

Αναφορά Έργου

Εφαρμογή Προβολής

Πληροφοριών Γευμάτων

ΕΑΠ – ΠΛΗ 24

3^η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ 2022-2023

Νο	Ονοματεπώνυμο	Α.Μ.
1	ΚΟΣΜΑΣ ΠΕΤΡΟΣ	100053
2	ΣΚΟΡΔΑΣ ΧΡΗΣΤΟΦΑΝΗΣ	128320
3	ΣΠΥΡΟΠΟΥΛΟΥ ΒΑΣΙΛΙΚΗ	080751
ΣΧΟΛΙΑ ΠΡΟΣ ΤΟΝ ΚΑΘΗΓΗΤΗ		

Υπεύθυνη Δήλωση Φοιτητή: Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία αυτής της εργασίας, είναι πλήρως αναγνωρισμένη και αναφέρεται, είτε στο σημείο «Σχόλια προς καθηγητή», είτε μέσα στην εργασία. Επίσης, έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς, είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη Θεματική Ενότητα.

☒ Συμφωνώ και αποδέχομαι την ανωτέρω δήλωση

☐ Δε συμφωνώ και δεν αποδέχομαι την ανωτέρω δήλωση (στην περίπτωση αυτή, ο Κ-Σ έχει δικαίωμα να μην αξιολογήσει την εργασία του φοιτητή)

Ημερομηνία ανακοίνωσης εργασίας:, 24/01/2023

Ημερομηνία παράδοσης εργασίας: 08/03/2023

ΔΕΝ ΘΑ ΔΟΘΟΥΝ ΠΑΡΑΤΑΣΕΙΣ

ΣΤΟΙΧΕΙΑ ΠΟΥ ΣΥΜΠΛΗΡΩΝΕΙ Ο ΚΑΘΗΓΗΤΗΣ

ΗΜΕΡΟΜΗΝΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	
ΒΑΘΜΟΣ	

ΣΧΟΛΙΑ ΠΡΟΣ ΦΟΙΤΗΤΗ / ΦΟΙΤΗΤΡΙΑ

Αναλυτική Αξιολόγηση

<u>Άσκηση</u>	<u>Περιγραφή</u>	<u>Ποσοστό</u>
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1	Διαχείριση του Έργου με έμφαση στη χρήση του εργαλείου συνεργασίας	20
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Α	Διάγραμμα ER, Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java (περιλαμβάνει χρήση API και επεξεργασία json)	20
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Β	Δημιουργία GUI Εφαρμογής	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Γ	Παρουσίαση Στατιστικών Στοιχείων	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Δ	Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3	Κριτικός Απολογισμός του Έργου	5
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4	Δημιουργία video	10
	Σύνολο	100

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	4
2	ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ	6
2.1	Υπολογισμός της απαιτούμενης προσπάθειας ανά απαίτηση	6
2.2	Υπολογισμός των προτεραιοτήτων	10
2.3	Χρονοδιάγραμμα του έργου	11
2.4	Το product backlog	12
2.5	Οργάνωση ομάδος και αναθέσεις αρμοδιοτήτων	14
2.6	Παρακολούθηση της προσπάθειας κατά τη διάρκεια του έργου	14
2.7	Χρήση εργαλείου trello.....	14
	ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2 - ΥΛΟΠΟΙΗΣΗ ΕΡΓΟΥ	16
2.8	Ερώτημα Α– Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java	16
2.9	Ερώτημα Β – Δημιουργία GUI Εφαρμογής.....	21
2.10	Ερώτημα Δ – Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής.....	30
3	ΚΡΙΤΙΚΟΣ ΑΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΕΡΓΟΥ.....	32
4	VIDEO	33
5	ΑΝΑΦΟΡΕΣ.....	34

1 ΕΙΣΑΓΩΓΗ

[Δώστε μια σύντομη περιγραφή του συστήματος. Παρουσιάστε τους βασικούς στόχους του έργου, καθώς και τις παραδοχές σας. Αναφέρετε τα χαρακτηριστικά και τις λειτουργίες που πρέπει να επιτελεί το έργο ώστε να ανταποκρίνεται στις απαιτήσεις και τις ανάγκες του πελάτη. Αναφέρετε με ποιόν/ποιους τρόπο/τρόπους το έργο ικανοποιεί αυτές τις απαιτήσεις.]

Η παρούσα εργασία αφορά την εκπόνηση της 3ης ομαδικής εργασίας για τη ΘΕ της ΠΛΗ-24 και περιλαμβάνει τον σχεδιασμό, την ανάπτυξη και την υλοποίηση της εφαρμογής MealsApp, η οποία επεξεργάζεται δεδομένα γευμάτων που λαμβάνει από την υπηρεσία ιστού (web service). Αναπτύξαμε μία εφαρμογή σε γραφικό περιβάλλον που ανταποκρίνεται στις πιο κάτω προδιαγραφές.

No	Περιγραφή
R1	<p>Το σύστημα περιλαμβάνει τις παρακάτω επιλογές:</p> <ol style="list-style-type: none"> 1. Προβολή δεδομένων γευμάτων 2. Προβολή λίστας γευμάτων ανά κατηγορία γεύματος 3. Προβολή στατιστικών δεδομένων γευμάτων και εκτύπωση σε αρχείο pdf 4. Έξοδος
R2	<p>Με την προβολή των δεδομένων γευμάτων εμφανίζεται οθόνη στην οποία ο χρήστης επιλέγει τα δεδομένα που επιθυμεί να αντληθούν από το API. Συγκεκριμένα, επιλέγει το όνομα ενός γεύματος, και εφόσον υπάρχει στο API, επιστρέφονται τα κατάλληλα δεδομένα.</p> <p>Οι λεπτομέρειες που μας ενδιαφέρουν να αντληθούν για κάθε γεύμα είναι οι ακόλουθες 4:</p> <ul style="list-style-type: none"> • strMeal (Το όνομα του γεύματος) • strCategory (Η κατηγορία στην οποία υπάγεται το γεύμα) • strArea (Η θεματική περιοχή του εν λόγω γεύματος) • strInstructions (Οι οδηγίες παρασκευής του γεύματος) <p>Εν συνεχεία, ο χρήστης έχει τη δυνατότητα να αποθηκεύσει τα δεδομένα που ανακτήθηκαν από το API σε ΒΔ. Σε αυτή την περίπτωση προηγείται έλεγχος ύπαρξης στη βάση δεδομένων των δεδομένων αυτών (από προηγούμενη αναζήτηση) και αν δεν υπάρχουν, αποθηκεύονται στη βάση δεδομένων.</p> <p>Στην οθόνη προβολής των δεδομένων εμφανίζονται επιπλέον τα πλήκτρα «Διαγραφή δεδομένων γεύματος» και «Επεξεργασία δεδομένων γεύματος»</p> <p>Με την επιλογή της διαγραφής δεδομένων γεύματος διαγράφονται από τη βάση δεδομένων τα σχετικά δεδομένα βάσει του ονόματος του εν λόγω γεύματος. Αντίστοιχα, με την επιλογή της τροποποίησης των δεδομένων ενός γεύματος, ο χρήστης έχει τη δυνατότητα να τροποποιήσει μία ή/και περισσότερες από τις αποθηκευμένες 4 λεπτομέρειες του κάθε γεύματος.</p> <p>Η ολοκλήρωση οποιασδήποτε διαγραφής ή επεξεργασίας δεδομένων γίνεται αφού προηγηθεί ερώτημα επαλήθευσης με επιβεβαίωση από το χρήστη.</p> <p>Στην οθόνη προβολής λίστας γευμάτων ανά κατηγορία γεύματος, ο χρήστης έχει τη δυνατότητα να επιλέξει από μια λίστα που περιέχει τις διαθέσιμες κατηγορίες γευμάτων μια κατηγορία που τον ενδιαφέρει. Με την επιλογή της κατηγορίας εμφανίζονται τα διαθέσιμα γεύματά της. Για κάθε γεύμα</p>

R3	που υπάρχει στην επιλεγμένη από τον χρήστη κατηγορία, μας ενδιαφέρει να εμφανίζονται μόνο τα ονόματα των γευμάτων και όχι άλλη πληροφορία. (Αν κάποιος χρήστης ενδιαφέρεται περαιτέρω για ένα γεύμα, μπορεί απλά να το αναζητήσει βάσει ονόματος, κάνοντας χρήση της προηγούμενης λειτουργίας).
R4	Με την επιλογή προβολής στατιστικών δεδομένων γευμάτων και εκτύπωση σε αρχείο pdf, ουσιαστικά αναζητάμε μέσα στην ΒΔ που έχει δημιουργηθεί, ποια είναι τα πιο δημοφιλή γεύματα, δηλαδή τα γεύματα που έχει προβάλλει ο χρήστης της εφαρμογής περισσότερες φορές. Τα στατιστικά αυτά πρέπει να μπορούν να προβληθούν υπό τη μορφή πίνακα (δημοφιλέστερα γεύματα σε φθίνουσα μορφή), ενώ στη συνέχεια με την ύπαρξη αντίστοιχης επιλογής, δίνεται η δυνατότητα εκτύπωσης του παραγόμενου πίνακα σε αρχείο PDF. Για την εν λόγω απαίτηση θα χρειαστεί να υλοποιησετε/τροποποιήσετε κατάλληλα τους πίνακες της ΒΔ της εφαρμογής, ώστε να καταγράφετε και ποιο γεύμα έχει προβληθεί και πόσες φορές.

Για την υλοποίηση του λογισμικού χρησιμοποιήσαμε τα παρακάτω εργαλεία – προγράμματα:

- Την εφαρμογή Visual Paradigm for UML για τον σχεδιασμό των διαγραμμάτων της UML.
- Την εφαρμογή NetBeans IDE 16 για την ανάπτυξη λογισμικού σε Java.
- Την βάση δεδομένων Derby
- Την μέθοδο planning poker - <http://www.planningpoker.com/>).
- Την εφαρμογή mindview για το χρονοδιάγραμμα του έργου σε μορφή Gantt.
- Την εφαρμογή MS Excel για το product backlog και το διάγραμμα κατανάλωσης προσπάθειας (burnt down chart) ανά sprint.
- Το συνεργατικό εργαλείο για τη διαχείριση έργων <http://Trello.com>

Για να λειτουργεί σωστά η εφαρμογή έχουμε προσθέσει στο Project στον φάκελο \lib τις ακόλουθες βιβλιοθήκες.

1. derbyclient.jar (Βιβλιοθήκη για την σύνδεση με την database)
2. gson-2.8.5.jar (Βιβλιοθήκη για JSON Parsing)
3. itextpdf-5.5.10.jar (Βιβλιοθήκη για την δημιουργία pdf αρχείων)
4. kotlin-stdlib-1.3.61.jar (Kotlin Standard Library for JVM. OkHttp 4 the implementation language is Kotlin)
5. okhttp-4.2.1.jar (Βιβλιοθήκη για εκτέλεση μεθόδων HTTP)
6. okio-2.4.3.jar (A modern I/O API for Java. Okio is a library that complements java.io and java.nio to make it much easier to access, store, and process your data. It started as a component of OkHttp)

2 ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ

2.1 Υπολογισμός της απαιτούμενης προσπάθειας ανά απαίτηση

[Θα πρέπει να υπολογίσετε την απαιτούμενη προσπάθεια για το έργο σε ανθρωπομήρες με τη χρήση της μεθόδου *planning poker*. Να τεκμηριώσετε τις παραδοχές σας, το υλικό στο οποίο βασιστήκατε για την εφαρμογή της μεθόδου, καθώς και τον τρόπο εργασίας. Να παραθέσετε σχετικές αναφορές. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

Στο δεύτερο Webex-meeting, αφού πρώτα μελετήσαμε τις απαιτήσεις του συστήματος η Ομάδα ανάπτυξης (Scrum Team) συζητήσαμε τα παρακάτω:

Τι χρειάζεται να μάθουμε πριν να ξεκινήσουμε να δουλεύουμε το project; Χρειαζόταν να πειραματιστούμε με το JAVA Jframe form λόγω ότι δεν είχαμε καθόλου γνώση του γραφικού περιβάλλοντος, να πειραματιστούμε με τα Bind ώστε να μάθουμε να τα δουλεύουμε και να συνδέουμε την βάση των δεδομένων με το γραφικό περιβάλλον.

Πόσο κώδικα θα γράψουμε για αυτό το project; Έχουμε γράψει παρόμοιο κώδικα άλλη φορά; Διαπιστώσαμε καθώς είδαμε και Εργασίες προηγούμενων ετών ότι χρειάζεται να γράψουμε πολύ κώδικα και ότι σε αρκετά σημεία ήμασταν ελλιπής.

Τι θα χρειαστούμε για να δοκιμάσουμε αυτό το project; Για να μπορέσουμε να δοκιμάσουμε το project θα έπρεπε να έχουμε φτιάξει κάτι στο 1o spring.

Μπορούμε να ενοποιήσουμε εύκολα αυτό το project; Και εδώ είδαμε ότι θα συναντήσουμε αρκετά προβλήματα καθώς θα έπρεπε να ενώσουμε τον κώδικα πράγμα που μας φάνηκε αρκετά δύσκολο λόγω ότι δεν είχαμε τέτοια εμπειρία στο παρελθόν.

Για την διαχείριση του έργου, χρησιμοποιήθηκε η μέθοδος Scrum που είναι μια επαναληπτική και αυξητική μεθοδολογία ανάπτυξης έργων.

Χρησιμοποιείται σαν μια μέθοδο ανάπτυξης λογισμικού, όσον αφορά την οργάνωση και λειτουργία των ομάδων ανάπτυξης αλλά και στον σχεδιασμό και διαχείριση των ευέλικτων έργων.

Για τη διαχείριση έργου οι βασικοί ρόλοι είναι οι ακόλουθοι:

1. Ο Product Owner, το ρόλο αυτό παίζει ο ΣΕΠ.
2. Την ομάδα ανάπτυξης Scrum Team η οποία αποτελείται από τρία άτομα. Η ομάδα Scrum είναι επιφορτισμένη με την ανάπτυξη του έργου και έχει την εξουσιοδότηση να αποφασίζει και να εκτελεί τις κατάλληλες διαδικασίες και εργασίες που απαιτούνται για την υλοποίηση του.

3. Τον διαχειριστή του έργου Scrum Master ο οποίος βοηθά τα μέλη της ομάδας να μάθουν και να εφαρμόσουν καλύτερα τις πρακτικές που ακολουθούνται στην διαδικασία του Scrum. Είναι υπεύθυνος στο να διασφαλίσει ότι τυχόν εμπόδια κατά τη διάρκεια του έργου θα εξαλειφθούν, έτσι ώστε η ομάδα ανάπτυξης να διατηρήσει όσο το δυνατόν ψηλότερα το επίπεδο παραγωγικότητάς της.

Για τον υπολογισμό της απαιτούμενης προσπάθειας ανά απαίτηση χρησιμοποιήθηκε η μέθοδος του planning poker όπως βρίσκεται στην διεύθυνση: <http://www.planningpoker.com>

Ο Scrum Master δημιουργεί ένα παιχνίδι και το κοινοποιεί στην υπόλοιπη ομάδα. Κατά το παιχνίδι, οι παίκτες χρησιμοποιούν μια προκαθορισμένη με τιμές τράπουλα. Η τιμή κάθε τραπουλόχαρτου αντιστοιχεί σε ανθρωποώρες εργασίας που απαιτούνται για την υλοποίηση των εργασιών.

Για κάθε απαίτηση, ο καθένας από εμάς, με βάση την εμπειρία του, δίνει την εκτίμηση του ως προς την προσπάθεια (σε ανθρωποώρες) που θα καταβάλει προκειμένου να την υλοποιήσει. Σε περίπτωση διαφωνίας ακολουθεί 2ος ή 3ος γύρος ώστε να επέλθει συμφωνία.

Ως κλίμακα βαθμολόγησης για το παρόν έργο επιλέχθηκε η κλίμακα «T-Shirt Sizes» με εύρος τομών XXS, XS, S, M, L, XL, XXL αντιπροσωπεύοντας ανθρωποώρες, όντας το πρακτικότερο.

Εδώ διαπιστώσαμε ότι δεν είχαμε ξανά εργαστεί ομαδικά σε κανένα project και ήταν κάτι νέο και μία πρόκληση για εμάς.

Αφού λοιπόν συζητήσαμε μεταξύ μας μοιράστηκαν στην ομάδα κάρτες «T-Shirt Sizes» με την παραδοχή ότι το XXS αντιστοιχεί στην μικρότερη προτεραιότητα και το XXL στη μεγαλύτερη και ξεκινήσαμε να ρίχνουμε τις κάρτες.

Ο καθένας μας έβαλε μία κάρτα σύμφωνα με τον βαθμό δυσκολίας που πίστευε ότι είχε η κάθε απαίτηση του συστήματος.

Γνωριμία ομάδας

STORY 1/22

XXS Pass XXS

Vassiliki S... Christofan... Petros Ko...

XXS

xs s m l xl xxl ? Pass

Settings Vassiliki Spyros

XXS

Edit Score

Story Score

Edit Story Close

1	Γνωριμία ομάδας	xxs
2	Ανάλυση απαιτήσεων, ανάθεση αρμοδιοτήτων	xl
3	Προγραμματισμός εργασιών, sprint και καρτελών Trello	xxl
4	Κατασκευή Διαγράμματος κλάσεων	m
5	Δημιουργία Βάσης Δεδομένων	l
6	Κατασκευή GUI	xl

PlanningPoker.com - Sprints Made Simple. Estimates Made Easy. - Google Chrome

play.planningpoker.com/view_score/sW7T3XES2OXCK9DiXsUHYmOHxdMymSsR

Your Planning Poker® Game Summary

The Meal

Εφαρμογή Προβολής Πληροφοριών Γευμάτων

Story	Story Title	Score
1	Γνωριμία ομάδας	xxs
2	Ανάλυση απαιτήσεων, ανάθεση αρμοδιοτήτων	xl
3	Προγραμματισμός εργασιών, sprint και καρτελών Trello	xxl
4	Κατασκευή Διαγράμματος κλάσεων	m
5	Δημιουργία Βάσης Δεδομένων	l
6	Κατασκευή GUI	xl
7	Υλοποίηση απαιτήσης R1	l
8	Έλεγχος/debug R1	l
9	Υλοποίηση απαιτήσης R2	xl
10	Έλεγχος/debug R2	xl
11	Υλοποίηση απαιτήσης R3	xl
12	Έλεγχος/debug R3	xl
13	Υλοποίηση απαιτήσης R4	xxl
14	Έλεγχος/debug R4	xl
15	Διάγραμμα Gantt	m
16	Burn down chart και υπολογισμός ταχύτητας παράδοσης	m
17	Ενοποίηση - βελτίωση κώδικα	xl
18	Βελτίωση GUI	l
19	Τελικός έλεγχος εφαρμογής	l
20	Δημιουργία Video	l
21	Απολογισμός Έργου	m
22	Συγγραφή εργασίας	l

PlanningPoker.com - Sprints Made Simple. Estimates Made Easy. - Google Chrome

play.planningpoker.com/view_score/sW7T3XES2OXCK9DiXsUHYmOHxdMymSsR

Your Planning Poker® Game Summary

The Meal

Εφαρμογή Προβολής Πληροφοριών Γευμάτων

Story	Story Title	Score
1	Γνωριμία ομάδας	xxs
2	Ανάλυση απαιτήσεων, ανάθεση αρμοδιοτήτων	xl
3	Προγραμματισμός εργασιών, sprint και καρτελών Trello	xxl
4	Κατασκευή Διαγράμματος κλάσεων	m
5	Δημιουργία Βάσης Δεδομένων	l
6	Κατασκευή GUI	xl
7	Υλοποίηση απαίτησης R1	l
8	Έλεγχος/debug R1	l
9	Υλοποίηση απαίτησης R2	xl
10	Έλεγχος/debug R2	xl
11	Υλοποίηση απαίτησης R3	xl
12	Έλεγχος/debug R3	xl
13	Υλοποίηση απαίτησης R4	xxl
14	Έλεγχος/debug R4	xl
15	Διάγραμμα Gantt	m
16	Burn down chart και υπολογισμός ταχύτητας παράδοσης	m
17	Ενοποίηση - βελτίωση κώδικα	xl
18	Βελτίωση GUI	l
19	Τελικός έλεγχος εφαρμογής	l
20	Δημιουργία Video	l
21	Απολογισμός Έργου	m
22	Συγγραφή εργασίας	l

Αφού τελειώσαμε με τις κάρτες ξεκινήσαμε την συζήτηση για τις απαιτήσεις στις οποίες ο βαθμός δυσκολίας με βάση τις κάρτες που ρίξαμε ήταν διαφορετικές.
Τελικά συμφωνήσαμε στις απαιτήσεις του συστήματος με τους παρακάτω βαθμούς δυσκολίας και την Αναγωγή σε ημέρες.

Περιγραφή	Βαθμός Δυσκολίας	Αναγωγή σε ημέρες
-----------	---------------------	-------------------------

Ανάλυση απαιτήσεων, ανάθεση αρμοδιοτήτων	XL	5
Προγραμματισμός εργασιών, sprint και καρτελών Trello	XL	4
Κατασκευή Διαγράμματος κλάσεων	M	3
Δημιουργία Βάσης Δεδομένων	L	4
Κατασκευή GUI	XL	5
Υλοποίηση απαίτησης R1	L	4
Έλεγχος/debug R1	L	4
Υλοποίηση απαίτησης R2	XL	5
Έλεγχος/debug R2	L	4
Υλοποίηση απαίτησης R3	XL	5
Έλεγχος/debug R3	L	4
Υλοποίηση απαίτησης R4	XL	5
Έλεγχος/debug R4	L	4
Ενοποίηση - βελτίωση κώδικα	XL	3
Βελτίωση GUI	L	3
Τελικός έλεγχος εφαρμογής	L	2
Δημιουργία Video	M	2
Απολογισμός Έργου	M	2
Συγγραφή εργασίας	L	2

2.2 Υπολογισμός των προτεραιοτήτων

[Θα πρέπει να υπολογίσετε την προτεραιότητα της κάθε απαίτησης χρησιμοποιώντας τη μέθοδο Priority Poker. Στο σημείο αυτό θα παρουσιάσετε το τελικό αποτέλεσμα από την εφαρμογή της μεθόδου (ποιες απαιτήσεις θα υλοποιηθούν σε κάθε επανάληψη), καθώς και παρατηρήσεις σχετικές με τον τρόπο εργασίας σας. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

PlanningPoker.com - Sprints Made Simple. Estimates Made Easy. - Google Chrome

play.planningpoker.com/view_score/8ad3RCW3689APpbPJbptUDgHYPAHNj0A

Your Planning Poker® Game Summary

PRIORITY

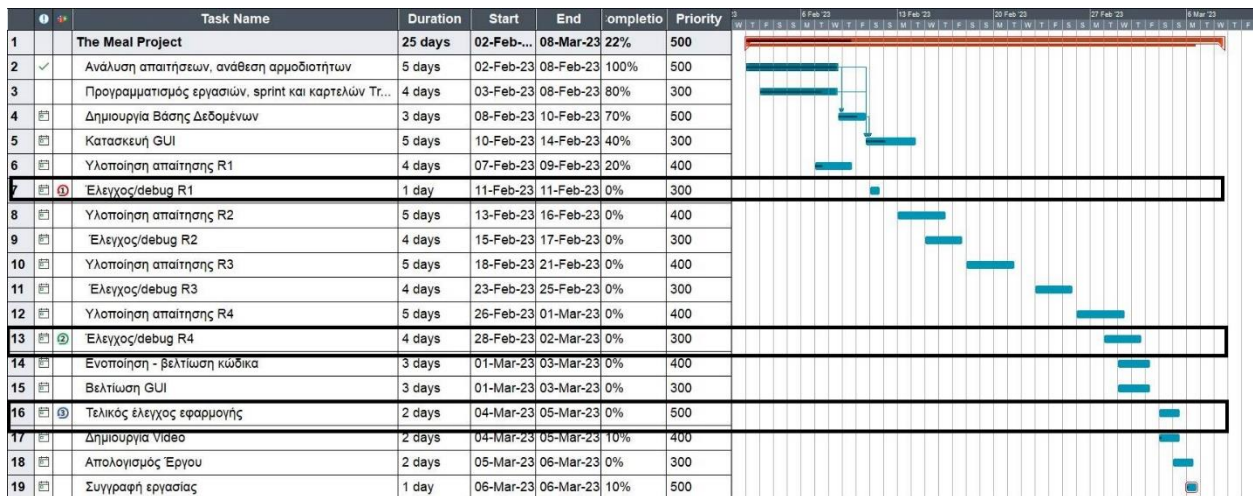
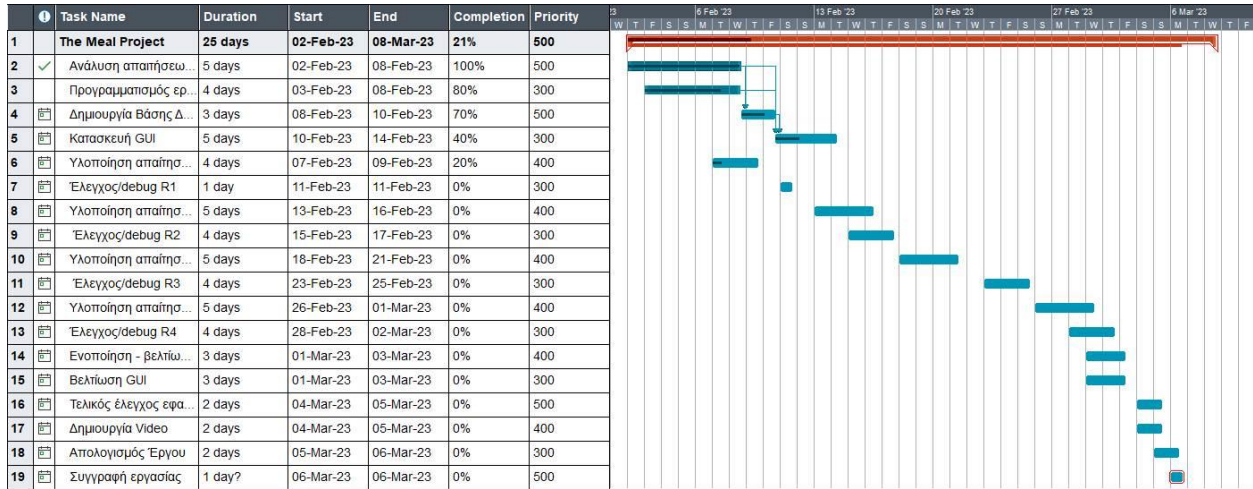
Προτεραιοποίηση των εργασιών **tasks**, ώστε να αποφασιστεί ποιες απαιτήσεις θα συμπεριληφθούν σε κάθε **sprint**

Story	Story Title	Score
1	R1 - Το σύστημα περιλαμβάνει τις παρακάτω επιλογές:	-
2	1. Προβολή δεδομένων γευμάτων	VHP
3	2. Προβολή λίστας γευμάτων ανά κατηγορία γεύματος	VHP
4	3. Προβολή στατιστικών δεδομένων γευμάτων και εκτύπωση σε αρχείο pdf	VHP
5	4. Έξοδος	LP
6	R2 - Με την προβολή των δεδομένων γευμάτων εμφανίζεται οθόνη στην οποία ο χρήστης επιλέγει τα δεδομένα που επιθυμεί να αντληθούν από το API. Συγκεκριμένα, επιλέγει το όνομα ενός γεύματος, και εφόσον υπάρχει στο API, επιστρέφονται τα κατάλληλα δεδομένα.	HP
7	R3 - Στην οθόνη προβολής λίστας γευμάτων ανά κατηγορία γεύματος, ο χρήστης έχει τη δυνατότητα να επιλέξει από μια λίστα που περιέχει τις διαθέσιμες κατηγορίες γευμάτων μια κατηγορία που τον ενδιαφέρει. Με την επιλογή της κατηγορίας εμφανίζονται τα διαθέσιμα γεύματά της. Για κάθε γεύμα που υπάρχει στην επιλεγμένη από τον χρήστη κατηγορία, μας ενδιαφέρει να εμφανίζονται μόνο τα ονόματα των γευμάτων και όχι άλλη πληροφορία.	MP
8	R4 - 1. Με την επιλογή προβολής στατιστικών δεδομένων γευμάτων και εκτύπωση σε αρχείο pdf, ουσιαστικά αναζητάμε μέσα στην ΒΔ που έχει δημιουργηθεί, ποια είναι τα πιο δημοφιλή γεύματα, δηλαδή τα γεύματα που έχει προβάλλει ο χρήστης της εφαρμογής περισσότερες φορές.	HP
9	2. Τα στατιστικά αυτά πρέπει να προβληθούν υπό τη μορφή πίνακα (δημοφιλέστερα γεύματα σε φθίνουσα μορφή), ενώ στη συνέχεια με την ύπαρξη αντίστοιχης επιλογής, δίνεται η δυνατότητα εκτύπωσης του παραγόμενου πίνακα σε αρχείο PDF.	HP
10	3. Υλοποίηση/τροποποίηση πινάκων της ΒΔ της εφαρμογής, ώστε να καταγράφεται και ποιο γεύμα έχει προβληθεί και πόσες φορές.	MP

2.3 Χρονοδιάγραμμα του έργου

[Εισάγετε σχετική εικόνα. Στην Εικόνα θα πρέπει να φαίνεται το χρονοδιάγραμμα του έργου σε μορφή Gantt (που θα έχει κατασκευαστεί με χρήση του σχετικού εργαλείου). Παρουσιάστε με συντομία και με μορφή κειμένου τις βασικές δραστηριότητες της κάθε επανάληψης. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

Στην συνέχεια υλοποιήθηκε το χρονοδιάγραμμα του έργου σε μορφή Gantt. Η αποτύπωση του χρονοδιαγράμματος έγινε μετά τον υπολογισμό των απαιτούμενων προσπαθειών ανά απαίτηση. Καθώς και οι τρεις επαναλήψεις (sprint).

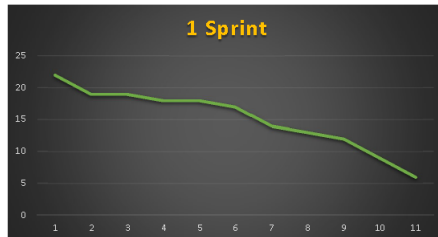


2.4 Το product backlog

[Θα πρέπει να παρουσιάσετε το product backlog σε μορφή φύλλου εργασίας (xls) που θα περιέχει τρία φύλλα εργασίας, ένα για κάθε επανάληψη, μαζί με τις συμπληρωματικές/βοηθητικές πληροφορίες που αποφασίσατε να συλλέξετε. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

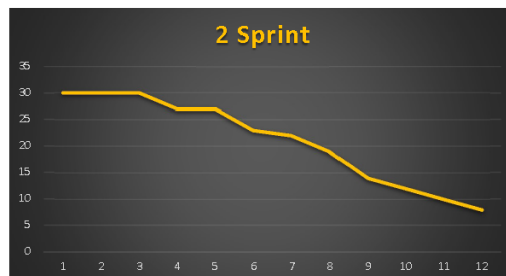
The Meal Project

1 Sprint		02-02-23	03-02-23	04-02-23	05-02-23	06-02-23	07-02-23	08-02-23	09-02-23	10-02-23	11-02-23	12-02-23
1	Ανάλυση απαιτήσεων, ανάθεση αρμοδιοτήτων	5	5	4	4	4	4	3	3	2	2	1
2	Προγραμματισμός εργασιών, sprint και καρτελών Trello	4	4	4	4	4	4	3	4	3	2	1
3	Δημιουργία Βάσης Δεδομένων	3	3	3	3	2	2	2	2	1	1	1
4	Κατασκευή GUI	5	5	4	4	4	4	3	2	2	1	1
5	Υλοποίηση απαίτησης R1	4	4	3	3	3	3	2	2	3	2	1
6	Έλεγχος/debug R1	1	1	1	1	1	1	1	1	1	1	1
Συνολική Προσπάθεια		22	22	19	19	18	18	17	14	13	12	9

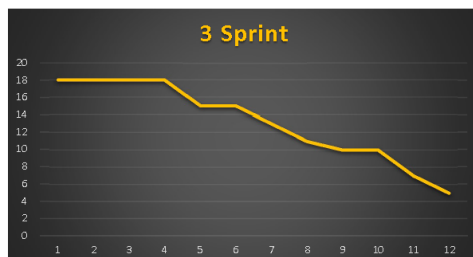


The Meal Project

2 Sprint		13-02-23	14-02-23	15-02-23	16-02-23	17-02-23	18-02-23	19-02-23	20-02-23	21-02-23	22-02-23	23-02-23
1	Υλοποίηση απαίτησης R2	5	5	5	4	4	3	3	3	2	2	1
2	Έλεγχος/debug R2	4	4	4	4	4	3	3	3	2	1	1
3	Υλοποίηση απαίτησης R3	5	5	5	4	4	3	3	2	2	1	1
4	Έλεγχος/debug R3	4	4	4	4	4	3	3	2	2	1	1
5	Υλοποίηση απαίτησης R4	5	5	5	4	4	3	3	3	2	2	1
6	Έλεγχος/debug R4	4	4	4	4	4	4	3	2	2	2	1
7	Ενοποίηση - βελτίωση κώδικα	3	3	3	3	3	3	2	2	2	2	2
Συνολική Προσπάθεια		30	30	30	27	27	23	22	19	14	12	8



3 Sprint		14-02-23	15-02-23	16-02-23	17-02-23	18-02-23	19-02-23	20-02-23	21-02-23	22-02-23	23-02-23	24-02-23
1	Βελτίωση GUI	5	5	5	4	4	3	3	2	2	1	1
2	Τελικός έλεγχος εφαρμογής	4	4	4	3	3	3	2	2	2	1	1
3	Δημιουργία Video	3	3	3	2	2	2	2	2	2	1	1
4	Απολογισμός Έργου	3	3	3	3	3	2	2	2	2	1	1
5	Συγγραφή εργασίας	3	3	3	3	3	3	2	2	2	2	1
Συνολική Προσπάθεια		18	18	18	15	15	13	11	10	10	7	5



2.5 Οργάνωση ομάδος και αναθέσεις αρμοδιοτήτων

[Αναφέρετε την οργάνωση της ομάδας του έργου, καθώς και τους ρόλους των μελών της ομάδας. Πιο συγκεκριμένα:

- Ποιος ήταν ο ρόλος του κάθε μέλους της ομάδας;
- Ποιος ήταν ο υπεύθυνος για την υλοποίηση της κάθε απαίτησης;
- Αναφέρετε τον τρόπο επικοινωνίας των μελών της ομάδας, τη διαχείριση διαφορών μεταξύ των μελών, κ.λπ. και ποιες από τις πρακτικές του SCRUM χρησιμοποιήσατε;

Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

2.6 Παρακολούθηση της προσπάθειας κατά τη διάρκεια του έργου

[Παρουσιάστε το χρόνο υλοποίησης ανά απαίτηση/ανά επανάληψη (sprint) σε σύγκριση με αυτά που είχαν προϋπολογιστεί, το διάγραμμα κατανάλωσης προσπάθειας (burnt down chart) ανά sprint, καθώς και την ταχύτητα υλοποίησης (velocity). Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]

2.7 Χρήση εργαλείου trello

Παρουσιάστε τον τρόπο χρήσης του εργαλείου:

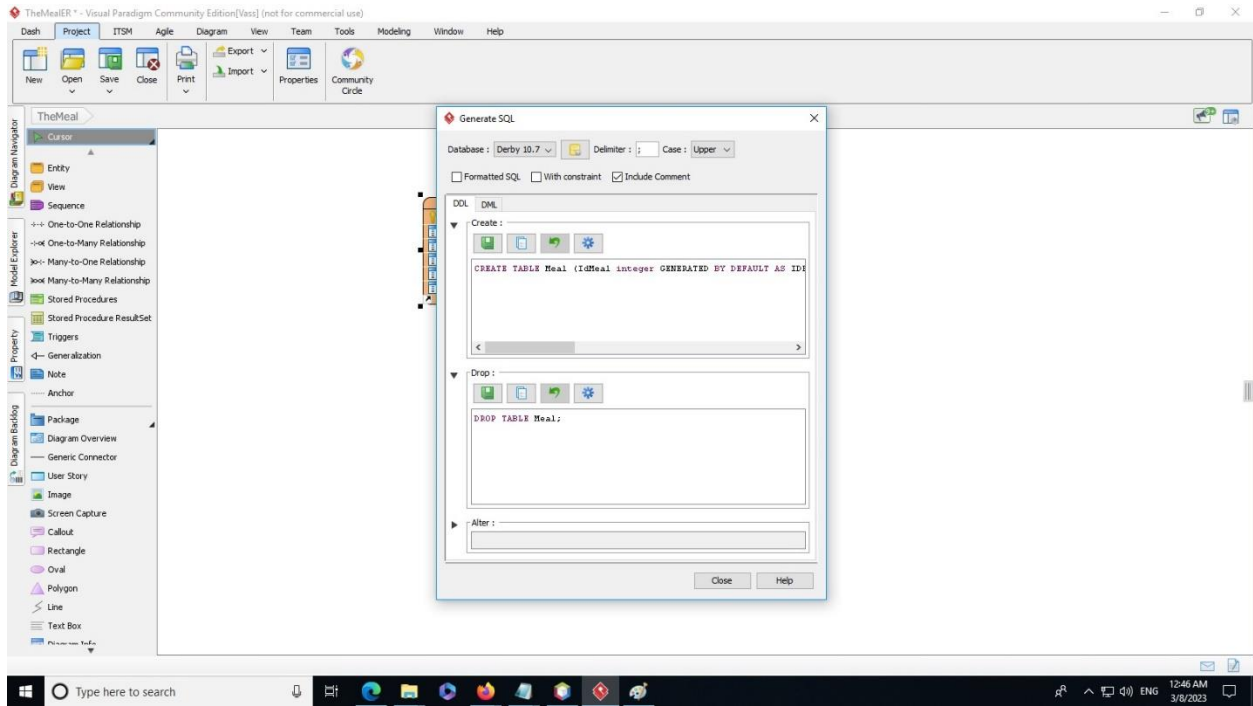
1. Πώς οργανώσατε το Agile Board, ποιες λίστες χρησιμοποιήσατε. Δώστε ενδεικτικό Screenshot.
2. Πώς ορίσατε τις κατηγορίες δραστηριοτήτων
3. Πώς καταγράψατε τον χρόνο που χρειάστηκε η κάθε δραστηριότητα
4. Πώς συνεργάστηκε η ομάδα στο Agile Board
5. Πως εκμεταλλευτήκατε τις λειτουργίες του Trello. Ποιες σας φάνηκες χρήσιμες, ποιες εύκολες στη χρήση και ποιες όχι.
6. Αναλύστε ποιες συνδέσεις με τρίτες εφαρμογές χρησιμοποιήσατε και πώς. Πχ με το github.

Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ]







Για τις ανάγκες του έργου, ο Scrum Master της ομάδας δημιούργησε ένα λογαριασμό στην πλατφόρμα του Trello, που βρίσκεται στον δικτυακό τόπο trello.com. Το Trello είναι μια εφαρμογή διαχείρισης λιστών τύπου Kanban, το οποίο δίνει τη δυνατότητα οργάνωσης σε μια Scrum Team, την παρακολούθηση και γενικά διαχείριση των έργων λογισμικού που έχει αναλάβει.

The screenshot shows the Trello interface for a workspace named "Ergasia 3 - plh 24". The left sidebar contains navigation options: Boards, Members (selected), Workspace settings, Workspace views (Table, Calendar), and Your boards (1o Sprint, 2o Sprint, 3o Sprint, Backlog). The main content area is titled "Members" and shows "Workspace members (3)". A purple banner promotes upgrading to Premium for more permissions. Below, a list of members is shown: Christofanis Skordas (@std128320), VASSILIKI (@vassiliki17), and ΚΟΣΜΑΣ ΠΕΤΡΟΣ (@user36226341). An "Invite members to join you" section explains that anyone with an invite link can join this Free Workspace.

The screenshot shows the Trello interface for the same workspace, now with the "Boards" tab selected. The left sidebar is identical. The main content area is titled "Boards" and shows "Showing 5 of 5 boards". At the top, there are filters for "Sort by" (Most recently active) and "Filter by" (Choose a collection), along with a search bar. The boards displayed are: "Create new board" (5 remaining), "3o Sprint", "Backlog", "1o Sprint", "2o Sprint", and "Ergasia3".



(β) Εισάγεται των κώδικα για τις κλάσεις του συστήματος και

Meal		
	IdMeal	integer(10)
	strMeal	varchar(255) N
	strCategory	varchar(255) N
	strArea	varchar(255) N
	strInstructions	varchar(2550) N
	status	integer(10) N

CREATE TABLE Meal (IdMeal integer GENERATED BY DEFAULT AS IDENTITY, strMeal varchar(255), strCategory varchar(255), strArea varchar(255), strInstructions varchar(2550), status integer, PRIMARY KEY (IdMeal));

(γ) Εισάγετε τον κώδικα java για τις ροjo κλάσεις. Θα πρέπει να εισάγεται ΜΟΝΟ τον κώδικα που έχετε γράψει εσείς και όχι αυτόν που παράγεται αυτόματα προσδιορίζοντας ευκρινώς το όνομα της κλάσης και της μεθόδου. Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει τεκμηρίωση με μορφή σχολίων. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ.]

Meal.java

```
package org.mealsapp.model;

import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
```

```
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;

import java.io.Serializable;

@Entity
@Table(name = "MEAL")
@NamedQueries({
    @NamedQuery(name = "Meal.findAll", query = "SELECT m FROM Meal m"),
    @NamedQuery(name = "Meal.findByIdmeal", query = "SELECT m FROM Meal m
WHERE m.idmeal = :idmeal"),
    @NamedQuery(name = "Meal.findByIdStrmeal", query = "SELECT m FROM Meal
m WHERE m.strmeal = :strmeal"),
    @NamedQuery(name = "Meal.findByIdStrcategory", query = "SELECT m FROM
Meal m WHERE m.strcategory = :strcategory"),
    @NamedQuery(name = "Meal.findByIdStrarea", query = "SELECT m FROM Meal
m WHERE m.strarea = :strarea"),
    @NamedQuery(name = "Meal.findByIdStrinstructions", query = "SELECT m
FROM Meal m WHERE m.strinstructions = :strinstructions"),
    @NamedQuery(name = "Meal.findByIdStatus", query = "SELECT m FROM Meal m
WHERE m.status = :status"))})
public class Meal implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "IDMEAL")
    private Integer idmeal;
    @Column(name = "STRMEAL")
    private String strmeal;
    @Column(name = "STRCATEGORY")
    private String strcategory;
    @Column(name = "STRAREA")
    private String strarea;
    @Column(name = "STRINSTRUCTIONS")
    private String strinstructions;
    @Column(name = "STATUS")
    private Integer status;

    public Meal() {
        // Constructor
    }

    public Meal(Integer idmeal) {
        this.idmeal = idmeal;
    }

    public Integer getIdmeal() {
        return idmeal;
    }

    public void setIdmeal(Integer idmeal) {
        this.idmeal = idmeal;
    }
}
```

```
public String getStrmeal() {
    return strmeal;
}

public void setStrmeal(String strmeal) {
    this.strmeal = strmeal;
}

public String getStrcategory() {
    return strcategory;
}

public void setStrcategory(String strcategory) {
    this.strcategory = strcategory;
}

public String getStrarea() {
    return strarea;
}

public void setStrarea(String strarea) {
    this.strarea = strarea;
}

public String getStrinstructions() {
    return strinstructions;
}

public void setStrinstructions(String strinstructions) {
    this.strinstructions = strinstructions;
}

public Integer getStatus() {
    return status;
}

public void setStatus(Integer status) {
    this.status = status;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (idmeal != null ? idmeal.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (!(object instanceof Meal)) {
        return false;
    }
    Meal other = (Meal) object;
    if ((this.idmeal == null && other.idmeal != null) || (this.idmeal !=
null && !this.idmeal.equals(other.idmeal))) {
```

```
        return false;
    }
    return true;
}

@Override
public String toString() { return "org.mealsapp.model.Meal[ idmeal=" +
idmeal + " ]"; }
}
```

MealJpaController

```
package org.mealsapp.controller;

import jakarta.persistence.EntityManager;
import org.mealsapp.model.MainModel;
import org.mealsapp.model.Meal;
import org.mealsapp.model.NoneexistentEntityException;
import org.mealsapp.model.PreexistingEntityException;

import jakarta.persistence.EntityNotFoundException;
import java.io.Serializable;

public class MealJpaController implements Serializable {

    private MainModel mainModel;
    private EntityManager entityManager;

    public MealJpaController(MainModel mainModel) {
        this.mainModel = mainModel;
        this.entityManager = mainModel.entityManager;
    }

    public void create(Meal meal) throws PreexistingEntityException,
Exception {
        try {
            entityManager.getTransaction().begin();
            entityManager.persist(meal);
            entityManager.getTransaction().commit();
        } catch (Exception ex) {
            if (findMeal(meal.getIdmeal()) != null) {
                throw new PreexistingEntityException("Meal " + meal + "
already exists.", ex);
            }
            throw ex;
        }
    }

    public void edit(Meal meal) throws NoneexistentEntityException, Exception
{
        try {
            entityManager.getTransaction().begin();
            meal = entityManager.merge(meal);
            entityManager.getTransaction().commit();
        }
    }
}
```

```
        } catch (Exception ex) {
            String msg = ex.getLocalizedMessage();
            if (msg == null || msg.length() == 0) {
                Integer id = meal.getIdmeal();
                if (findMeal(id) == null) {
                    throw new NoneexistentEntityException("The meal with ID " +
+ id + " no longer exists.");
                }
            }
            throw ex;
        }
    }

    public void destroy(Integer id) throws NoneexistentEntityException {
        try {
            entityManager.getTransaction().begin();
            Meal meal;
            try {
                meal = entityManager.getReference(Meal.class, id);
                meal.getIdmeal();
            } catch (EntityNotFoundException enfe) {
                throw new NoneexistentEntityException("The meal with ID " +
id + " no longer exists.", enfe);
            }
            entityManager.remove(meal);
            entityManager.getTransaction().commit();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public Meal findMeal(Integer id) {
        try {
            return entityManager.find(Meal.class, id);
        } catch (Exception ex) {
            System.out.println(ex);
        }
        return null;
    }
}
```

2.9 Ερώτημα Β – Δημιουργία GUI Εφαρμογής

[Εισάγετε τον κώδικα java για τις σχετικές κλάσεις. Θα πρέπει να εισάγεται ΜΟΝΟ τον κώδικα που έχετε γράψει εσείς και όχι αυτόν που παράγεται αυτόματα προσδιορίζοντας ευκρινώς το όνομα της κλάσης και της μεθόδου. Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει τεκμηρίωση με μορφή σχολίων. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ.]

MainView.form

```
package org.mealsapp.view;

import javax.swing.*;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class MealDataView extends JFrame {
    private JPanel pnlMealData;
    private JLabel lblMealName;
    public JTextField txtMealName;
    public JButton btnSearch;
    public JTable tblSearchResults;
    private JPanel pnlMealDataControls;
    public JButton btnSaveToDB;
    public JButton btnDeleteMealData;
    public JButton btnEditMealData;
    private JButton btnClearTable;
    public DefaultTableModel searchTableModel;

    public MealDataView() {

        // Run the configuration methods
        this.configureMealDataViewForm();
        this.configureSearchResultsTableHeaders();

        // Listeners
        btnClearTable.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Clear the table
                searchTableModel.setRowCount(0);
            }
        });
    }

    private void configureMealDataViewForm() {
        // The main configuration for the MealDataViewForm

        // Set the content pane
        this.setContentPane(this.pnlMealData);

        // Set the application title
        this.setTitle("MealsApp - Δεδομένα Γευμάτων");

        // Set the size
        this.setSize(650, 400);
    }
}
```

```
// Set the location
this.setLocationRelativeTo(null);

// Set close operation
this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);

// Set single selection mode on the table
this.tblSearchResults.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

// Set Auto resize mode
this.tblSearchResults.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);

// Set cell renderer
DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
centerRenderer.setHorizontalAlignment(JLabel.CENTER);
tblSearchResults.setDefaultRenderer(Object.class, centerRenderer);
}

private void configureSearchResultsTableHeaders() {
    // Define the table columns
    TableColumn strMealColumn = new TableColumn();
    TableColumn strCategoryColumn = new TableColumn();
    TableColumn strAreaColumn = new TableColumn();
    TableColumn strInstructionsColumn = new TableColumn();

    // Set the header values
    strMealColumn.setHeaderValue("strMeal");
    strCategoryColumn.setHeaderValue("strCategory");
    strAreaColumn.setHeaderValue("strArea");
    strInstructionsColumn.setHeaderValue("strInstructions");

    // Add the columns to the table
    this.tblSearchResults.addColumn(strMealColumn);
    this.tblSearchResults.addColumn(strCategoryColumn);
    this.tblSearchResults.addColumn(strAreaColumn);
    this.tblSearchResults.addColumn(strInstructionsColumn);

    // Add the table model
    String[] columnNames = {"strMeal", "strCategory", "strArea",
"strInstructions"};
    searchTableModel = new DefaultTableModel(columnNames, 0);
    tblSearchResults.setModel(searchTableModel);
}

public void displayWindow() {

    // Set visibility
    this.setVisible(true);

    // Bring to front
    this.toFront();
}
```



```
}
```

```
package org.mealsapp.view;

import javax.swing.*.*;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;

public class MealCategoriesView extends JFrame {
    private JPanel pnlMealCategories;
    private JLabel lblMealCategory;
    public JTextField txtCategoryName;
    public JButton btnSearchCategory;
    private JTable tblCategoryResults;
    public DefaultTableModel categoryResultsTableModel;

    public MealCategoriesView() {
        // Run the configuration methods
        this.configureMealCategoriesViewForm();
        this.configureCategoryResultsTableHeaders();
    }

    private void configureMealCategoriesViewForm() {
        // The main configuration for the MealCategoriesViewForm

        // Set the content pane
        this.setContentPane(this.pnlMealCategories);

        // Set the application title
        this.setTitle("MealsApp - Λίστα γευμάτων ανά κατηγορία");

        // Set the size
        this.setSize(500, 400);

        // Set the location
        this.setLocationRelativeTo(null);

        // Set close operation
        this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);

        // Set Auto resize mode

        this.tblCategoryResults.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);

        // Set cell renderer
        DefaultTableCellRenderer centerRenderer = new
        DefaultTableCellRenderer();
    }
}
```

```
centerRenderer.setHorizontalAlignment(javax.swing.JLabel.CENTER);
tblCategoryResults.setDefaultRenderer(Object.class, centerRenderer);
}

private void configureCategoryResultsTableHeaders() {
    // Define the table columns
    TableColumn strMealColumn = new TableColumn();

    // Set the header values
    strMealColumn.setHeaderValue("strMeal");

    // Add the columns to the table
    this.tblCategoryResults.addColumn(strMealColumn);

    // Add the table model
    String[] columnNames = {"strMeal"};
    categoryResultsTableModel = new DefaultTableModel(columnNames, 0);
    tblCategoryResults.setModel(categoryResultsTableModel);
}

public void displayWindow() {
    // Set visibility
    this.setVisible(true);

    // Bring to front
    this.toFront();
}
}
```

```
package org.mealsapp.view;

import javax.swing.*.*;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class MealDataView extends JFrame {
    private JPanel pnlMealData;
    private JLabel lblMealName;
    public JTextField txtMealName;
    public JButton btnSearch;
    public JTable tblSearchResults;
    private JPanel pnlMealDataControls;
    public JButton btnSaveToDB;
    public JButton btnDeleteMealData;
    public JButton btnEditMealData;
}
```

```
private JButton btnClearTable;
public DefaultTableModel searchTableModel;

public MealDataView() {

    // Run the configuration methods
    this.configureMealDataViewForm();
    this.configureSearchResultsTableHeaders();

    // Listeners
    btnClearTable.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Clear the table
            searchTableModel.setRowCount(0);
        }
    });
}

private void configureMealDataViewForm() {
    // The main configuration for the MealDataViewForm

    // Set the content pane
    this.setContentPane(this.pnlMealData);

    // Set the application title
    this.setTitle("MealsApp - Δεδομένα Γευμάτων");

    // Set the size
    this.setSize(650, 400);

    // Set the location
    this.setLocationRelativeTo(null);

    // Set close operation
    this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);

    // Set single selection mode on the table
    this.tblSearchResults.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

    // Set Auto resize mode
    this.tblSearchResults.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);

    // Set cell renderer
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment(javax.swing.JLabel.CENTER);
    tblSearchResults.setDefaultRenderer(Object.class, centerRenderer);
}

private void configureSearchResultsTableHeaders() {
    // Define the table columns
    TableColumn strMealColumn = new TableColumn();
```

```
TableColumn strCategoryColumn = new TableColumn();
TableColumn strAreaColumn = new TableColumn();
TableColumn strInstructionsColumn = new TableColumn();

// Set the header values
strMealColumn.setHeaderValue("strMeal");
strCategoryColumn.setHeaderValue("strCategory");
strAreaColumn.setHeaderValue("strArea");
strInstructionsColumn.setHeaderValue("strInstructions");

// Add the columns to the table
this.tblSearchResults.addColumn(strMealColumn);
this.tblSearchResults.addColumn(strCategoryColumn);
this.tblSearchResults.addColumn(strAreaColumn);
this.tblSearchResults.addColumn(strInstructionsColumn);

// Add the table model
String[] columnNames = {"strMeal", "strCategory", "strArea",
"strInstructions"};
searchTableModel = new DefaultTableModel(columnNames, 0);
tblSearchResults.setModel(searchTableModel);
}

public void displayWindow() {

    // Set visibility
    this.setVisible(true);

    // Bring to front
    this.toFront();
}
}
```

Ερώτημα Γ – Παρουσίαση στατιστικών στοιχείων γευμάτων

[Εισάγετε τον κώδικα java που αναφέρεται στην παρουσίαση των στατιστικών στοιχείων. Θα πρέπει να προσδιορίζεται ευκρινώς το όνομα της κλάσης και της μεθόδου. Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει τεκμηρίωση με μορφή σχολίων. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ.]

```
package org.mealsapp.controller;

import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;
import org.mealsapp.model.MainModel;
```

```
import org.mealsapp.view.MealStatisticsView;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileOutputStream;
import java.util.List;

public class MealStatisticsController {

    private final MealStatisticsView view;
    private final MainModel model;

    public MealStatisticsController(MealStatisticsView view, MainModel model)
    {
        this.view = view;
        this.model = model;

        // Update the meal table
        this.view.btnRefreshTable.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Clear existing rows
                view.statisticsTableModel.setRowCount(0);

                // Get meals sorted by descending order of views
                List<Object[]> queryResults = model.getQueryResult();

                for (Object[] object : queryResults) {
                    // Add row
                    String[] row = new String[]{
                        object[0].toString(),
                        object[5].toString(),
                        object[2].toString(),
                        object[3].toString(),
                        object[1].toString()
                    };
                    view.statisticsTableModel.addRow(row);
                }
            }
        });

        // Create .pdf file
        this.view.btnSaveToPDF.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (view.tblMealStatistics.getRowCount() >= 1 &&
                    view.tblMealStatistics.getValueAt(0, 0) != null) {
                    try {
                        // Create document
                        Document doc = new Document(PageSize.A4.rotate());
                        PdfWriter.getInstance(doc, new
```

```
FileOutputStream("meal.pdf"));

        // Open .pdf
        doc.open();

        // Create statistics pdf table
        PdfPTable pdfTable = new
PdfPTable(view.tblMealStatistics.getColumnCount());

        // Set full width
        pdfTable.setWidthPercentage(100);

        // Set title
        for (int i = 0; i <
view.tblMealStatistics.getColumnCount(); i++) {
            PdfPCell cell = new PdfPCell(new
Paragraph(view.tblMealStatistics.getColumnName(i)));
            // Set color
            cell.setBackgroundColor(BaseColor.GREEN);
            pdfTable.addCell(cell);
        }

        // Add the data to the pdf table
        for (int rows = 0; rows <
view.tblMealStatistics.getRowCount(); rows++) {
            for (int cols = 0; cols <
view.tblMealStatistics.getColumnCount(); cols++) {

pdfTable.addCell(view.tblMealStatistics.getModel().getValueAt(rows,
cols).toString());

            }
        }

        // Add to pdf and close the doc
        doc.add(new Paragraph("R4. Προβολή στατιστικών
δεδομένων γευμάτων σε αρχείο pdf "));
        doc.add(Chunk.NEWLINE);
        doc.add(pdfTable);
        doc.close();

        // Open the .pdf file to the user
        File myFile = new File("meal.pdf");
        Desktop.getDesktop().open(myFile);
    } catch (Exception ex) {

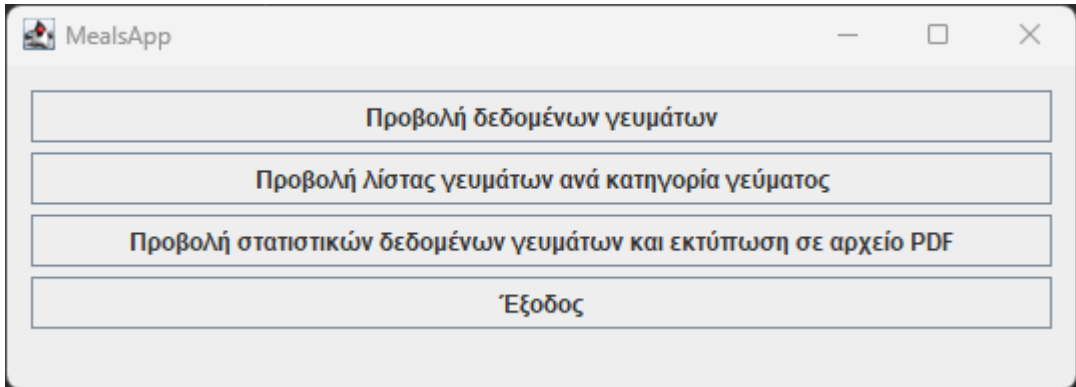
        System.out.println(ex);

    }
} else {
    JOptionPane.showMessageDialog(
        null,
        "Παρακαλώ πατήστε πρώτα το κουμπί Προβολή
στατιστικών ",
        "Ενημέρωση",
        JOptionPane.INFORMATION_MESSAGE);
}
```



2.10 Ερώτημα Δ – Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής

[Να εκτελέσετε την εφαρμογή και να εισαγάγετε σε μορφή πίνακα εικόνες (screendumps), όπου θα φαίνεται το αποτέλεσμα της εκτέλεσης. Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα, ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ. Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα, ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ. Εάν δεν έχετε υλοποιήσει μια απαίτηση γράψτε στην αντίστοιχη γραμμή ΔΕΝ ΥΛΟΠΟΙΗΘΗΚΕ. Εάν δεν έχετε υλοποιήσει ΠΛΗΡΩΣ μια απαίτηση γράψτε στην αντίστοιχη γραμμή ΔΕΝ ΥΛΟΠΟΙΗΘΗΚΕ ΠΛΗΡΩΣ, προσδιορίζοντας στη συνέχεια το βαθμό υλοποίησής της.]

Απαίτηση	Εικόνα
R1	<p>Εισαγωγή σχετικής εικόνας. Σύντομα σχόλια που να επεξηγούν τη λειτουργία.</p> 

R2

MealsApp - Λίστα γευμάτων ανά κατηγορία

Κατηγορία Γεύματος

dessert

Αναζήτηση

strMeal

Bakewell tart

Banana Pancakes

Battenberg Cake

BeaverTails

Blackberry Fool

Bread and Butter Pudding

Budino Di Ricotta

Canadian Butter Tarts

Carrot Cake

Cashew Ghoriba Biscuits

Chelsea Buns

Chinon Apple Tarts

Choc Chip Pecan Pie

Chocolate Avocado Mousse

Chocolate Caramel Crispy

Chocolate Gateau

Chocolate Raspberry Brownies

Chocolate Souffle

Christmas cake

Christmas Pudding Flapjack

Christmas Pudding Trifle

Classic Christmas pudding

Dundee cake

Eccles Cakes

Eton Mess

Honey Yogurt Cheesecake

Hot Chocolate Fudge

R3

MealsApp - Λεξιλόγιο Γεγονότων

Όνομα γεύματος

saganaki

strMeal	strCategory	strArea	strInstructions
Pancakes	Dessert	American	Put the flour, eggs, milk, 1 tbsp oil and a pinch of salt into a bowl or large jug, then whisk to a smooth batter. Set aside for 30 mins to rest if you have time, or start cooking.
Rock Cakes	Dessert	British	Preheat oven to 180C/350F/Gas 4 and line a baking tray with baking parchment. Mix the flour, sugar and baking powder in a bowl and rub in the cubed butter until the mixture resembles fine breadcrumbs.
Carrot Cake	Dessert	British	For the carrot cake, preheat the oven to 160C/325F/Gas 3. Grease and line a 25cm/10in springform cake tin with all of the ingredients for the carrot cake, except the almonds.
Dundee cake	Dessert	British	Put the almonds into a small bowl and pour over boiling water to just cover. Leave for 5 mins then drain in a sieve and leave to dry. Preheat the oven to 160C/325F/Gas 3.
Parisin Cake	Dessert	British	Heat oven to 160C/140C fan/gas 3. Grease a deep 22cm/9in square cake tin and line with baking parchment. Beat the egg and milk together with a fork. Gently stir in the flour.
Eccles Cakes	Dessert	British	To make the pastry, dice the butter and put it in the freezer to go really hard. Tip flour into the bowl of a food processor with half the butter and pulse to the texture of fine breadcrumbs.
Madeira Cake	Dessert	British	Pre-heat the oven to 150C/300F/Gas 4. Grease an 18cm/7in round cake tin, line the base with greaseproof paper and grease the paper. Cream the butter and sugar.
Christmas cake	Dessert	British	Heat oven to 160C/fan 140C/gas 3. Line the base and sides of a 20 cm round, 7.5 cm deep cake tin. Beat the butter and sugar with an electric hand mixer for 1-2 mins.
Banana Pancakes	Dessert	American	In a bowl, mash the banana with a fork until it resembles a thick purée. Stir in the eggs, baking powder and vanilla. Heat a large non-stick frying pan or pancake pan.
Battenberg Cake	Dessert	British	Heat oven to 180C/160C fan/gas 4 and line the base and sides of a 20cm square tin with baking parchment. The easiest way is to cross 2 x 20cm-long strips of paper over the tin.
New York cheesecake	Dessert	American	Position an oven shelf in the middle of the oven. Preheat the oven to fan 150C/conventional 180C/gas 4. Line the base of a 23cm springform cake tin with parchment paper.
Vegan Chocolate Cake	Vegan	American	Simply mix all dry ingredients with wet ingredients and blend altogether. Bake for 45 mins on 180 degrees. Decorate with some melted vegan chocolate.
Tunisian Orange Cake	Dessert	Tunisian	Preheat oven to 190 C / Gas 5. Grease a 23cm round springform tin. Cut off the hard bits from the top and bottom of the orange. Slice the orange and remove all the seeds.
Honey Yogurt Cheesecake	Dessert	Greek	Heat oven to 160C/140C fan/gas 3. Crush the biscuits and most of the almonds inside a plastic food bag using a rolling pin. Mix with the butter, then press into the tin.
Salted Caramel Cheesecake	Dessert	American	1) Blitz the biscuits and the pretzels in a food processor and mix the biscuits with the melted butter. Spread on the bottom of an 8' 20cm Deep Springform Tin and press down.
Peanut Butter Cheesecake	Dessert	American	Oil and line a 20cm round loose-bottomed cake tin with cling film, making it as smooth as possible. Melt the butter in a pan. Crush the biscuits by bashing them in a bag.
Provençal Omelette Cake	Vegetarian	French	Break the eggs into two bowls, five in each. Whisk lightly and season with salt and pepper. Heat the oil in a pan, add the courgettes and spring onions, then fry for 5 mins.
Polskie Naleśniki (Polish Pancakes)	Dessert	Polish	Add flour, eggs, milk, water, and salt in a large bowl then mix with a hand mixer until you have a smooth, lump-free batter. At this point, mix in the butter or the vegetable oil.
Garides Saganaki	Seafood	Greek	Place the prawns in a pot and add enough water to cover. Boil for 5 minutes. Drain, reserving the liquid, and set aside. Heat 2 tablespoons of oil in a saucepan.

Αποθήκευση σε ΒΔ

Διαγραφή δεδομένων γεύματος

Επεξεργασία δεδομένων γεύματος

R4

R4.pdf

ID	Name	Category	Notes
1	Soup Amabile Perme	Italian	Vegetarian
7	Mediterranean Pasta Salad	Italian	Seafood
6	Beef Wellington	British	Beef
5	Beef Dumpling Stew	British	Beef
4	Beef Brisket Pot Roast	American	Beef
3	Braised Beef Chili	Mexican	Beef
2	Beef Bourguignon	French	Beef

R5

3 ΚΡΙΤΙΚΟΣ ΑΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΕΡΓΟΥ

[Κάντε έναν αναλυτικό και κριτικό απολογισμό του έργου αναφέροντας

- 1) τα προβλήματα που αντιμετωπίσατε, τι θα αλλάζατε στο έργο αν είχατε τη δυνατότητα,
- 2) τις αποκλίσεις που είχατε στις επαναλήψεις σε σχέση με αυτά που ορίζονται στην εργασία,
- 3) το χρόνο που απαιτήθηκε για την κάθε δραστηριότητα με βάση τα στοιχεία που κρατήσατε και σε σχέση με το χρόνο που προϋπολογίσατε,
- 4) πως αξιολογείτε το εργαλείο συνεργασίας trello και με ποιο τρόπο σας βοήθησε στην καθημερινή σας εργασία
- 5) τους κινδύνους που πραγματώθηκαν, καθώς και πώς αποκριθήκατε σε αυτούς,
- 6) τι θα αλλάζατε σε ένα επόμενο αντίστοιχο έργο, καθώς και
- 7) ποια ήταν τα θετικά σημεία που αποκομίσατε από αυτή την εργασία].

1. Το μεγαλύτερο πρόβλημα ήταν η αντιμετώπιση της έλλειψης χρόνου. Η παρούσα εργασία όχι μόνο ήταν απαιτητική στη συγγραφή της αλλά και στην υλοποίηση της εφαρμογής της. Επομένως, προϋπέθετε να υπάρχει αρκετός ελεύθερος χρόνος για την δημιουργία της. Η κατάσταση μάλιστα επιβαρύνθηκε από το γεγονός ότι θα πρέπει οι ελεύθεροι χρόνοι του κάθε μέλους να συγχρονίζονται, καθώς η εργασία ήταν ομαδική και έπρεπε να υπάρχει συνεννόηση μεταξύ μας. Επίσης η εισαγωγή νέων μεθοδολογιών, όπως η μεθοδολογία SCRUM, η διαχείριση JSON αρχείων κλπ, που απαιτήσαν δαπάνη χρόνου για έρευνα και εκμάθηση των σχετικών εννοιών και εργαλείων, η οποία συμπίεσε το χρονικό περιθώριο για την εκπόνηση του έργου. Η Διαχείριση έργου (Project management) με συγκεκριμένη μεθοδολογία, αυτή της Scrum method ήταν δύσκολη γιατί κανένα μέλος της ομάδας δεν γνώριζε πως να διαχειριστεί ένα έργο με τη Scrum method. Παρότι η ομάδα μας προσομοίωσε μία πραγματική ομάδα ανάπτυξης σε πραγματικές συνθήκες εργασίες, δεν έπαψε να είναι μια ομάδα φοιτητών με κύριο στόχο την αφομοίωση των νέων τεχνολογιών. Κάποια προβλήματα αντιμετωπίσαμε κατά την ανάθεση εργασιών, παρόλο που στην αρχή μας φάνηκε εύκολο, όμως στην πραγματικότητα αποδείχτηκε δύσκολο να αποφασίσουμε ποιος θα κάνει τι, ειδικά για εργασίες που δε γνωρίζαμε ούτε πως γίνονται, ούτε πόσο χρόνο χρειάζονται. Αλλά και μη γνωρίζοντας τις δυνατότητες και ικανότητες ο ένας του άλλου. Ήταν η πρώτη φορά που καλεστήκαμε να υλοποιήσουμε μια πλήρη εφαρμογή, με συγκεκριμένες απαιτήσεις.

2. Είχαμε αρκετές αποκλίσεις σε σχέση με το αρχικό πλάνο διαχείρισης έργου που προτείναμε λόγω απρόοπτων καταστάσεων υγείας των μελών καθώς και λόγω παράλληλης εκπόνησης και άλλων εργασιών άλλων Θεματικών ενότητων που είχαμε αυτή την χρονιά. Παρουσιάστηκαν. Δυστυχώς, αποκλίσεις σε όλους τους τομείς της εργασίας.
3. Παρά τις δυσκολίες, η άψογη συνεργασία και η σχεδόν καθημερινή επικοινωνία, μας βοήθησαν να μην παρεκκλίνουμε από τον αρχικό σχεδιασμό μας και να ολοκληρώσουμε το έργο στις προβλεπόμενες ανθρωποημέρες που αρχικά θέσαμε ως στόχο.
4. Το Trello είναι ένα αρκετά χρήσιμο εργαλείο, με φιλικό γραφικό περιβάλλον. Παρόλα αυτά, έχει αρκετό βάθος και παρόλο που η φιλικότητά του προϋδεάζει θετικά ένα νέο χρήστη, το βάθος και η λεπτομέρειά του θα προβληματίσουν τον άπειρο χρήστη που θα βρεθεί αντιμέτωπος με τις πολλές επιλογές και τις εκτεταμένες δυνατότητες παραμετροποίησης του εργαλείου.
5. Ο βασικότερος κίνδυνος που αντιμετώπισε η ομάδα ήταν ο χρόνος και πιο συγκεκριμένα, η έλλειψη αυτού. Από νωρίς έγινε αντιληπτό ότι το συγκεκριμένο έργο θα ήταν ιδιαίτερα απαιτητικό, τόσο από τεχνικής άποψης, για τα πιο άπειρα μέλη της ομάδας, όσο και από πλευράς προγραμματισμού και τήρησης του χρονοδιαγράμματος. Ως εκ τούτου, εξ αρχής τέθηκε ο στόχος το χρονοδιάγραμμα του έργου να διαμορφωθεί με τρόπο που θα κατένειμε τις εργασίες στα 3 sprints, έτσι ώστε να υπάρχει επαρκής ευελιξία στην υλοποίηση και να μη χρειαστεί να μεταβεί κάποια εργασία σε επόμενο sprint. Κάτι τέτοιο, θα καθιστούσε ανέφικτη την πλήρη υλοποίηση του έργου και θα είχε ως αποτέλεσμα την παράδοση μιας ελλιπούς εργασίας.
6. Γνωρίζοντας πλέον τις παραπάνω δυσκολίες θα εστιάζαμε περισσότερο την προσοχή μας στην οργάνωση του έργου καθιστώντας μια ποιο αποτελεσματική επικοινωνία και ανάθεση εργασιών. Πολλές φορές χρειάστηκε να επικοινωνήσουμε αρκετές φορές για την επίλυση πολύ απλών προβλημάτων τόσο στο σχεδιαστικό τομέα όσο και στον εκτελεστικό.
7. Μια πολύ ενδιαφέρουσα εμπειρία ήταν να συνεργαστούμε σε έναν κοινό σκοπό. Όπως πολύ καλά γνωρίζουμε, οργάνωση αυτού του τύπου μπορούμε να συναντήσουμε μόνο σε πολύ μεγάλες εγκαταστάσεις και σε ποιο ώριμα περιβάλλοντα εργασίας. Η διαμόρφωση της βάσης με foreign keys, η κωδικοποίηση με επιπλέον ελέγχους συνδεσιμότητας και προσβασιμότητας, η επεξεργασία εξωτερικών αρχείων. Όπως επιπλέον ο συντονισμός εργασιών και η ανάθεση έργου σε κομμάτια SCRUM, οι μέθοδοι κατάρτισης Gantt (Project) και οι μέθοδοι αξιολόγησης Planning Poker αποτελούν τα σημαντικότερα πράγματα που είδαμε τον τελευταίο μήνα.

4 VIDEO

[Παραθέστε το link του video που δημιουργήσατε]

https://github.com/skordaschristofanis/MealsApp/blob/main/video/screen_recorder_video_2023_08_3_14_30_53.mp4

5 ΑΝΑΦΟΡΕΣ

[Παραθέστε τις βιβλιογραφικές αναφορές (βιβλία, άρθρα, σύνδεσμοι στο διαδίκτυο) που χρησιμοποιήσατε για την υλοποίηση της εργασίας.]

1. Φιτσιλής Π., Ξένος Μ., Σταμέλος Ι. Σχεδιασμός Λογισμικού- Τόμος Δ'. Προγραμματισμός Έργων Πληροφορικής-Αντικειμενοστρεφείς Μεθοδολογίες, Ελληνικό Ανοικτό Πανεπιστήμιο, 2008.
2. Εφαρμογές ΒΔ με χρήση του JPA: Java Persistence API. Ν. ΔΡΟΣΟΣ - Α. ΘΑΝΟΣ
3. «Εισαγωγή στη Γλώσσα Προγραμματισμού Java» του Εργαστηρίου Τεχνολογίας Πολυμέσων του Ε.Μ.Π.
4. Πρώτυπα σχεδίασης GUI (GUI Design Patterns) Άγγελος Θάνος
5. Γραφικά Περιβάλλοντα με Swing Α. ΘΑΝΟΣ - Ν. ΔΡΟΣΟΣ
6. Netbeans Binding Άγγελος Θάνος
7. Διαχείριση Εκδόσεων με το Subversion στο Netbeans Άγγελος Θάνος
8. Java Tutorials Creating a simple Database Application in NetBeans
1. <http://www.itvarsity.co.za/creating-a-simple-database-application-in-netbeans/>
9. Learn Programming in Java - Lesson 14: ArrayList and Collections
2. <https://www.youtube.com/watch?v=FhqdMFJbsxs>
10. Μέθοδος Planning Poker. Γιάννης Σταμέλος
11. Prepare Scrum Project
https://www.youtube.com/watch?v=ACcfYB_5iV8&list=PLebXUHgaZIX59v5YBgxRYCV7batgaCN2G
12. Εισαγωγή στις ευέλικτες μεθοδολογίες ανάπτυξης λογισμικού (Agile Software Development) SCRUM
Αλέξανδρος Χατζηγεωργίου
13. Programming in Java · Computer Science · An Interdisciplinary Approach
https://introcs.cs.princeton.edu/java/home/?utm_medium=email&utm_source=other&utm_campaign=notifications.auto.76vWA1kGEeWgdg4yGJ4GCQ
14. W3 Java Tutorial <https://www.w3schools.com/java/default.asp>
15. Φιτσιλής Π., Σταμέλος Γ. ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ 2007