.
6. Η πρόσβαση στο REST API θα πρέπει να γίνεται με λογαριασμούς χρηστών. Κάθε λογαριασμός χρήστη θα περιέχει εκτός από τα συνηθισμένα δεδομένα (username, password, email κ.λπ.) και όριο χρήσης (quota) ώστε να περιορίζεται η κατανάλωση πόρων του API.
7. Θα πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις χρηστικές ή προγραμματιστικές διεπαφές μέσω self-signed certificate.

Οι απαιτήσεις αυτές εξειδικεύονται ανάλογα με το πλήθος των μελών της ομάδας, όπως φαίνεται στον ακόλουθο πίνακα:

ΑΠΑΙΤΗΣΗ	**Ομάδες**			
	3 ατόμων	**4 ατόμων**	**5 ατόμων**	**6 ατόμων**
1. Εισαγωγή δεδομένων	Back-end, CLI			
2. Διαχείριση χρηστών	Back-end, CLI, Web-based front-end			
3. Πρόσβαση σε δεδομένα	Back-end, CLI			
4. Παρουσίαση δεδομένων για τις περιπτώσεις χρήσης κάθε ομάδας	Web-based front-end			
5. Αριθμός περιπτώσεων χρήσης (N)	2	3	4	5

Οι λειτουργίες που θα περιλαμβάνονται στις περιπτώσεις χρήσης θα επιλεγούν από τον πίνακα που δίνεται στο αρχείο **SE2020-RequirementsStakeholdersFINAL_2.xlsx**. Οι λειτουργίες που σημειώνονται με κίτρινο πρέπει να επιλεγούν υποχρεωτικά από όλες τις ομάδες για να ενταχθούν σε 2-3 περιπτώσεις χρήσης. Οι ομάδες που θα υλοποιήσουν περισσότερες περιπτώσεις χρήσης, μπορούν να επιλέξουν για να εντάξουν σε αυτές, οποιεσδήποτε από τις υπόλοιπες λειτουργίες του πίνακα **SE2020-RequirementsStakeholdersFINAL_2.xlsx**.

Τεχνικοί περιορισμοί

Επιπλέον όσων αναφέρθηκαν ισχύουν και ορισμένοι άλλοι τεχνικοί περιορισμοί για την εκπόνηση της εργασίας. Το σύνολο των περιορισμών συνοψίζεται ως ακολούθως:

- Περιβάλλον διαχείρισης εκδόσεων: **github.** Δεν θα χρησιμοποιηθούν άλλα παρεμφερή περιβάλλοντα (πχ Bitbucket, GitLab). Το repository θα πρέπει να είναι ιδιωτικό (private) και θα έχουν πρόσβαση μόνο τα μέλη της ομάδας και οι διδάσκοντες.
- Εργαλείο παραγωγής διαγραμμάτων UML ένα από τα ακόλουθα: (α) Visual Paradigm 16, community edition (β) Visual Paradigm 16 web
- Διαχείριση δεδομένων: ένα εκ των **MySQL**, **MariaDB, PostgreSQL**, **Mongo**, **Elastic Search**

Τεχνικές επιλογές οι οποίες δεν περιλαμβάνονται στους παραπάνω περιορισμούς (π.χ. βιβλιοθήκες JS για διαγράμματα / γραφήματα στο web), είναι ελεύθερες.

Παραδοτέα

Δομή φακέλων στο Git repository της ομάδας

Η δομή των φακέλων στο git repository της κάθε ομάδας θα πρέπει να ακολουθεί την εξής δομή:

- Φάκελος **back-end**: περιέχει τον κώδικα της εφαρμογής back-end.
- Φάκελος **cli-client**: περιέχει τον κώδικα της εφαρμογής CLI.
- Φάκελος **documentation**: περιέχει την τεκμηρίωση της εργασίας (συνολικά).
- Φάκελος **front-end**: περιέχει τον κώδικα της εφαρμογής front-end.

Σε περίπτωση που το κρίνετε απαραίτητο, μπορείτε να προσθέσετε νέους φακέλους στην παραπάνω δομή (για παράδειγμα, λόγω δημιουργίας κάποιου component/module που (επανα)χρησιμοποιείται σε δύο ή περισσότερες εφαρμογές).

Λίστα παραδοτέων

Η λίστα των παραδοτέων του μαθήματος φαίνεται στον παρακάτω πίνακα. Τα παραδοτέα θα πρέπει να καταχωριστούν υποχρεωτικά στον αντίστοιχο φάκελο του github repository κάθε ομάδας, με ημ/νία καταχώρισης (commit date) προγενέστερη της προθεσμίας.

<table>
<tr><th rowspan="2">Παραδοτέο</th><th rowspan="2">Φάκελος github</th><th colspan="4">Ομάδα</th></tr>
<tr><th>3 ατόμων</th><th>4 ατόμων</th><th>5 ατόμων</th><th>6 ατόμων</th></tr>
<tr><td colspan="6">Documentation – Diagrams</td></tr>
<tr><td>Εγγραφο SRS - Software Requirements Specification</td><td rowspan="8">documentation</td><td>2 Stakeholders</td><td>2 Stakeholders</td><td>3 Stakeholders</td><td>4 Stakeholders</td></tr>
<tr><td>Εγγραφο SRS - Software Requirements Specification</td><td>2 Use Cases</td><td>3 Use Cases</td><td>4 Use Cases</td><td>5 Use Cases</td></tr>
<tr><td>Διαγράμματα UML Activity</td><td colspan="4">αντίστοιχα με τα Use Cases</td></tr>
<tr><td>Διαγράμματα UML Sequence</td><td colspan="4">αντίστοιχα με τα Use Cases</td></tr>
<tr><td>Διαγράμματα UML Deployment</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Διαγράμματα UML Component</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Διαγράμματα ER</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Διαγράμματα UML Class</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td colspan="6">Source code – implementation</td></tr>
<tr><td>Πηγαίος κώδικας εφαρμογής για εισαγωγή, διαχείριση και πρόσβαση σε δεδομένα (backend)</td><td>back-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Database dump (sql ή json)</td><td>back-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>RESTful API</td><td>back-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Command line interface (CLI)</td><td>cli-client</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Εκτελέσιμη μορφή (build & deploy your code from source)</td><td>back-end, cli-client, front-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Back-end unit tests</td><td>back-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>CLI unit tests</td><td>cli-client</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Back-end functional tests</td><td>back-end</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>CLI functional tests</td><td>cli-client</td><td colspan="4">ΝΑΙ</td></tr>
<tr><td>Front-end tests</td><td>front-end</td><td colspan="2"></td><td colspan="2">ΠΡΟΑΙΡΕΤΙΚΑ</td></tr>
</table>

Όλα τα σενάρια ελέγχων (unit and functional tests) θα πρέπει να εκτελούνται αυτοματοποιημένα από το build εργαλείο σας.

Μορφότυποι Παραδοτέων

Η μορφή των παραδοτέων φαίνεται στον παρακάτω πίνακα.

Παραδοτέο	**Μορφότυπος**	**Παρατηρήσεις**
Documentation – Diagrams		
Έγγραφα StRS & SRS	docx, odt ή άλλη ανοικτή **επεξεργάσιμη** μορφή, σύμφωνα με το πρότυπο που διατέθηκε	Δεν θα γίνουν δεκτά pdf, παρά μόνο αν συνοδεύονται από αρχεία σε επεξεργάσιμη μορφή
Διαγράμματα UML	Αρχείο Visual Paradigm (vpp) ή Παραπομπή στα αντίστοιχα online διαγράμματα Visual Paradigm	Θα πρέπει να παραδοθεί ***ένα*** αρχείο visual paradigm (vpp) με όλα τα διαγράμματα. Θα πρέπει να υπάρχει κατάλληλη ονομασία των διαγραμμάτων ώστε να είναι εύκολα ανιχνεύσιμη η ένταξή τους στα κείμενα, όπου γίνεται. Επίσης, στα διαγράμματα είναι επιθυμητές αναφορές (παραπομπές) στα αντίστοιχα συστατικά στοιχεία πηγαίου κώδικα, όπου αυτό έχει νόημα, καθώς και συνδέσεις μεταξύ τους. Δεν θα γίνουν δεκτά διαγράμματα σε μορφή εικόνων ή από σχεδιαστικά λογισμικά γενικής χρήσης.
Source code – implementation		
Πηγαίος κώδικας για το σύνολο των λειτουργιών που υλοποιήσατε	Το σύνολο των πηγαίων αρχείων σύμφωνα με το περιβάλλον, εργαλεία κλπ που θα χρησιμοποιήσετε.	Θα πρέπει να περιλάβετε στο αρχείο README σε κάθε επιμέρους εφαρμογής, παρέχοντας μια συνοπτική περιγραφή των συστατικών της εφαρμογής και των βημάτων που απαιτούνται για το στήσιμο του περιβάλλοντος ανάπτυξης (IDE, βιβλιοθήκες, κλπ).

Προθεσμίες

Το πρώτο σκέλος των παραδοτέων (τα έγγραφα και διαγράμματα τεκμηρίωσης) θα υποβληθεί δύο φορές, σε δύο εκδόσεις, σύμφωνα με το χρονοδιάγραμμα που ακολουθεί. Το δεύτερο σκέλος των παραδοτέων (πηγαίος κώδικας, υλοποίηση) θα καταχωρείται στο github καθώς εξελίσσεται η ανάπτυξη του λογισμικού, με καταληκτική ημερομηνία για το τελευταίο commit που δίνεται ακολούθως. Για όλες τις προθεσμίες, λαμβάνεται υπόψη η ημερομηνία καταχώρησης (commit) στο github repository της ομάδας σας.

1η έκδοση τεκμηρίωσης

Η πρώτη έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documentation & diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυκτα της Κυριακής 13 Δεκεμβρίου 2020** (με βάση, πάντα, την ημερομηνία καταχώρισης των αρχείων στο git repository της ομάδας σας).

2η έκδοση τεκμηρίωσης (τελική)

Η τελική έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documents & diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυκτα της Κυριακής 17 Ιανουαρίου 2021.** Μπορείτε να υποβάλλετε τροποποιημένες εκδοχές των εγγράφων και των διαγραμμάτων, εφόσον έχετε κάνει τροποποιήσεις σε σχέση με την πρώτη τους έκδοση, καταγράφοντας σε ένα ξεχωριστό αρχείο **documentation/updates.md** μια συνοπτική περιγραφή τους και **χωρίς να διαγράψετε τις αρχικές εκδόσεις των εγγράφων ή διαγραμμάτων**.

Πηγαίος κώδικας

Η τελική εκδοχή του πηγαίου κώδικα της εργασίας πρέπει να έχει καταχωρηθεί (commit) στο github έως τα μεσάνυκτα της προηγούμενης από την ημέρα που θα ξεκινήσουν οι προφορικές εξετάσεις, **για όλες τις ομάδες, ανεξάρτητα από την ημέρα και ώρα εξέτασης κάθε ομάδας**. Η σταδιακή ανάπτυξή του πρέπει να μπορεί να ανιχνεύεται στα commits που θα έχουν γίνει στο github καθ' όλη τη διάρκεια του εξαμήνου.

Εξέταση εργασίας

Το ακριβές πρόγραμμα εξέτασης των ομάδων θα καθοριστεί εντός του Ιανουαρίου 2021. Η ένταξη των ομάδων στα διαθέσιμα χρονοπαράθυρα (διάρκειας 30 λεπτών) θα ανακοινωθεί.