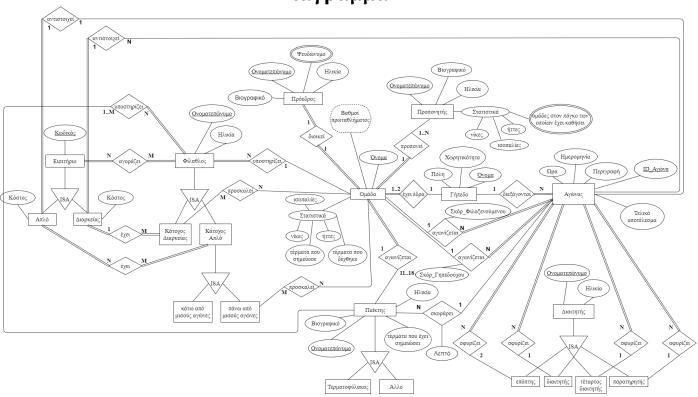


Τμήμα Μηχανικών Η/Υ & Πληροφορικής Μάθημα: Βάσεις Δεδομένων Εργασία Ακ. Έτος 2016-2017

Ονοματεπώνυμο	AM
Παναγιώτης Σταυρινάκης	6217
Εμμανουήλ Κατεφίδης	6077

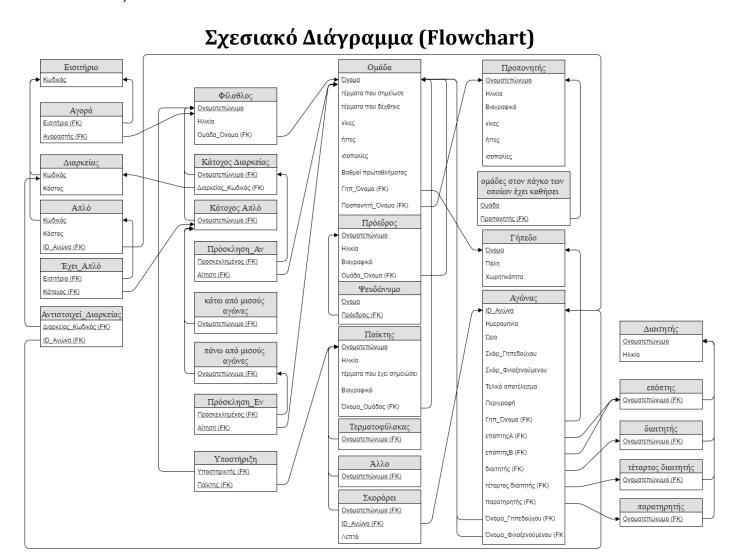
Το ER διάγραμμα και Σχεσιακό βρίσκονται επίσης σε αρχείο png στον φάκελο για ευκολότερη ανάγνωση.

Διάγραμμα ΕR



- Με διπλή γραμμή δηλώνουμε υποχρεωτική συσχέτιση.
- Κάθε ομάδα έχει υποχρεωτικά έναν ως πολλούς προπονητές και κάθε προπονητής προπονεί μια το πολύ ομάδα.
- Κάθε ομάδα πρέπει να έχει έναν πρόεδρο και κάθε πρόεδρος πρέπει να διοικεί μια ομάδα ακριβώς.
- Κάθε ομάδα πρέπει να έχει μια έδρα, κάθε γήπεδο μπορεί να φιλοξενεί ως το πολύ δυο ομάδες.
- Αγώνες πρέπει να διεξάγονται σε γήπεδο, σε ένα γήπεδο μπορεί να μην διεξάγονται αγώνες.
- Δύο ομάδες ακριβώς (Φιλοξενούμενος-Γηπεδούχος) αγωνίζονται σε έναν αγώνα κάθε φορά.

- Οι διαιτητές χωρίζονται σε υποκατηγορίες, οι αγώνες πρέπει να έχουν διαιτητές.
- Οι ομάδες πρέπει να έχουν παίκτες (11..18), οι παίκτες χωρίζονται σε υποκατηγορίες, πολλοί παίκτες μπορούν να σκοράρουν σε κάθε αγώνα οποιοδήποτε λεπτό.
- Ο φίλαθλος χωρίζεται σε υποκατηγορίες, πρέπει να αγοράζει εισιτήρια και τα εισιτήρια να αγοράζονται από φιλάθλους, πρέπει να υποστηρίζει μια το πολύ ομάδα και μπορεί να υποστηρίζει όσους παίκτες θέλει.
- Το εισιτήριο χωρίζεται σε υποκατηγορίες.
- Ο κάτοχος διαρκείας πρέπει να έχει εισιτήριο διαρκείας και ο κάτοχος απλού να έχει απλό.
- Ένα απλό εισιτήριο αντιστοιχεί σε έναν αγώνα, ένα εισιτήριο διαρκείας αντιστοιχεί σε πολλούς.



Ερώτημα 3(τα ζητούμενα ερωτήματα σε SQL ή τα αντίστοιχα stored procedures) Τα SELECT και STORED PROCEDURES βρίσκονται επίσης σε αρχείο txt στον φάκελο για ευκολότερη ανάγνωση.

1. SELECT Proedros.Biografiko AS Biografiko_Proedrou, Proponitis.Biografiko AS Biografiko_Proponitis FROM Proponitis

LEFT JOIN Omada ON Omada.Propon_Onoma = Proponitis.Onomateponumo LEFT JOIN Proedros ON Proedros.Omada_Onoma = Omada.Onoma ORDER BY Omada.nikes DESC LIMIT 0,1;

```
2. DROP PROCEDURE IF EXISTS ER2;
DELIMITER $
CREATE PROCEDURE ER2()
BEGIN
DECLARE Onoma Paikti VARCHAR(50);
DECLARE termata1 TINYINT(3);
DECLARE termata2 TINYINT(3);
DECLARE Diafora TINYINT(3);
SELECT Onomateponumo, termata simeiose
INTO Onoma_Paikti, termata1
FROM Paiktis
ORDER BY termata_simeiose DESC LIMIT 0,1;
SELECT termata simeiose
INTO termata2
FROM Paiktis
ORDER BY termata simeiose DESC LIMIT 1,1;
SET Diafora=termata1-termata2;
SELECT Onoma_Paikti, Diafora;
END
DELIMITER:
CALL ER2();
3. DROP PROCEDURE IF EXISTS ER3;
DELIMITER $
CREATE PROCEDURE ER3()
BEGIN
DECLARE Onom VARCHAR(25);
DECLARE finishedFlag INT:
DECLARE OmCursor CURSOR FOR
SELECT Onoma FROM Omada;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET finishedFlag=1;
OPEN OmCursor;
SET finishedFlag=0;
FETCH OmCursor INTO Onom;
WHILE (finishedFlag=0) DO
SELECT Omada. Onoma AS Onoma Omadas, Paiktis. Onomateponumo AS Onoma Paikti,
GROUP_CONCAT(ypostiriksi.Ypostiriktis SEPARATOR ', ') AS Ypostiriktis
FROM Omada
LEFT JOIN Paiktis ON Paiktis.Onoma_Omadas=Omada.Onoma
LEFT JOIN ypostiriksi ON ypostiriksi.Paiktis=Paiktis.Onomateponumo
WHERE Omada.Onoma=Onom
GROUP BY Paiktis.Onomateponumo
ORDER BY Paiktis.termata_simeiose DESC LIMIT 0,3;
FETCH OmCursor INTO Onom;
END WHILE;
CLOSE OmCursor:
END
```

```
DELIMITER;
CALL ER3();
4.DROP PROCEDURE IF EXISTS ER4;
DELIMITER $
CREATE PROCEDURE ER4()
BEGIN
DECLARE Onom VARCHAR(25);
DECLARE finishedFlag INT;
DECLARE OmCursor CURSOR FOR
SELECT Onoma FROM Omada:
DECLARE CONTINUE HANDLER FOR NOT FOUND SET finishedFlag=1;
OPEN OmCursor;
SET finishedFlag=0;
FETCH OmCursor INTO Onom:
WHILE (finishedFlag=0) DO
SELECT Omada.Onoma AS Onoma_Omadas, COUNT(Katoxos_diarkeias.Diarkeias_Kodikos) AS
Arithmos Eisitiriwn Diarkeias.
GROUP_CONCAT(Katoxos_diarkeias.Onomateponumo SEPARATOR ', ') AS
Katoxos_Eisitiriou_Diarkeias
FROM Omada
LEFT JOIN Filathlos ON Filathlos.Omada_Onoma=Omada.Onoma
LEFT IOIN Katoxos diarkeias ON Katoxos diarkeias. Onomateponumo = Filathlos. Onomateponumo
WHERE Omada.Onoma=Onom
GROUP BY Omada.Onoma;
FETCH OmCursor INTO Onom;
END WHILE;
CLOSE OmCursor:
END
DELIMITER:
CALL ER4();
5. DROP PROCEDURE IF EXISTS ER5;
DELIMITER $
CREATE PROCEDURE ER5()
BEGIN
DECLARE Onoma_Filathlou VARCHAR(50);
DECLARE Arithmos INT;
SELECT Katoxos_diarkeias.Onomateponumo, COUNT(Agonas.ID_Agona) AS
Arithmos_Parakolouthisewn
INTO Onoma Filathlou, Arithmos
FROM Katoxos diarkeias
LEFT JOIN Filathlos ON Katoxos_diarkeias.Onomateponumo=Filathlos.Onomateponumo
LEFT JOIN Omada ON Filathlos.Omada_Onoma=Omada.Onoma
```

LEFT JOIN Agonas ON (Omada.Onoma=Agonas.Onoma_Gipedouxou OR Omada.Onoma=Agonas.Onoma_Filoksenoumenou) LEFT JOIN antistoixei_diarkeias ON Agonas.ID_Agona=antistoixei_diarkeias.ID_Agona LEFT JOIN Diarkeias ON Diarkeias.Kodikos=antistoixei diarkeias.Diarkeias Kodikos LEFT JOIN Eisitirio ON Eisitirio.Kodikos = Diarkeias.Kodikos LEFT JOIN Agora ON Eisitirio.Kodikos = Agora.Eisitirio WHERE Agora. Agorastis = Katoxos_diarkeias. Onomateponumo GROUP BY Katoxos diarkeias. Onomateponumo ORDER BY Arithmos Parakolouthisewn DESC LIMIT 0,1; SELECT Katoxos_diarkeias.Onomateponumo, prosklisi_an.Aitisi FROM Katoxos_diarkeias LEFT JOIN prosklisi_an ON Katoxos_diarkeias.Onomateponumo=prosklisi_an.Proskeklimenos WHERE Katoxos diarkeias. Onomateponumo = Onoma Filathlou; SELECT Agonas.ID_Agona, Agonas.Imerominia **FROM Agonas** WHERE (Agonas.ID Agona, Agonas.Imerominia) NOT IN(SELECT DISTINCT Agonas.ID Agona, Agonas.Imerominia FROM Katoxos_diarkeias LEFT JOIN Filathlos ON Katoxos diarkeias. Onomateponumo = Filathlos. Onomateponumo LEFT JOIN Omada ON Filathlos.Omada_Onoma=Omada.Onoma LEFT JOIN Agonas ON (Omada.Onoma=Agonas.Onoma_Gipedouxou OR Omada.Onoma=Agonas.Onoma_Filoksenoumenou) LEFT JOIN antistoixei_diarkeias ON Agonas.ID_Agona=antistoixei_diarkeias.ID_Agona LEFT JOIN Diarkeias ON Diarkeias.Kodikos=antistoixei diarkeias.Diarkeias Kodikos LEFT JOIN Eisitirio ON Eisitirio.Kodikos = Diarkeias.Kodikos LEFT JOIN Agora ON Eisitirio.Kodikos = Agora.Eisitirio WHERE Agora. Agorastis = Katoxos_diarkeias. Onomateponumo AND Katoxos_diarkeias.Onomateponumo=Onoma_Filathlou) AND Agonas.Imerominia < CURRENT_DATE; **END** \$ **DELIMITER:** CALL ER5(); 6. DROP PROCEDURE IF EXISTS ER6: **DELIMITER \$** CREATE PROCEDURE ER6() DECLARE Onom VARCHAR(25); DECLARE Diff INT(3); SELECT Omada.Onoma, MAX(Omada.termata_dextike-Omada.termata_simeiose) AS Difference INTO Onom, Diff FROM Omada GROUP BY Omada.Onoma ORDER BY Difference LIMIT 0,1;

SELECT Omada. Gip Onoma, Proedros. Onomateponumo, Omada. Propon Onoma

LEFT JOIN Proedros ON Omada.Onoma=Proedros.Omada_Onoma

FROM Omada

WHERE Omada.Onoma=Onom;

END
\$
DELIMITER;
CALL ER6();

7. Η Βαθμολογία του πρωταθλήματος

SELECT Omada.Onoma as Omada,Omada.vathmoi as Vathmoi, Omada.nikes as Nikes, Omada.isopalies as Isopalies, Omada.ittes as Httes
FROM Omada

ORDER BY Omada.vathmoi DESC;

8. Ο νεότερος παίκτης του πρωταθλήματος και τα γκολ που έχει σημειώσει

SELECT Paiktis.Onomateponumo as Onoma, Paiktis.Onoma_Omadas as Omada,Paiktis.Ilikia as Ilikia, Paiktis.termata_simeiose as Goal FROM Paiktis
ORDER BY Paiktis.Ilikia ASC;

9. Η Ομάδα με τα περισσότερα γκολ που σκόραρε στο πρωτάθλημα

SELECT Omada.Onoma as Omada, Omada.termata_simeiose as Goal FROM Omada
ORDER BY Omada.termata_simeiose DESC LIMIT 0,1;

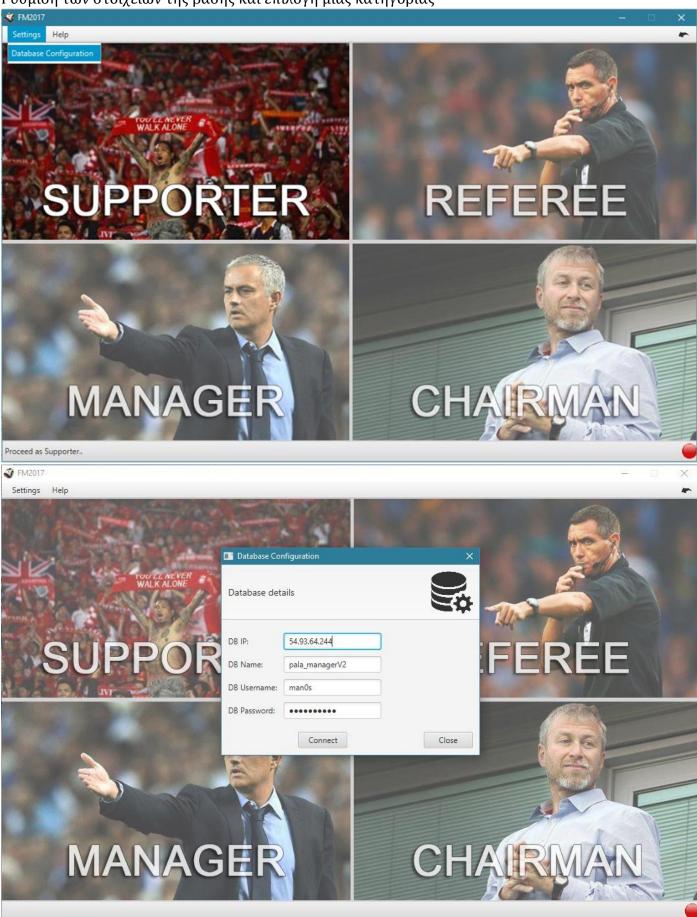
10. Ο αγώνας πρωταθλήματος με τις περισσότερες πωλήσεις εισιτηρίων

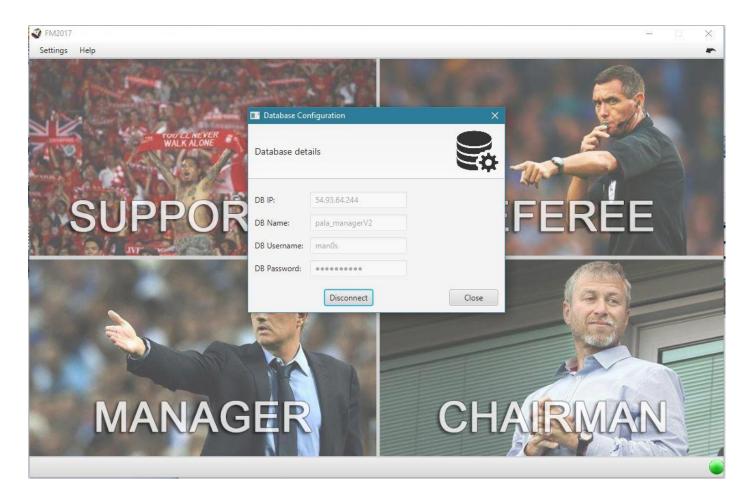
SELECT Agonas.Onoma_Gipedouxou as Gipedouxos, Agonas.Onoma_Filoksenoumenou as Filoksenoumenos, COUNT(Agora.Eisitirio) as Eisitiria FROM Agonas, Aplo, Eisitirio, Agora WHERE (Agonas.ID_Agona=Aplo.ID_Agona AND Aplo.Kodikos = Eisitirio.Kodikos AND Eisitirio.Kodikos = Agora.Eisitirio) GROUP BY Agonas.ID_Agona ORDER BY Eisitiria DESC LIMIT 0,1;

Εφαρμογή

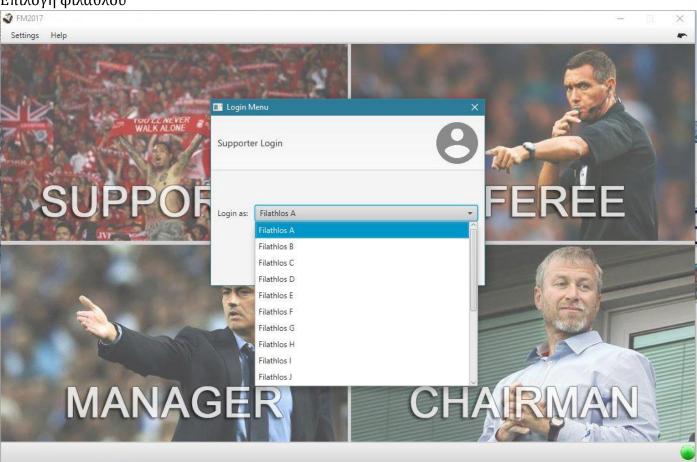
Μενού

Ρύθμιση των στοιχείων της βάσης και επιλογή μιας κατηγορίας





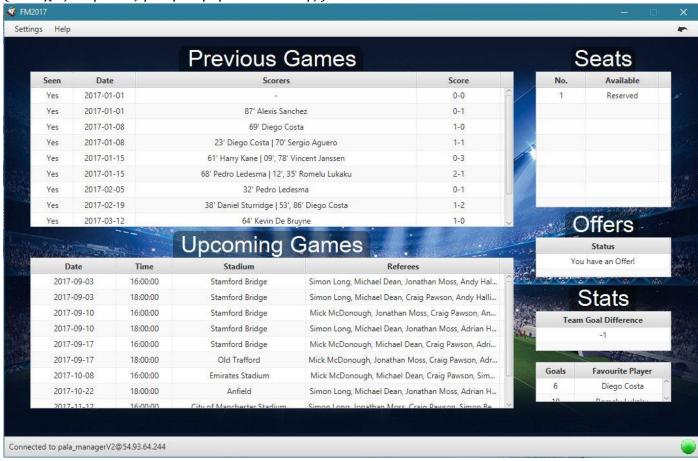
<u>Κατηγορία φιλάθλου</u> Επιλογή φίλαθλου



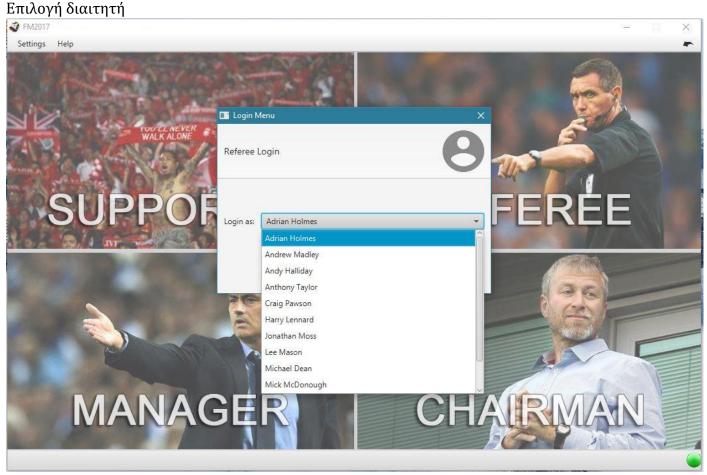
Προεπισκόπηση των στοιχείων του φιλάθλου(Κάτοχος διαρκείας-χωρίς προσφορά ανανέωσης)



(Κάτοχος διαρκείας-με προσφορά ανανέωσης)



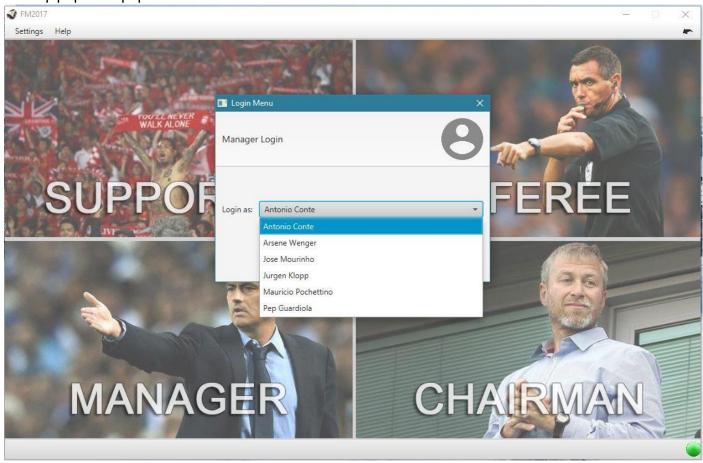
Κατηγορία διαιτητή



Προεπισκόπηση των στοιχείων του διαιτητή



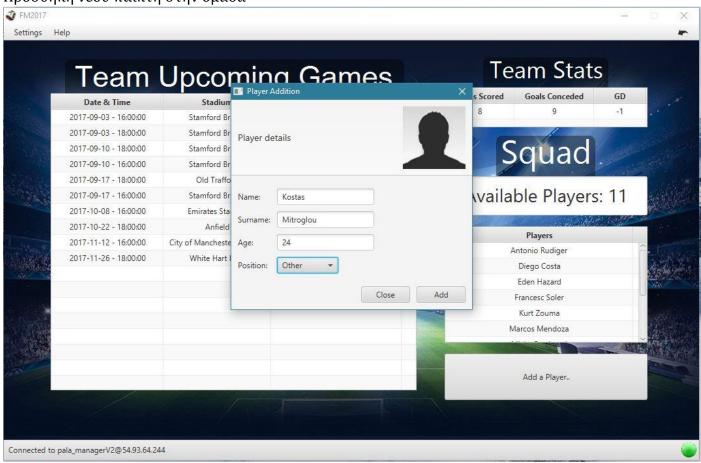
<u>Κατηγορία προπονητή</u> Επιλογή προπονητή



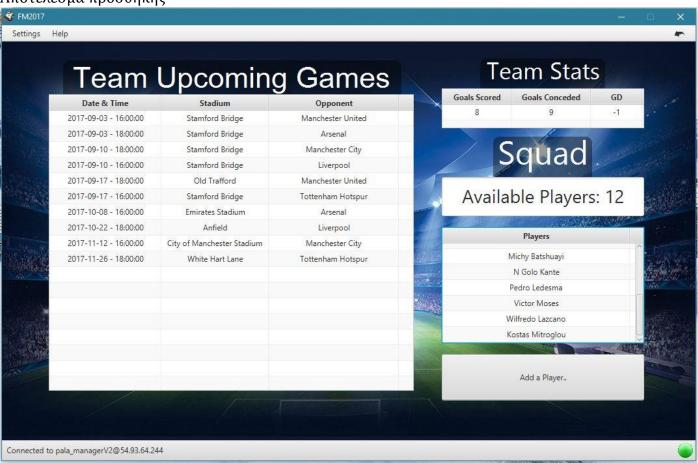
Προεπισκόπηση των στοιχείων του προπονητή



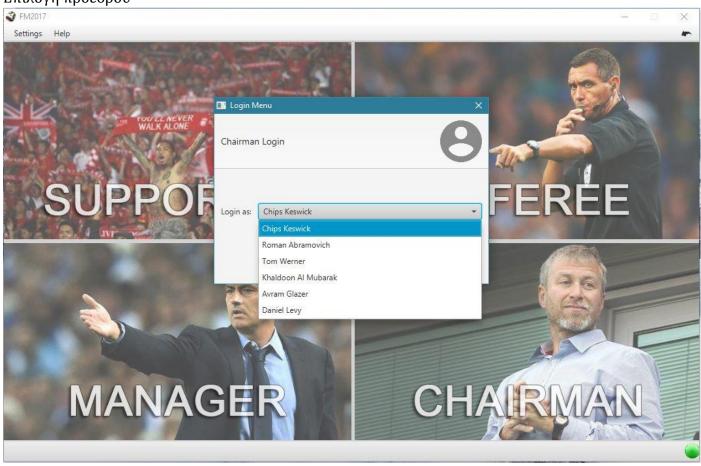
Προσθήκη νέου παίκτη στην ομάδα



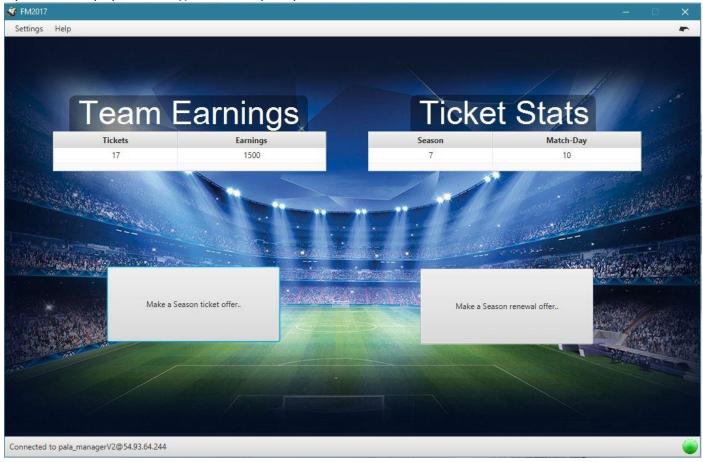
Αποτέλεσμα προσθήκης



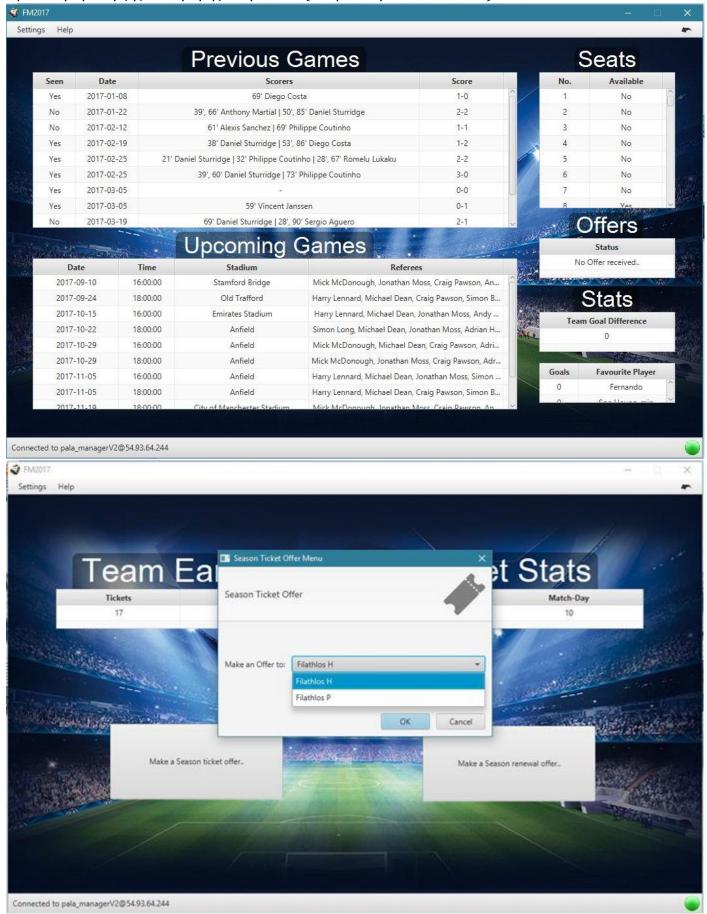
<u>Κατηγορία προέδρου</u> Επιλογή προέδρου



Προεπισκόπηση των στοιχείων του προέδρου



Πρόσκληση έναρξης συνδρομής σε φίλαθλο(επόμενα τρία screenshots)





Πρόσκληση ανανέωσης συνδρομής σε φίλαθλο(επόμενα τρία screenshots)



