Ανοικτο Πανεπιστημιο Κυπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών
«Ειδίκευση στα Πληροφοριακά Συστήματα»

Διπλωματική Εργασία

Ανάπτυξη διασύνδεσης εφαρμογής ευφυούς ηλεκτρονικού ημερολογίου με τεχνολογία AJAX (Intelligent Calendar Mobile User Interface based on AJAX technologies)

SelfPlanner

SelfPlanner

Κωτούλα Μαλαματή
Α.Μ.: 0700436

Επιβλέπων Καθηγητής
Ρεφανίδης Ιωάννης

Κύπρος, Λευκωσία
Ιούλιος 2011

Ανάπτυξη διασύνδεσης εφαρμογής ευφυούς ηλεκτρονικού ημερολογίου με τεχνολογία AJAX (Intelligent Calendar Mobile User Interface based on AJAX technologies)

SelfPlanner

SelfPlanner

Κωτούλα Μαλαματή
Α.Μ.: 0700436

Τριμελής Εξεταστική Επιτροπή

Ρεφανίδης Ιωάννης	Χατζηλάκος Θανάσης	Βασιλακόπουλος Μιχαήλ

Κύπρος, Λευκωσία
Ιούλιος 2011

Αφιερώνεται

στην Ελισάβετ, στον Δημήτρη

και στους γονείς μου που δεν βρίσκονται πια κοντά μου

αλλά θα είναι για πάντα μέσα στην καρδιά μου.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ρεφανίδη Ιωάννη που πρωτίστως με εμπιστεύτηκε και με ενθάρρυνε να επιλέξω αυτή την εργασία και στη συνέχεια με καθοδήγησε δίνοντάς μου πολύτιμες συμβουλές για την εκπόνησή της.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην κόρη μου Ελισάβετ και κυρίως στο σύζυγό μου Δημήτρη για την ανεκτίμητη βοήθεια και την αγάπη τους.

Μαλαματή Κωτούλα

Περίληψη

Οι εφαρμογές των ηλεκτρονικών ημερολογίων χρησιμοποιούνται όλο και περισσότερο τα τελευταία χρόνια, καθώς προσφέρουν αποτελεσματική διαχείριση του χρόνου των χρηστών τους. Ωστόσο σε όλες αυτές τις εφαρμογές δεν γίνεται αυτόματος προγραμματισμός των εργασιών του χρήστη. Η εφαρμογή SelfPlanner είναι μία διαδικτυακή εφαρμογή ευφυούς ηλεκτρονικού ημερολογίου που αναπτύσσεται στο Πανεπιστήμιο Μακεδονίας και ασχολείται με τον αυτόματο προγραμματισμό των εργασιών του χρήστη.

Η παρούσα διπλωματική εργασία είναι μια διαφορετική εκδοχή της εφαρμογής SelfPlanner. Χρησιμοποιείται για τον αυτόματο προγραμματισμό εργασιών του χρήστη στο Google Calendar αλλά η υλοποίησή της έγινε με τεχνολογίες AJAX, οι οποίες προσφέρουν μεγαλύτερη διαδραστικότητα και καλύτερη ποιότητα στις web εφαρμογές, κάτι το οποίο η συγκεκριμένη εργασία προσπαθεί να αποδείξει και για την εφαρμογή SelfPlanner.

Η παρούσα εφαρμογή SelfPlanner δίνει τη δυνατότητα στο χρήστη να διαχειρίζεται τις εργασίες του και με αυτόματο προγραμματισμό η εφαρμογή τοποθετεί τις εργασίες αυτές στο Google Calendar. Ο χρήστης μπορεί να καταχωρήσει εργασίες δίνοντας πολλές πληροφορίες. Μεταξύ αυτών είναι η διάρκεια της εργασίας, το subcalendar του χρήστη που θα ενημερωθεί η εργασία αυτή, η δυνατότητα η εργασία να μπορεί να εκτελεστεί τμηματικά ή όχι, ή να επαναλαμβάνεται σε τακτά χρονικά διαστήματα, οι τοποθεσίες στις οποίες μπορεί να βρίσκεται ο χρήστης για να εκτελέσει την εργασία, το χρονικό της πεδίο που αποτελείται από χρονικά διαστήματα, οι προτιμήσεις ως προς το χρονικό πεδίο όπως να προγραμματιστεί όσο το δυνατόν νωρίτερα ή αργότερα η εργασία αυτή. Μεταξύ των εργασιών μπορεί να υπάρχουν περιορισμοί διάταξης, οι οποίοι δείχνουν ποια εργασία θα εκτελεστεί πριν από κάποια άλλη. Στο Google Map ο χρήστης μπορεί να δημιουργήσει ή να διαγράψει τις τοποθεσίες που επιθυμεί, για τις οποίες η εφαρμογή υπολογίζει τη χρονική απόσταση μεταξύ τους, δηλαδή το χρόνο που χρειάζεται ο χρήστης για να μετακινηθεί μεταξύ των διαφόρων τοποθεσιών πηγαίνοντας με αυτοκίνητο. Στη συνέχεια, όταν ζητηθεί από το χρήστη, το σύστημα προσπαθεί να τοποθετήσει όσο περισσότερες εργασίες μπορεί στο Google Calendar του χρήστη, δεσμεύοντας συγκεκριμένα χρονικά διαστήματα και ακολουθώντας όλους τους περιορισμούς που του έχουν τεθεί.

Η εφαρμογή SelfPlanner με τη χρήση των τεχνολογιών AJAX δεν αντιμετώπισε προβλήματα κατά την υλοποίησή της ενώ αποδείχτηκε ότι αυξήθηκε η διαδραστικότητα και η ταχύτητα της και υπήρξε άψογη συνεργασία με το Google Calendar και με το Google Map. Συγκεκριμένα, η εφαρμογή διαχειρίστηκε τον προγραμματισμό των προσωπικών εργασιών του χρήστη πιο γρήγορα και πιο αξιόπιστα στο διαδίκτυο.

Στην παρούσα διπλωματική εργασία παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, ο αλγόριθμος SWO που αντιμετωπίζει το πρόβλημα του προγραμματισμού των προσωπικών εργασιών με χρονικούς περιορισμούς και προτιμήσεις, η βάση δεδομένων webcalendar που χρησιμοποιεί η εφαρμογή. Τέλος, παρουσιάζεται η ίδια η εφαρμογή SelfPlanner, τα συμπεράσματα και οι μελλοντικές επεκτάσεις.

Λέξεις-κλειδιά: αυτόματος προγραμματισμός εργασιών, αλγόριθμος SWO, ευφυής εφαρμογή ημερολογίου, MySql, τεχνολογίες AJAX, ασύγχρονα, διαδραστικότητα, Google Map, Google Calendar.

Abstract

The use of electronic calendar applications is considerably increased during the last years, since they help a user to organize his activities. However, they do not provide automated scheduling capabilities. SelfPlanner is an intelligent web-based calendaring application that is developed in "Macedonia" University and it has to do with the automated scheduling of a user's activities.

The present work illustrates an alternative version of SelfPlanner application. It comes as an automated calendaring tool, using Google Calendar for presentation. But the innovative part is the scheduling algorithm that is based on AJAX technologies, which provide more interactive service and better quality for web applications, something that this work tries to prove for SelfPlanner application.

Through the present SelfPlanner version, the user has the possibility to manage his activities, before the automated system places them on Google Calendar. The user formulates the scheduling problem, entering tasks and events followed by all the necessary information, like the task duration, the user's subcalendar, whether the activity is interruptible or not, whether it is periodic or not and the location the user should be in order to accomplish it. Further, the user enters the temporal domain of the activity with its allowed time intervals and the way an activity is scheduled within its domain. Moreover, task scheduling should reflect user's preferences, concerning the temporal order of accomplishing them. The user can also create or delete certain locations, using Google Map. For these locations, the application counts the meantime duration (the time a user needs to drive from one location to another). When it is asked by the user, the automated scheduling system decides where to place the more tasks possible, on user's Google Calendar, occupying specific time intervals, under the constraints that have been set.

This SelfPlanner version, based on AJAX technologies did not confront with difficulties. On the contrary, application's interactivity and speed were increased. Moreover, there was a perfect interaction with Google Calendar and Google Map. To conclude, the application accomplished to calendar user's personal tasks - events faster and more reliable on the internet.

The present work introduces first the technologies the application is based on, then the SWO algorithm that is responsible for scheduling the user's tasks and the webcalendar Data Base that the application uses. Last, the main SelfPlanner application is introduced, followed by the conclusions and the future developments.

Key-words: automated activity scheduling, SWO algorithm, intelligent calendaring application, MySql, AJAX technologies, asynchronous, interactive, Google Map, Google Calendar.

Περιεχόμενα

Κατάλογος εικόνων

Κατάλογος πινάκων

Κεφάλαιο 1ο

Εισαγωγή

Η σύγχρονη ζωή αναγκάζει τους περισσότερους ανθρώπους να οργανώνουν, σχεδόν καθημερινά ένα μεγάλο αριθμό πληροφοριών. Η ευέλικτη και δυναμική διαχείριση του χρόνου γίνεται με τον καιρό επιτακτική ανάγκη στην κοινωνία μας. Η ανάπτυξη των ηλεκτρονικών ημερολογίων τα τελευταία χρόνια έχει κάνει εύκολη την οργάνωση ενός συνεχώς αυξανόμενου όγκου πληροφοριών και έχει βοηθήσει στη διαχείριση του προσωπικού χρόνου σε ορισμένες κατηγορίες ανθρώπων. Για το λόγο αυτό έχει αναπτυχθεί ένα μεγάλο πλήθος από εφαρμογές ηλεκτρονικών ημερολογίων, οι οποίες βοηθούν το χρήστη να διαχειρίζεται καλύτερα το χρόνο και τις πληροφορίες του, καταγράφοντας και παρακολουθώντας τις υποχρεώσεις του.

Σήμερα, οι εφαρμογές ηλεκτρονικών ημερολογίων που χρησιμοποιούνται συχνά από πολλούς χρήστες είναι οι Microsoft Outlook, Yahoo Calendar, Google Calendar κλπ., οι οποίες προσφέρουν ποικίλες δυνατότητες για την αποτελεσματική διαχείριση του χρόνου. Μεταξύ αυτών είναι η διαχείριση γεγονότων και προσωπικών εργασιών του χρήστη, οι συναντήσεις με άλλα άτομα, οι προσκλήσεις, η δημοσίευση των γεγονότων, η αναζήτηση, η δημιουργία δημόσιου ή ιδιωτικού ημερολογίου.

Οι υποχρεώσεις των χρηστών μπορεί να είναι γεγονότα (events) ή εργασίες (tasks). Τα γεγονότα εκτελούνται σε συγκεκριμένο σύντομο χρονικό διάστημα ή χρονική στιγμή και σε συγκεκριμένο τόπο και μπορεί να συμμετέχουν ή όχι άλλα άτομα. Ένα γεγονός μπορεί να είναι μία συνάντηση με φίλους, μία συναυλία, μία σύσκεψη κ.α. Οι εργασίες εκτελούνται σε ένα ευρύτερο χρονικό διάστημα, συνήθως δεν συμμετέχουν άλλα άτομα, αλλά είναι προσωπικές υποχρεώσεις που χαρακτηρίζονται από μια προθεσμία. Μεταξύ άλλων μία εργασία μπορεί να είναι μία προετοιμασία παρουσίασης σε ένα συνέδριο, η συγγραφή ενός βιβλίου ή η διδασκαλία μαθημάτων.

Οι εφαρμογές των ηλεκτρονικών ημερολογίων χειρίζονται τις εργασίες με διαφορετικό τρόπο από τα γεγονότα. Οι εργασίες συνήθως δεν καταλαμβάνουν χώρο πάνω στο ημερολόγιο και έχουν λιγότερες πληροφορίες. Τα γεγονότα εμφανίζονται πάνω στα ημερολόγια και δείχνουν το χρόνο που καταλαμβάνουν με τις απαραίτητες πληροφορίες. Στη παρούσα διπλωματική, εργασίες και γεγονότα θα αναφέρονται ως εργασίες.

Ωστόσο, σε όλες τις εφαρμογές ηλεκτρονικών ημερολογίων, η διαχείριση των εργασιών του χρήστη δεν γίνεται αυτόματα από την εφαρμογή, αλλά ο ίδιος ο χρήστης τοποθετεί τις εργασίες του στο ημερολόγιο. Τα τελευταία χρόνια έχουν αναπτυχθεί σε ερευνητικό επίπεδο, διάφορα συστήματα ηλεκτρονικών ημερολογίων, για την αυτοματοποίηση του προγραμματισμού συναντήσεων - δηλαδή συμβάντων στα οποία εμπλέκονται περισσότεροι του ενός χρήστες - τα οποία όμως δεν έχουν ασχοληθεί με τον αυτόματο προγραμματισμό των προσωπικών εργασιών του χρήστη. Ασχολήθηκαν κυρίως με τον αυτόματο προγραμματισμό των συναντήσεων, γιατί θεωρούνται περισσότερο χρονοβόρες και περίπλοκες. Μερικές από τις ερευνητικές εφαρμογές που έχουν ασχοληθεί με τον προγραμματισμό των συναντήσεων, είναι οι Ptime (Personalized Time Manager) [1] και Rcal (Retsina Calendar Agent) [6].

Από την άλλη, ο αυτόματος προγραμματισμός των προσωπικών εργασιών του χρήστη, δηλαδή η τοποθέτηση των εργασιών σε ένα ημερολόγιο από την εφαρμογή, είναι δύσκολο να γίνει αποδεκτός από τους χρήστες. Όταν όμως υπάρχουν κατάλληλες επιλογές στις εργασίες που θέλει να εκτελέσει ο χρήστης, τότε του δίνεται η δυνατότητα να ελέγχει και να αποδέχεται τα αποτελέσματα που θα παράγει η εφαρμογή, η οποία τοποθετεί με αυτόματο προγραμματισμό τις προσωπικές εργασίες σε ένα ημερολόγιο.

1.1. Αντικείμενο της διπλωματικής

Πάνω σ' αυτή τη φιλοσοφία, το σύστημα SelfPlanner [4] [5] έχει αναπτυχθεί για τον αυτόματο προγραμματισμό των προσωπικών εργασιών του χρήστη. Το SelfPlanner είναι μια διαδικτυακή ευφυής εφαρμογή ημερολογίου, η οποία αναπτύσσεται στο Πανεπιστήμιο Μακεδονίας, και είναι διαθέσιμη δικτυακά στη διεύθυνση http://SelfPlanner.uom.gr. Ο πυρήνας της εφαρμογής βασίζεται στον αλγόριθμο «Squeaky Wheel Optimization», ή SWO [2]. Η τρέχουσα έκδοση του συστήματος SelfPlanner βασίζεται σε JAVA applications για τον server και Java applets για τον client, που σημαίνει ότι υπάρχουν διάφοροι περιορισμοί στις δυνατότητές της λόγω των περιορισμών των applets (π.χ. αδυναμία πρόσβασης στον τοπικό δίσκο). Επιπλέον το user interface της εφαρμογής δεν έχει πολλές δυνατότητες ανάπτυξης.

Η εφαρμογή SelfPlanner χειρίζεται τα γεγονότα και τις εργασίες με τον ίδιο τρόπο. Ο χρήστης μπορεί να καταχωρήσει τις ίδιες πληροφορίες είτε είναι γεγονός είτε

είναι εργασία και στην εφαρμογή όλα αναφέρονται ως εργασίες - tasks. Δίνει τη δυνατότητα στον χρήστη να διαχειρίζεται εργασίες με πολλές πληροφορίες. Μεταξύ αυτών είναι η διάρκεια της εργασίας, η περιοδικότητα και η τμηματοποίηση, το χρονικό της πεδίο, οι προτιμήσεις ως προς το χρονικό πεδίο, οι τοποθεσίες στις οποίες μπορεί να βρίσκεται ο χρήστης, καθώς επίσης και οι περιορισμοί διάταξης μεταξύ των εργασιών. Στο Google Map ο χρήστης διαχειρίζεται τις τοποθεσίες που θέλει και υπολογίζεται η χρονική απόσταση μεταξύ τους. Στη συνέχεια, κατά την επίλυση, ο αλγόριθμος SWO χρησιμοποιεί τις πληροφορίες, τους περιορισμούς και τις προτιμήσεις που δίνονται ώστε να προγραμματίσει κατάλληλα τις εργασίες. Το σύστημα προσπαθεί να τοποθετήσει όσο περισσότερες εργασίες μπορεί στο Google Calendar του χρήστη, ακολουθώντας όλους τους περιορισμούς που του έχουν τεθεί.

Η παρούσα εργασία αποτελεί διαφορετική εκδοχή της εφαρμογής SelfPlanner που αναπτύσσεται στο Πανεπιστήμιο Μακεδονίας, η οποία παρουσιάζει την ανάπτυξη διασύνδεσης εφαρμογής ευφυούς ηλεκτρονικού ημερολογίου με τεχνολογίες AJAX στο Google Calendar. Η χρήση των τεχνολογιών AJAX έχει οδηγήσει στην ανάπτυξη web εφαρμογών καλύτερης ποιότητας και αυξημένης διαδραστικότητας. Οι web εφαρμογές μπορούν να λαμβάνουν δεδομένα από τον server ασύγχρονα στο υπόβαθρο, χωρίς να υπάρχει αλληλεπίδραση με την εμφάνιση και τη συμπεριφορά της ιστοσελίδας.

Εφόσον η τρέχουσα εφαρμογή βασίζεται σε μία που ήδη υπάρχει, δεν έγιναν όλες οι διαδικασίες στο σχεδιασμό και την ανάλυση. Οι οθόνες δημιουργήθηκαν από την αρχή με διαφορετική τεχνολογία αλλά σχεδιάστηκαν με βάση το γραφικό περιβάλλον των οθονών της εφαρμογής SelfPlanner, που αναπτύσσεται στο Πανεπιστήμιο Μακεδονίας, ώστε ο χρήστης να μην αντιμετωπίζει δυσκολίες στην καινούργια εφαρμογή, αν είναι εξοικειωμένος με την παλιά. Πραγματοποιήθηκαν φυσικά οι αλλαγές που θεωρήθηκαν απαραίτητες.

Στόχος της εργασίας είναι ο αυτόματος προγραμματισμός των προσωπικών εργασιών του χρήστη στο Google Calendar αλλά και η χρήση τεχνολογιών AJAX στην ανάπτυξη της εφαρμογής SelfPlanner, ώστε να αποδειχτεί η αυξημένη διαδραστικότητα, η ταχύτητα και η αξιοπιστία της εφαρμογής στο διαδίκτυο καθώς και η συνεργασία χωρίς προβλήματα με το περιβάλλον ημερολογίου της Google (Google Calendar) και με την εφαρμογή διαχείρισης τοποθεσιών (Google Map).

1.2. Συνοπτική περιγραφή

Στη συνέχεια παρουσιάζεται μια σύντομη περιγραφή της διπλωματικής εργασίας:

Στο 2ο κεφάλαιο γίνεται αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για να αναπτυχθεί η εφαρμογή SelfPlanner.

Στο 3ο κεφάλαιο διατυπώνεται το πρόβλημα του προγραμματισμού των προσωπικών εργασιών και παρουσιάζεται ο αλγόριθμος SWO, ο οποίος χρησιμοποιείται για την επίλυση των προσωπικών εργασιών του χρήστη με χρονικούς περιορισμούς και προτιμήσεις.

Στο 4ο κεφάλαιο παρουσιάζεται ο σχεδιασμός της βάσης δεδομένων webcalendar που χρησιμοποιείται στην εφαρμογή SelfPlanner, με παρουσίαση διαγραμμάτων, πινάκων, σχέσεων κ.α.

Στο 5ο κεφάλαιο εξηγείται με λεπτομέρεια η λειτουργία της εφαρμογής SelfPlanner. Αρχικά γίνεται αναφορά στον τρόπο με τον οποίο χρησιμοποιήθηκαν οι τεχνολογίες για να αναπτυχθεί η εφαρμογή. Στη συνέχεια γίνεται παρουσίαση των οθονών, δίνονται λεπτομερείς επεξηγήσεις στη χρήση της εφαρμογής και παρουσιάζονται ενδεικτικά, σημαντικά μέρη του κώδικα. Τέλος, δίνονται μερικά παραδείγματα χρήσης.

Στο 6ο κεφάλαιο αναφέρονται τα συμπεράσματα της εργασίας και οι μελλοντικές επεκτάσεις.

Κεφάλαιο 2ο

Τεχνολογίες

Για την υλοποίηση της εφαρμογής SelfPlanner χρησιμοποιήθηκε ένα σύνολο τεχνολογιών, μεταξύ αυτών η τεχνολογία Ajax. Η εγκατάσταση της εφαρμογής έγινε σε ένα web server και για την αποθήκευση των στοιχείων χρησιμοποιήθηκε μία βάση δεδομένων. Στη συνέχεια παρουσιάζονται αναλυτικά οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

2.1. WAMP Server

Ο WAMP Server [21] που είναι συντομογραφία των όρων Windows, Apache, MySQL, Php, είναι ένα δικτυακό περιβάλλον ανάπτυξης στα Windows, το οποίο επιτρέπει στο χρήστη να δημιουργήσει web εφαρμογές με Apache, Php και τη βάση δεδομένων MySql. Συμπεριλαμβάνει και το phpMyAdmin για εύκολη διαχείριση των βάσεων δεδομένων.

2.2. Apache Web Server

Ο Apache Web Server [11] είναι η πιο δημοφιλής εφαρμογή Web server στον κόσμο. Είναι μια ανοικτή εφαρμογή λογισμικού, δηλαδή ο κάθε χρήστης μπορεί να κατεβάσει, να εγκαταστήσει και να χρησιμοποιήσει την εφαρμογή δωρεάν. Ο Apache Server μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα όπως Linux, Unix, Microsoft Windows κ.λ.π., υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl, Python κ.λ.π.

Ο ρόλος του Apache Server είναι να αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες (clients) όπως είναι ένας ο φυλλομετρητής (browser) ενός χρήστη και στη συνέχεια να εξυπηρετεί αυτές τις αιτήσεις "ανεβάζοντας" τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης (url), είτε μέσω ενός συνδέσμου (link). Ο τρόπος με τον οποίο ο Apache Server εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο http.

2.3. MySQL

Η MySQL [17] είναι το δημοφιλέστερο σύστημα βάσης δεδομένων ανοικτού κώδικα, το οποίο χρησιμοποιείται από ένα ευρύ φάσμα οργανώσεων για τη διαχείριση των δεδομένων τους. Χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων. Είναι διαθέσιμη δωρεάν, γρήγορη, εύκολη στη χρήση, μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα και είναι πολύ καλή στην οργάνωση και τη διαχείριση μεγάλων ποσοτήτων πληροφοριών. Η MySQL είναι κατάλληλη για την πρόσβαση των βάσεων δεδομένων στο διαδίκτυο λόγω της ταχύτητας, της ποιότητας σύνδεσης και της ασφάλειας που παρέχει.

2.4. PHP

Η PHP (Hypertext PreProcessor) [18] είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία ενός συμβατού διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Η PHP είναι μια γλώσσα συγγραφής σεναρίων στην πλευρά του διακομιστή (server-side scripting language) και έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα.

Η PHP είναι παρόμοια με την JavaScript καθώς και οι δύο μας δίνουν τη δυνατότητα να ενσωματώσουμε μικρά προγράμματα (scripts) μέσα στον κώδικα HTML μιας ιστοσελίδας (Web page). Η διαφορά ανάμεσα στην JavaScript και την PHP είναι ότι ο Web browser διερμηνεύει την JavaScript μόλις έχει φορτωθεί η ιστοσελίδα (Web page) που περιέχει το script, ενώ οι server-side scripting languages, όπως η PHP, διερμηνεύονται από τον Web server πριν ακόμα σταλεί η σελίδα στον browser.

Η PHP παρέχει πλήρη υποστήριξη για επικοινωνία με τις βάσεις δεδομένων της MySQL. Η επικοινωνία της PHP με την MySQL γίνεται με συναρτήσεις της PHP.

2.5. PhpMyAdmin

Το PhpMyAdmin [20] είναι ένα εργαλείο ανοικτού κώδικα γραμμένο σε php το οποίο διαχειρίζεται την MySQL στο δίκτυο. Μπορεί να εκτελεί διάφορες εργασίες όπως δημιουργία, τροποποίηση, διαγραφή βάσεων δεδομένων, πινάκων, πεδίων, ευρετηρίων κ.λ.π. καθώς επίσης μπορεί να διαχειρίζεται τους χρήστες και τα δικαιώματά τους και να εκτελεί εντολές Sql.

2.6. Html - Css - Xml - Javascript

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML (HyperText Markup Language), που περιλαμβάνει το κείμενο της σελίδας και τα tags της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερ-κειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πολυμέσα).

Οι σελίδες HTML είναι απλά αρχεία κειμένου σε μορφή ASCII, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο. Η HTML είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας.

Τα CSS (Cascading Style Sheets) ορίζουν τον τρόπο εμφάνισης των στοιχείων της HTML. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο ή σ' ένα εξωτερικό αρχείο CSS. Τα εξωτερικά αρχεία CSS μας δίνουν τη δυνατότητα να αλλάξουμε την εμφάνιση και τη διάταξη όλων των σελίδων με απλή επεξεργασία ενός μόνου CSS εγγράφου. Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του web επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων ταυτόχρονα.

Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων, η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML τα tags

είναι προκαθορισμένα, ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Η JavaScript είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα στις ιστοσελίδες. Ο κώδικας της JavaScript ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεστεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript. Όπως αναφέρθηκε παραπάνω, ο Web browser διερμηνεύει την JavaScript μόλις έχει φορτωθεί η ιστοσελίδα (Web page) που περιέχει το script.

2.7. JQuery

Η JQuery [15] είναι μια γρήγορη και περιεκτική JavaScript βιβλιοθήκη ανοικτού κώδικα που απλοποιεί τα έγγραφα HTML. Ο χρήστης χρησιμοποιώντας την JQuery μπορεί να κάνει πάρα πολλά πράγματα, όπως επιλογή και διαμόρφωση HTML και CSS στοιχείων, εφέ JavaScript και animations, χρήση AJAX και πληθώρα άλλων εφαρμογών, γράφοντας λιγότερο κώδικα.

Η ενσωμάτωση της βιβλιοθήκη σε ένα html αρχείο γίνεται με τον εξής κώδικα που βρίσκεται στο <head> μέρος της σελίδας:

```
<script type="text/javascript" src="jquery.js"></script>
```

Η εφαρμογή SelfPlanner χρησιμοποιεί τη βιβλιοθήκη JQuery για την εμφάνιση καρτελών-tabs και για την εμφάνιση DatePicker.

2.8. Google Maps and Google Maps API

Το Google Maps είναι μια υπηρεσία χαρτογράφησης στο Web που παρέχεται δωρεάν από την Google. Προσφέρει χάρτες δρόμου, δρομολόγια με τα πόδια, το αυτοκίνητο ή δημόσιες μεταφορές και τον εντοπισμό επιχειρήσεων σε πολλές χώρες. Το Google Maps χρησιμοποιεί Ajax για να εμφανίσει ένα συγκεκριμένο κομμάτι χάρτη που έχει επιλέξει ο χρήστης.

Το Google Maps API (Application Program Interface) [14] χρησιμοποιείται για να ενσωματώσει το Google Maps στις ιστοσελίδες με κώδικα JavaScript. Το Maps API

παρέχει μια σειρά από βοηθητικά προγράμματα για το χειρισμό των χαρτών και την προσθήκη περιεχομένου στο χάρτη μέσα από μια ποικιλία υπηρεσιών, που επιτρέπει την δημιουργία ισχυρών εφαρμογών χαρτών στον ιστοχώρο. Το Maps API είναι μια δωρεάν υπηρεσία, που διατίθενται για κάθε δικτυακό τόπο που είναι δωρεάν για τους καταναλωτές χωρίς χρέωση.

Ο χρήστης για να ενσωματώσει Google Maps στις σελίδες του, πρέπει να λάβει από την Google ένα API Key. Ο χρήστης πρέπει να έχει λογαριασμό Google για να μπορέσει να λάβει ένα Map API key. Η εγγραφή για την απόκτηση του Map API key γίνεται στη διεύθυνση http://code.google.com/apis/maps/signup.html.

Στην εφαρμογή SelfPlanner χρησιμοποιούνται διάφορες ιδιότητες και μέθοδοι του Google Maps API για την εμφάνιση του χάρτη και των markers, τη δημιουργία marker στο σημείο του χάρτη που κάνει κλικ ο χρήστης, τη διαγραφή marker που επιλέγει ο χρήστης, την εμφάνιση σύννεφου διαλόγου όταν ο χρήστης κάνει κλικ πάνω σε ένα marker και για τον υπολογισμό χρονικής απόστασης μεταξύ των markers.

2.9. Google Calendar API and Zend_Gdata

Το Google Data API είναι ένα σύνολο διαδικτυακών υπηρεσιών για read / write πρόσβαση σε εφαρμογές που φιλοξενούνται από τη Google.

Μια εφαρμογή μπορεί να χρησιμοποιήσει το Google Calendar API [13] για την δημιουργία νέων γεγονότων, την επεξεργασία ή διαγραφή γεγονότων και για την εμφάνιση γεγονότων που ταιριάζουν με συγκεκριμένα κριτήρια στο Google Calendar.

Το πακέτο Zend_Gdata περιέχει όλα όσα χρειάζονται για να υπάρχει πρόσβαση στα δεδομένα της Google, από την PHP [12]. Μερικές από τις υπηρεσίες που είναι προσβάσιμες από το Zend_Gdata είναι οι εξής: Google Calendar, Google Spreadsheets, Google Documents List, YouTube κ.λ.π.

Η κλάση Zend_Gdata_Calendar [22] χρησιμοποιείται για να προβάλει, να δημιουργήσει, να ενημερώσει και να διαγράψει τα γεγονότα στην online υπηρεσία Google Calendar.

Το Google Calendar API βασίζεται στο Atom Publishing Protocol (APP), ένα Xml βασικό format για τη διαχείριση δικτυακών πόρων. Η κυκλοφορία μεταξύ ενός πελάτη - client και των διακομιστών της Google Calendar εμφανίζεται μέσω http και

επιτρέπει τη γνησιότητα και τις συνδέσεις χωρίς έλεγχο ταυτότητας. Η δημιουργία μιας σύνδεσης με τους διακομιστές του Calendar περιλαμβάνει δύο στάδια: τη δημιουργία ενός πελάτη http και την δέσμευση μιας υπηρεσίας Zend_Gdata_Calendar γι' αυτόν τον πελάτη.

Στην εφαρμογή SelfPlanner χρησιμοποιείται η κλάση Zend_Gdata_Calendar για την εμφάνιση των SubCalendars ενός χρήστη, τη δημιουργία γεγονότος σε κάποιο SubCalendar, τη διαγραφή γεγονότων από την τρέχουσα ημερομηνία και μετά και όσων γεγονότων έχουν δημιουργηθεί από την εφαρμογή και για το διάβασμα ενός γεγονότος από κάποιο SubCalendar του χρήστη.

2.10. Η τεχνολογία AJAX

Ένα θέμα που μέχρι σήμερα διέκρινε τις εφαρμογές της επιφάνειας εργασίας (desktop applications) από τις εφαρμογές που βασιζόταν σε προγράμματα περιήγησης (browser-based applications) είναι ότι μέσα στο πρόγραμμα περιήγησης, πρέπει να γίνεται ανανέωση ολόκληρης της σελίδας για την εμφάνιση των αποτελεσμάτων. Οι Web εφαρμογές βασίζονται στη σύγχρονη επικοινωνία του περιηγητή με το διακομιστή για την εξυπηρέτηση ενός αιτήματος του χρήστη. Οποιαδήποτε ενέργεια του χρήστη σημαίνει επαναφόρτωση ολόκληρης της σελίδας και αυτό έχει σαν αποτέλεσμα να περιμένει ο χρήστης. Όλες αυτές οι απότομες εμφανίσεις δίνουν στις web εφαρμογές μια πολύ διαφορετική αίσθηση από τις desktop εφαρμογές.

Κάτι τέτοιο απαιτεί ασύγχρονη επικοινωνία της εφαρμογής με τον διακομιστή που σημαίνει ότι ο χρήστης συνεχίζει να εργάζεται μέχρι να ληφθεί η απάντηση. Η ανάπτυξη της τεχνολογίας AJAX καλύπτει ακριβώς αυτή την ανάγκη. Η ιδέα που καθοδηγεί την Ajax είναι να παίρνει την διαδραστικότητα από το internet και να την κάνει να φαίνεται τοπική, εξαλείφοντας τις ανανεώσεις των απότομα εμφανιζόμενων σελίδων.

Η τεχνολογία AJAX είναι ουσιαστικά η φόρτωση του περιεχομένου μιας ιστοσελίδας ασύγχρονα (χωρίς πλήρη επαναφόρτωση της παρούσας σελίδας), γεγονός που δηλώνει ότι η AJAX εργάζεται στο πίσω μέρος μιας εφαρμογής, ενώ ταυτόχρονα έχει τη δυνατότητα να ενημερώνει εν μέρει μια ιστοσελίδα με αποτέλεσμα την αύξηση της ταχύτητας και δίνοντας στον χρήστη την εμπειρία μιας εφαρμογής desktop.

2.10.1. Παραδοσιακή HTML

Μία παραδοσιακή HTML ιστοσελίδα εργάζεται με συγχρονισμένο παλμό, με τρόπο που ο χρήστης κάνει μία αίτηση από τον υπολογιστή του (π.χ. για να φορτώσει μια νέα σελίδα) και η οποία μεταφέρεται μέσω του πρωτοκόλλου http στον web server για μια απάντηση. Ο web server εξετάζει την αίτηση και αποφασίζει για την εκτέλεση που ικανοποιεί το αίτημα του χρήστη. Στη συνέχεια συλλέγει τα απαραίτητα στοιχεία και δημιουργεί μια νέα ιστοσελίδα που περιέχει την απάντηση του χρήστη. Τέλος, γίνεται η παράδοση της νέας σελίδας που αντικαθιστά πλήρως την προηγούμενη σελίδα και, συνεπώς, δικαιολογεί απόλυτα τον όρο "συγχρονισμένη". Δυστυχώς δεν υπάρχει τρόπος να κατέβει μέρος της σελίδας και για να ανανεωθούν δυναμικά τα δεδομένα απαιτείται να φορτωθεί ολόκληρη η σελίδα εξ αρχής. Αυτό οδηγεί σε καθυστερήσεις.

2.10.2. Ορισμός της AJAX

AJAX [8] είναι συντομογραφία των όρων Asynchronous JavaScript And XML (Ασύγχρονη JavaScript και XML). Με την AJAX, οι διαδικτυακές εφαρμογές μπορούν να στείλουν και να ανακτήσουν τα στοιχεία χωρίς να ξαναφορτωθεί ολόκληρη η ιστοσελίδα. Αυτό γίνεται με την αποστολή των αιτημάτων http στον web server, και με την τροποποίηση μόνο των μερών εκείνων της ιστοσελίδας χρησιμοποιώντας JavaScript όταν ο web server επιστρέφει στοιχεία. Η βασική διαφορά είναι ότι η AJAX δεν ζητά την δημιουργία νέας σελίδας, αλλά μόνο τα νέα δεδομένα. Αυτή η "μερική" ενημέρωση της ιστοσελίδας, είναι μια πολύ ταχύτερη ανταπόκριση και εισάγει μικρότερη καθυστέρηση στην όλη διαδικασία σε σχέση με την παραδοσιακή εξ ολοκλήρου επαναφόρτωση σελίδων. Η ιδέα είναι οτιδήποτε βρίσκεται στο Web να φαίνεται τοπικό, προσφέροντας πλούσια εμπειρία χρήστη και χαρακτηριστικά, τα οποία εμφανίζονται συνήθως μόνο σε Desktop εφαρμογές.

Ο όρος AJAX αναφέρθηκε για πρώτη φορά από τον Jesse James Garrett, τον πρόεδρο της Adaptive Path, στις 18 Φεβρουαρίου 2005, στο άρθρο "Ajax: A New Approach to Web Applications", το σημαντικότερο στα χρονικά της AJAX.

Η AJAX δεν είναι μια τεχνολογία. Είναι στην πραγματικότητα αρκετές τεχνολογίες, η καθεμία εκ των οποίων εξελίσσεται μόνη της, και οι οποίες συντίθενται με δυναμικούς νέους τρόπους. Η AJAX ενσωματώνει:

- Παρουσίαση βάσει προτύπων με χρήση XHTML και CSS.
- Δυναμική εμφάνιση και διαδραστικότητα με χρήση του Μοντέλου Αντικειμένου Εγγράφου (Document Object Model).
- Ανταλλαγή και χειρισμό δεδομένων με χρήση XML και XSLT.
- Ασύγχρονη ανάκτηση δεδομένων με χρήση XMLHttpRequest.
- Και σύνδεση όλων των παραπάνω με Javascript.

2.10.3. Λειτουργικότητα της AJAX

Στον browser γράφουμε κώδικα σε JavaScript που ανακτά δεδομένα από τον server όποτε απαιτείται. Δηλαδή επικοινωνεί με τον διακομιστή Web στο παρασκήνιο, για να μεταφέρει αυτές τις πληροφορίες, χωρίς να προκαλέσει ανανέωση σελίδας μέσα στο πρόγραμμα περιήγησης. Αυτό σημαίνει ότι ο τρόπος, με τον οποίο η Ajax μεταφέρει δεδομένα από τον διακομιστή είναι αόρατος για τον χρήστη. Η JavaScript χρησιμοποιεί ένα ειδικό αντικείμενο, που είναι ενσωματωμένο μέσα στο πρόγραμμα περιήγησης, το XMLHttpRequest object, για να στείλει παρασκηνιακά μία αίτηση στο server και να φορτώσει δεδομένα από τον server χωρίς να προκαλέσει ανανέωση σελίδας. Η JavaScript δεν χρειάζεται να σταματήσει οποιαδήποτε άλλη δουλειά όσο περιμένει τα δεδομένα να σταλούν από το server. Μπορεί να περιμένει τα δεδομένα στο background και να εκτελέσει κάποια λειτουργία όταν τα δεδομένα φτάσουν. Τα δεδομένα που στέλνονται από το server μπορούν να είναι σε μορφή XML ή ακόμα και σε απλό text. Ο JavaScript κώδικας στον browser μπορεί να διαβάσει τα δεδομένα και να τα εμφανίσει αμέσως.

Συνοπτικά η λειτουργία της Ajax: χρησιμοποιεί JavaScript στον browser και το XMLHttpRequest object για να επικοινωνήσει με το server χωρίς ανανέωση σελίδας, και χειρίζεται XML (ή άλλης μορφής text) δεδομένα που στέλνονται από το server.

2.10.4. Συστατικά της AJAX

Στη συνέχεια γίνεται ανάλυση των συστατικών της AJAX:

JavaScript

Η σύνδεση όλων των υπολοίπων συστατικών της AJAX γίνεται με Javascript.

XHTML και CSS

Η XHTML αποτελεί στην ουσία ένα υποσύνολο της HTML το οποίο συμμορφώνεται στις επιταγές του W3C ώστε να λαμβάνει υπόψη και την XML. Τα αρχεία CSS αναλαμβάνουν εξολοκλήρου την μορφοποίηση.

DOM (Document Object Model)

Το Document Object Model είναι ένα standard για την αναπαράσταση HTML, XHTML, XML. Χρησιμοποιείται από την Javascript (και άλλες scripting languages) για τη δυναμική πλοήγηση μέσα σε ένα αρχείο.

XML

Η XML είναι μια πλήρη γλώσσα mark-up. XML έχουν τα δεδομένα που στέλνονται από τον server στον browser. Όταν έχουμε συνδυασμό PHP+MySQL, ένα αρχείο PHP θα αναλάβει να φτιάξει μια δομή XML ή απλού κειμένου ώστε να την περάσει στον browser.

Αντικείμενο XMLHttpRequest

Το αντικείμενο XMLHttpRequest αποτελεί τον πυρήνα του μηχανισμού Ajax το οποίο αναπαριστά ένα αίτημα προς το διακομιστή. Είναι ενσωματωμένο μέσα στο πρόγραμμα περιήγησης, αλλά η προσπέλασή του γίνεται με διαφορετικό τρόπο, ανάλογα το πρόγραμμα περιήγησης.

Η δημιουργία του αντικειμένου γίνεται απλά, με τη βοήθεια της JavaScript, μέσω του παρακάτω κώδικα:

```
var XMLHttpRequestObject=false;
if (window.XMLHttpRequest)  // για τους περισσότερους browsers
{
  XMLHttpRequestObject = new XMLHttpRequest();
}
else if (window.ActiveXObject)          // για εκδόσεις <=6 του internet explorer
{
  XMLHttpRequestObject =  new ActiveXObject("Microsoft.XMLHTTP");
}
```

Εφόσον έχει δημιουργηθεί ένα XMLHttpRequest object, χρειάζεται να γίνει η προετοιμασία του με τη μέθοδο open. Η χρήση της μεθόδου open διαμορφώνει το αντικείμενο XMLHttpRequest – δεν συνδέει και δεν ανοίγει σύνδεση με τον διακομιστή.

Σύνταξη της μεθόδου open:

XMLHttpRequestObject.open(method, URL, asynchronous)

- method είναι κάποια http μέθοδος, συνήθως 'GET' ή 'POST'.
- Το URL αντιπροσωπεύει το που στέλνονται τα δεδομένα.
- Αν asynchronous είναι true, ο browser δεν περιμένει για απόκριση.

Καθώς εξελίσσεται η φόρτωση των δεδομένων, το πρόγραμμα περιήγησης μπορεί να κάνει άλλα πράγματα. Αυτό σημαίνει ότι η Ajax πρέπει να ειδοποιήσει το πρόγραμμα περιήγησης πότε φορτώθηκαν τα δεδομένα και είναι έτοιμα για χρήση. Το XMLHttpRequest object έχει μια ιδιότητα με όνομα onreadystatechange στην οποία αναθέτουμε μια συνάρτηση (μπορούμε να δημιουργήσουμε μια ανώνυμη συνάρτηση χρησιμοποιώντας απλώς τη λέξη κλειδί function).

Όταν λοιπόν υπάρχει μια αλλαγή στη κατάσταση των δεδομένων που φορτώνει το XMLHttpRequest object, καλείται η ανώνυμη συνάρτηση, γεγονός που σημαίνει ότι εκτελείται ο κώδικας μέσα στα άγκιστρα της συνάρτησης.

Μέσα στην ανώνυμη συνάρτηση, πρέπει να γίνει έλεγχος των δεδομένων που φορτώθηκαν. Αυτό μπορεί να γίνει με δύο ιδιότητες του XMLHttpRequest object: readyState, status.

Η ιδιότητα readyState μας δείχνει πώς εξελίσσεται η φόρτωση των δεδομένων. Όταν η τιμή του είναι 4 τότε η φόρτωση των δεδομένων ολοκληρώθηκε πλήρως.

Η ιδιότητα status περιέχει την πραγματική κατάσταση της φόρτωσης. Όταν η τιμή του είναι 200 τότε η φόρτωση των δεδομένων ολοκληρώθηκε κανονικά.

Μετά την φόρτωση των δεδομένων με επιτυχία, για να ανακτήσουμε τα δεδομένα χρησιμοποιούμε την ιδιότητα responseText για απλό κείμενο ή responseXML για XML δεδομένα.

Τέλος, η σύνδεση με τον διακομιστή εξαρτάται από την μέθοδο HTTP (GET, ή POST) και γίνεται με τη μέθοδο send του XMLHttpRequest object:

```
XMLHttpRequestObject.send(null);
Χρησιμοποιείται όταν γίνεται χρήση GET request


XMLHttpRequestObject.send(content);
Χρησιμοποιείται όταν γίνεται χρήση του POST request
```

Στη συνέχεια ακολουθεί ένα παράδειγμα, στο οποίο με τη μέθοδο Post στέλνεται το username στο πρόγραμμα data_users.php και αυτό επιστρέφει (με php-sql) το πεδίο upassword:

```
XMLHttpRequestObject.open("POST", "data_users.php");
XMLHttpRequestObject.onreadystatechange = function()
{
    if(XMLHttpRequestObject.readyState == 4 &&
    XMLHttpRequestObject.status == 200)
  {
        var upassword = XMLHttpRequestObject.responseText;
  }
}
XMLHttpRequestObject.send("username="+uname);
```

2.10.5. Πλεονεκτήματα και μειονεκτήματα της AJAX

Τα βασικά πλεονεκτήματα της Ajax [9] είναι:

- Η τεχνική Ajax χρησιμοποιεί έννοιες και τεχνολογίες με τις οποίες ένας προγραμματιστής είναι τυπικά ήδη εξοικειωμένος.
- Το αποτέλεσμα της ασύγχρονης επικοινωνίας και ανανέωσης των περιεχομένων της σελίδας, δίνει στον τελικό χρήστη την αίσθηση ότι δουλεύει σε μια εφαρμογή τύπου Desktop, με όλα τα πλεονεκτήματα που αυτό συνεπάγεται.
- Δεν χρειάζεται να ξαναφορτωθεί ολόκληρη η σελίδα. Απλά μεταφέρεται η απαραίτητη πληροφορία που ζητήθηκε και ο χρήστης έχει την αίσθηση ότι οι ενέργειες του έχουν άμεσο αποτέλεσμα.
- Σε μια σελίδα φορτώνουμε τα βασικά scripts και CSS αρχεία μία φορά και έπειτα αξιοποιούμε τις πολλαπλές συνδέσεις για να μεταφέρουμε στη σελίδα μας το περιεχόμενο που επιθυμούμε.
- Ο χρόνος αναμονής μειώνεται. Στην περίπτωση που ο χρήστης υποβάλει μια φόρμα δεν χρειάζεται να περιμένει όλη την σελίδα να ξαναφορτώσει ώστε να αποσταλούν τα δεδομένα της φόρμας. Αντί αυτού μπορεί να συνεχίσει να δουλεύει στην σελίδα, ενώ τα δεδομένα αποστέλλονται.
- Η κίνηση από και προς το server μειώνεται σημαντικά, γιατί δεν είναι απαραίτητο να σταλεί ολόκληρο το περιεχόμενο της σελίδας.
- Σε περίπτωση που παρουσιαστεί σφάλμα τότε επηρεάζεται μόνο το συγκεκριμένο τμήμα χωρίς να επηρεάζει τα υπόλοιπα και να χαθούν κρίσιμα ίσως δεδομένα.
- Χρησιμοποιεί την υπάρχουσα τεχνολογία που ήδη έχει δοκιμαστεί.

Τα βασικά μειονεκτήματα της Ajax είναι:

- Η ανάπτυξη μιας Ajax εφαρμογής μπορεί να απαιτεί περισσότερο χρόνο και κόστος. Θεωρείται πιο δύσκολη η ανάπτυξη λόγω των πολλών εμπλεκόμενων τεχνολογιών.

- Ο κώδικας επικοινωνεί με το διακομιστή και έχει άμεση πρόσβαση σ' αυτόν. Κατά συνέπεια δεν υπάρχει ασφάλεια σε επιθέσεις κακόβουλου κώδικα.
- Με την Ajax, μια σελίδα μπορεί να αλλάξει τη μορφή της χωρίς να επαναφορτωθεί. Με αυτόν τον τρόπο, οι browsers δεν μπορούν να δημιουργήσουν bookmarks. Εφόσον μια σελίδα δεν φορτώνεται ξανά, αυτό σημαίνει ότι δεν αλλάζει το url. Κατά συνέπεια δεν μπορεί να χρησιμοποιηθεί το πλήκτρο Back. Μπορούν να ξεπεραστούν αυτά τα προβλήματα με επιπλέον προγραμματισμό κάτι που αυξάνει το κόστος και το χρόνο ανάπτυξης.
- Ένας από τους κύριους λόγους που η Ajax δεν χρησιμοποιείται στο μεγαλύτερο ποσοστό των ιστοσελίδων είναι γιατί δεν μπορούν να γίνουν indexed από τις μηχανές αναζήτησης. Για να αντιμετωπιστεί αυτό το πρόβλημα μπορεί να χρησιμοποιηθεί η Ajax σε ορισμένα μέρη κατά την ανάπτυξη μιας ιστοσελίδας.
- Ένα σημαντικό πρόβλημα είναι ότι η Ajax υποστηρίζεται μόνο από browsers οι οποίοι υποστηρίζουν πλήρως JavaScript με έμφαση στο XMLHttpRequest αντικείμενο. Οπότε μπορεί να υπάρχουν επισκέπτες που να έχουν παλιότερους browsers και έτσι να μην μπορούν να δουν αυτές τις εφαρμογές.

2.10.6. Πλαίσια Ανάπτυξης Εφαρμογών Ajax (Ajax Frameworks)

Η τεχνολογία Ajax βοήθησε στη δημιουργία μιας νέας κατηγορίας πλαισίων ανάπτυξης εφαρμογών για το web αλλά και την αναβάθμιση κάποιων υπαρχόντων.

Το Ajax Framework [16] που βοηθάει στην ανάπτυξη web εφαρμογών που χρησιμοποιούν Ajax, είναι μία συλλογή από τεχνολογίες που χρησιμοποιούνται για την κατασκευή δυναμικών ιστοσελίδων. Είναι έτοιμες βιβλιοθήκες (κώδικας που έχει γραφτεί εκ των προτέρων, συχνά κώδικας javascript) που κάνει την χρήση της Ajax πολύ εύκολη και γρήγορη.

Αναφορά ενδεικτικών Frameworks:

- jQuery και jQuery UI:

  το πιο δημοφιλές εργαλείο που χρησιμοποιείται από πολλούς προγραμματιστές για τη γρήγορη ανάπτυξη των ιστοσελίδων.

- Prototype:

  JavaScript Framework που παρέχει και λειτουργικότητα AJAX.

- Yahoo! UI Library:

  συλλογή εργαλείων και βιβλιοθηκών για την ανάπτυξη πλούσιων σε λειτουργικότητα Web εφαρμογών που χρησιμοποιεί και τεχνικές Ajax.

2.10.7. Εφαρμογές Ajax

Στη συνέχεια παρουσιάζονται μερικές χαρακτηριστικές εφαρμογές που χρησιμοποιούν Ajax [10]:

- Google suggest (autocomplete application) (http://www.google.com)

  Πρόκειται για τη μηχανή αναζήτησης της Google. Όταν ο χρήστης πληκτρολογήσει ένα γράμμα στο πλαίσιο αναζήτησης τότε ένα pull down menu ανοίγει και εμφανίζει ανάλογα αποτελέσματα.

- Google Maps (http://maps.google.com)

  Η εφαρμογή Google Maps χρησιμοποιεί την Ajax ώστε να μη φορτώνεται κάθε φορά η σελίδα που εμφανίζει ένα χάρτη. Ο χάρτης είναι διασπασμένος σε μια σειρά από εικόνες και όταν ο χρήστης μετακινεί το χάρτη, οι ίδιες εικόνες χρησιμοποιούνται για να παρουσιάσουν διαφορετικά τμήματα του χάρτη.

- Google Calendar (https://www.google.com/calendar)

  Η Google Calendar χρησιμοποιεί Ajax έτσι ώστε να εμφανίζονται τα calendars του χρήστη χωρίς ανανέωση της σελίδας.

- Yahoo Search (http://demo.backbase.com/bbsearch)

  Το Yahoo Search που χρησιμοποιεί Ajax δουλεύει ως εξής: Όταν ο χρήστης κάνει κλικ στο πλήκτρο Search, τα αποτελέσματα της αναζήτησης εμφανίζονται σε ένα box και δεν κάνει ανανέωση της σελίδας.

- Google Gmail (http://www.gmail.com)

  Η εφαρμογή Gmail είναι μια ελεύθερη υπηρεσία e-mail. Όταν συνδέεται ο χρήστης, φορτώνεται το περιβάλλον εργασίας του μέσα σε ένα από τα πλαίσια

που χρησιμοποιεί η εφαρμογή. Η υπηρεσία λειτουργεί ταχύτατα και με τρόπο παραπλήσιο των εφαρμογών διαχείρισης e-mail.

- Yahoo email (https://login.yahoo.com)

  Το Yahoo mail χρησιμοποιεί την τεχνολογία Ajax και φιλοσοφία αντίστοιχη του Gmail.

- Google Docs and Spreadsheets της Google (http://doc.google.com).

Κεφάλαιο 3ο

Αλγόριθμος SWO

Η εφαρμογή SelfPlanner αναπτύχθηκε για τον αυτόματο προγραμματισμό των προσωπικών εργασιών του χρήστη. Ο αλγόριθμος «Squeaky Wheel Optimization», ή SWO, που χρησιμοποιείται από την εφαρμογή, αντιμετωπίζει το πρόβλημα του προγραμματισμού των προσωπικών εργασιών του χρήστη στο χρόνο και στο χώρο με χρονικούς περιορισμούς και προτιμήσεις. Λαμβάνοντας υπόψη ότι δεν υπάρχουν δύο εργασίες που μπορεί να συμπίπτουν χρονικά, τοποθετεί τις εργασίες σε μια σειρά για να εκτελεστούν στη συνέχεια.

3.1. Διατύπωση του προβλήματος

Για να επιλυθεί το πρόβλημα του προγραμματισμού των προσωπικών εργασιών ο αλγόριθμος SWO [5] χρησιμοποιείται τις εξής πληροφορίες:

- Ο χρόνος θεωρείται διακριτός και το ελάχιστο χρονικό διάστημα που χρησιμοποιείται είναι 30 λεπτά.
- Υπάρχει ένα σύνολο T από N εργασίες $T=\{T_1, T_2, \ldots, T_N\}$ όπου κάθε εργασία χαρακτηρίζεται από διάφορα χαρακτηριστικά:
  - Κάθε εργασία έχει διάρκεια που συμβολίζεται με dur. Όλες οι εργασίες θεωρούνται διακοπτόμενες, δηλαδή μπορούν να χωριστούν σε τμήματα και μπορούν να προγραμματιστούν ξεχωριστά.
  - Το κάθε τμήμα χαρακτηρίζεται από την ώρα έναρξης και τη διάρκεια του. Το άθροισμα των διαρκειών όλων των τμημάτων της εργασίας πρέπει να ισούται με τη συνολική διάρκειά της.
  - Για κάθε εργασία υπάρχει η μέγιστη και η ελάχιστη επιτρεπόμενη διάρκεια των τμημάτων της, smax, smin, καθώς επίσης και η ελάχιστη επιτρεπόμενη χρονική απόσταση μεταξύ κάθε ζεύγους των τμημάτων της, dmin. Ανάλογα τις τιμές των smax, smin και τη συνολική διάρκεια dur της εργασίας μπορεί να υπάρχουν περιορισμοί όπως να μην μπορεί να διασπαστεί η εργασία.
  - Κάθε εργασία έχει τον τομέα της (domain) που αποτελείται από ένα σύνολο χρονικών διαστημάτων, μέσα στο οποίο θα πρέπει να προγραμματιστούν όλα τα τμήματά της.

- Υπάρχει ένα σύνολο L από M τοποθεσίες $L=\{L_1, L_2, \ldots, L_M\}$ και ένας δισδιάστατος πίνακας Dist που έχει τις χρονικές αποστάσεις μεταξύ τους. Κάθε εργασία έχει το δικό της σύνολο τοποθεσιών, που δηλώνει εναλλακτικά μέρη που ο χρήστης θα πρέπει να βρίσκεται για να εκτελέσει κάθε τμήμα της εργασίας (ο χρήστης δεν χρειάζεται να εκτελέσει όλα τα τμήματα της εργασίας στην ίδια τοποθεσία).
- Για κάθε υποσύνολο S εργασιών του συνόλου T, μπορεί να οριστεί ένας περιορισμός-constraint μεταξύ των εργασιών που καθορίζει τους τρόπους χρονοδιαγράμματος του συνόλου τους. Οι περιορισμοί αφορούν το χρόνο και όχι τις τοποθεσίες. Ωστόσο ο ρόλος των τοποθεσιών είναι σημαντικός, δεδομένου ότι η απόφαση μία εργασία να προγραμματιστεί σε μία τοποθεσία, μπορεί να επηρεάσει τον τομέα άλλων εργασιών.
- Τέλος, επιτρέπονται οι προτιμήσεις χρόνου στο σύνολο εργασιών, που αφορούν το πότε οι εργασίες θα εκτελεστούν μέσα στο χρονικό τους πεδίο, όπως το να εκτελεστούν όσο το δυνατόν νωρίτερα ή αργότερα.

Ο αλγόριθμος SWO χρησιμοποιεί τις παραπάνω πληροφορίες που είναι ιδιότητες, περιορισμοί και προτιμήσεις των εργασιών και προγραμματίζει κατάλληλα τις εργασίες ώστε να μπορέσουν στη συνέχεια από την εφαρμογή SelfPlanner να εκτελεστούν στο Google Calendar.

3.2. Αλγόριθμος SWO

Στη συνέχεια δίνεται μια μικρή αναφορά για τον αλγόριθμο SWO που χρησιμοποιείται στην εφαρμογή SelfPlanner αλλά δεν αποτελεί μέρος της παρούσας εργασίας.

Ο αλγόριθμος SWO, που είναι γραμμένος σε C++, είναι μία γενική προσέγγιση βελτιστοποίησης που μπορεί να προσαρμοστεί σε πολλά προβλήματα ικανοποίησης περιορισμών. Ένας άπληστος αλγόριθμος χρησιμοποιείται για την κατασκευή μιας λύσης, η οποία στη συνέχεια αναλύεται για να βρεθούν τα προβληματικά σημεία. Τα αποτελέσματα της ανάλυσης χρησιμοποιούνται για τη δημιουργία νέων προτεραιοτήτων που καθορίζουν τη σειρά με την οποία ο άπληστος αλγόριθμος κατασκευάζει την επόμενη λύση. Αυτός ο κύκλος Construct/Analyze/Prioritize

(κατασκεύασε/ανάλυσε/βάλε) προτεραιοτήτων επαναλαμβάνεται μέχρι κάποιο όριο ή μέχρι να βρεθεί μια αποδεκτή λύση. Το SWO κάνει αναζητήσεις σε δύο χώρους: στις προτεραιότητες και στις λύσεις.

Ο αλγόριθμος SWO προσαρμόστηκε κατάλληλα για να επιλύσει το πρόβλημα του προγραμματισμού των προσωπικών εργασιών.

Κεφάλαιο 4ο

Βάση δεδομένων Webcalendar

Στο κεφάλαιο αυτό παρουσιάζεται ο σχεδιασμός και η υλοποίηση της βάσης δεδομένων που χρησιμοποιεί η εφαρμογή, παραθέτοντας μεταξύ άλλων τα διαγράμματα και τους σχετικούς πίνακες.

4.1. Συλλογή και ανάλυση απαιτήσεων της βάσης δεδομένων

Το πρώτο βήμα της διαδικασίας σχεδιασμού μιας βάσης δεδομένων [7] είναι η συλλογή και ανάλυση των απαιτήσεων. Κατά τη διάρκεια του βήματος αυτού, οι σχεδιαστές της βάσης δεδομένων συζητούν με τους υποψήφιους χρήστες της βάσης για να κατανοήσουν και να καταγράψουν τις απαιτήσεις τους σχετικά με τα δεδομένα. Η εφαρμογή SelfPlanner, όπως έχει αναφερθεί, βασίστηκε σε υπάρχουσα εφαρμογή στην οποία έχει γίνει συλλογή και ανάλυση των απαιτήσεων οι οποίες έχουν χρησιμοποιηθεί και στην καινούργια εφαρμογή.

Οι απαιτήσεις που χρησιμοποιήθηκαν είναι οι εξής:

- Οι χρήστες (users) που εισέρχονται στην εφαρμογή έχουν τα δικά τους αρχεία.
- Ο χρήστης έχει όνομα, password, κωδικό και password της Google, default calendar και τρέχουσα τοποθεσία.
- Υπάρχουν τοποθεσίες (locations) στο Google Map όπου η κάθε τοποθεσία έχει κωδικό, όνομα και συντεταγμένες του σημείου που βρίσκεται πάνω στο χάρτη.
- Ορίζονται χρονικές αποστάσεις (locations_duration) μεταξύ των τοποθεσιών.
- Ορίζεται κλάση-ομάδα (classonly) τοποθεσιών με στοιχεία κωδικό και όνομα κλάσης.
- Η κάθε τοποθεσία μπορεί να ανήκει σε μία κλάση-ομάδα, δηλαδή μια κλάση περιέχει τοποθεσίες (class_locations).
- Ο χρήστης διαχειρίζεται τοποθεσίες στο χάρτη και κλάσεις τοποθεσιών.
- Οι εργασίες (tasks) έχουν κωδικό, όνομα, ημερολόγιο, τοποθεσία, ημερομηνία έναρξης, ημερομηνία λήξης, διάρκεια, υπολειπόμενη διάρκεια, περιγραφή, διακοπτόμενη, περιοδική κ.α.
- Η εργασία έχει μία τοποθεσία ή μία κλάση τοποθεσιών.

- Ο χρήστης οργανώνει μία εργασία του, δηλαδή δίνει τις πληροφορίες που χρειάζεται σε μία εργασία.
- Η εργασία έχει ένα χρονικό τομέα (task_domain), δηλαδή το διάστημα ωρών που μπορεί να εκτελεστεί.
- Ο χρονικός τομέας της εργασίας έχει ανά ημερομηνία τις ώρες που περιέχει.
- Η εργασία μπορεί να έχει ημερομηνίες περιόδου (task_dates).
- Ο χρήστης διαχειρίζεται το χρονικό τομέα και τις περιόδους μιας εργασίας.
- Υπάρχουν περιορισμοί-ταξινομήσεις (constraints) μεταξύ των εργασιών όπου καθορίζονται από το χρήστη.
- Μια εργασία μπορεί να επιλυθεί στο Google Calendar και να ενημερωθεί (task_googlecalendar) με τις πληροφορίες κωδικό γεγονότος, αύξοντα αριθμός περιόδου, ημερομηνία και ώρα ενημέρωσης, διάρκεια και αν είναι ολοκληρωμένη ή όχι.
- Ο χρήστης εκτελεί εργασίες, δηλαδή επιλέγει επίλυση, και αυτές τοποθετούνται στο Google Calendar.
- Ο χρήστης ορίζει πρότυπα (template) για τον χρονικό τομέα με στοιχεία κωδικό, όνομα, κατηγορία και ώρες.

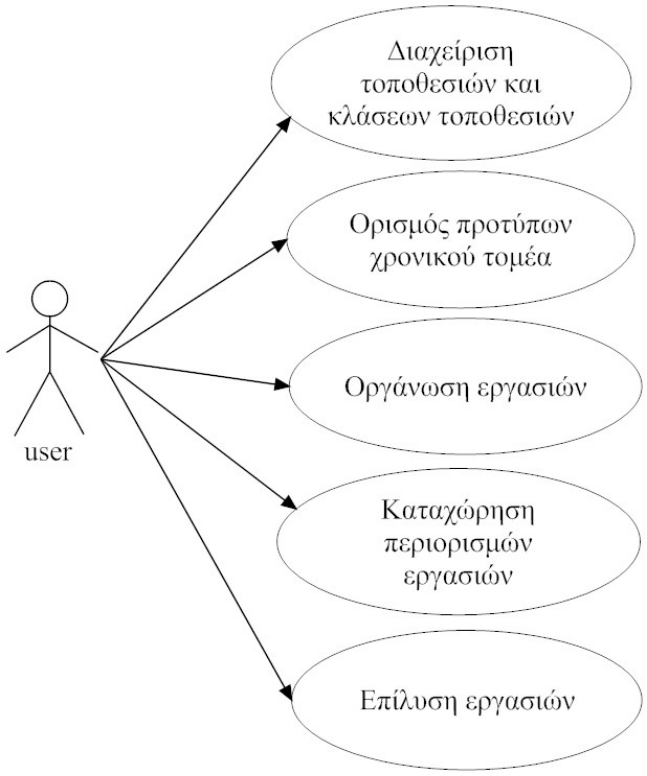
4.2. Η UML στο σχεδιασμό της βάσης δεδομένων

Η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language) UML [7] είναι μία πρότυπη γλώσσα μοντελοποίησης, η οποία χρησιμοποιείται για την απεικόνιση του σχεδιασμού και την τεκμηρίωση των συστατικών ενός αντικειμενοστραφούς συστήματος. Στη συγκεκριμένη εφαρμογή η UML χρησιμοποιείται στο σχεδιασμό βάσεων δεδομένων.

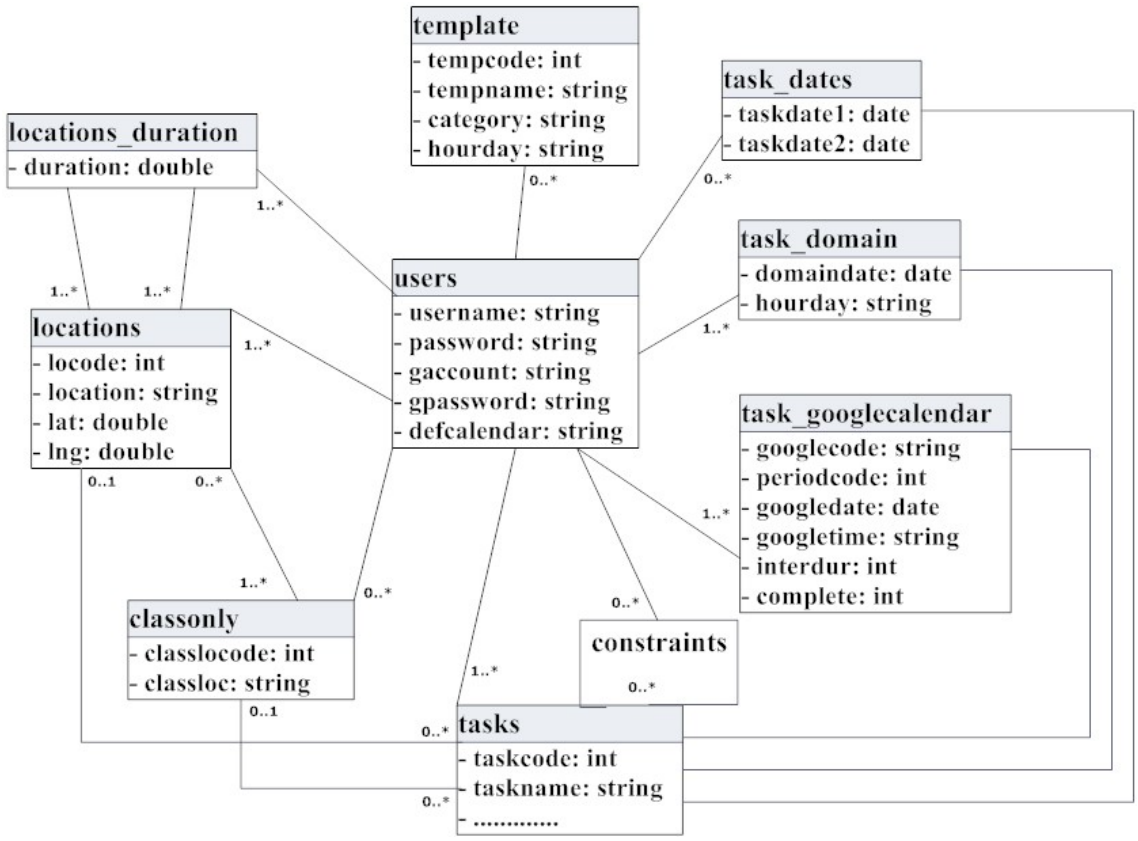
Οι απαιτήσεις που ορίστηκαν στη βάση δεδομένων μπορούν να παρασταθούν με τη χρήση διαγραμμάτων περιπτώσεων χρήσης. Τα διαγράμματα περιπτώσεων χρήσης χρησιμοποιούνται για τη μοντελοποίηση των αλληλεπιδράσεων μεταξύ χρηστών και του συστήματος (στη συγκεκριμένη περίπτωση ένα σύστημα βάσης δεδομένων). Όταν οι απαιτήσεις έχουν συλλεχθεί και έχουν οριστεί με διαγράμματα περιπτώσεων χρήσης

στη συνέχεια σχεδιάζονται τα διαγράμματα κλάσεων τα οποία δίνουν ένα δομικό ορισμό των σχημάτων των βάσεων δεδομένων με τη μορφή κλάσεων και συσχετίσεων μεταξύ τους, όπου οι κλάσεις αντιστοιχούν στις οντότητες της βάσης δεδομένων. Το διάγραμμα κλάσεων που ακολουθεί είναι ένα απλό διάγραμμα που δείχνει τα γνωρίσματα της κάθε κλάσης.

Με τη χρήση των διαγραμμάτων της UML που παρουσιάζονται στη συνέχεια έχουμε μια διαφορετική προσέγγιση στο σχεδιασμό της βάσης δεδομένων της εφαρμογής.



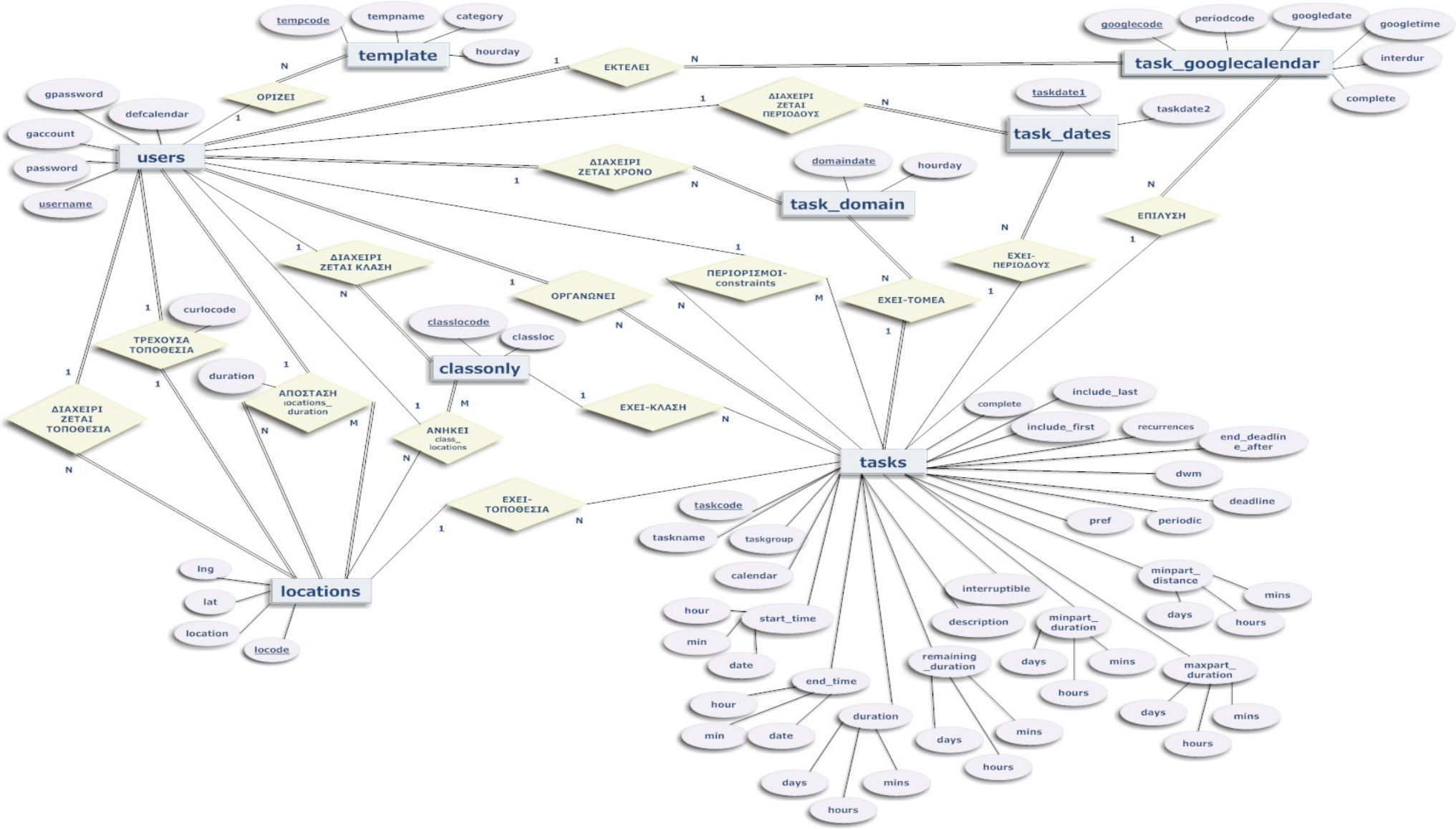
Εικόνα 1: Διάγραμμα περιπτώσεων χρήσης



Εικόνα 2: Διάγραμμα κλάσεων

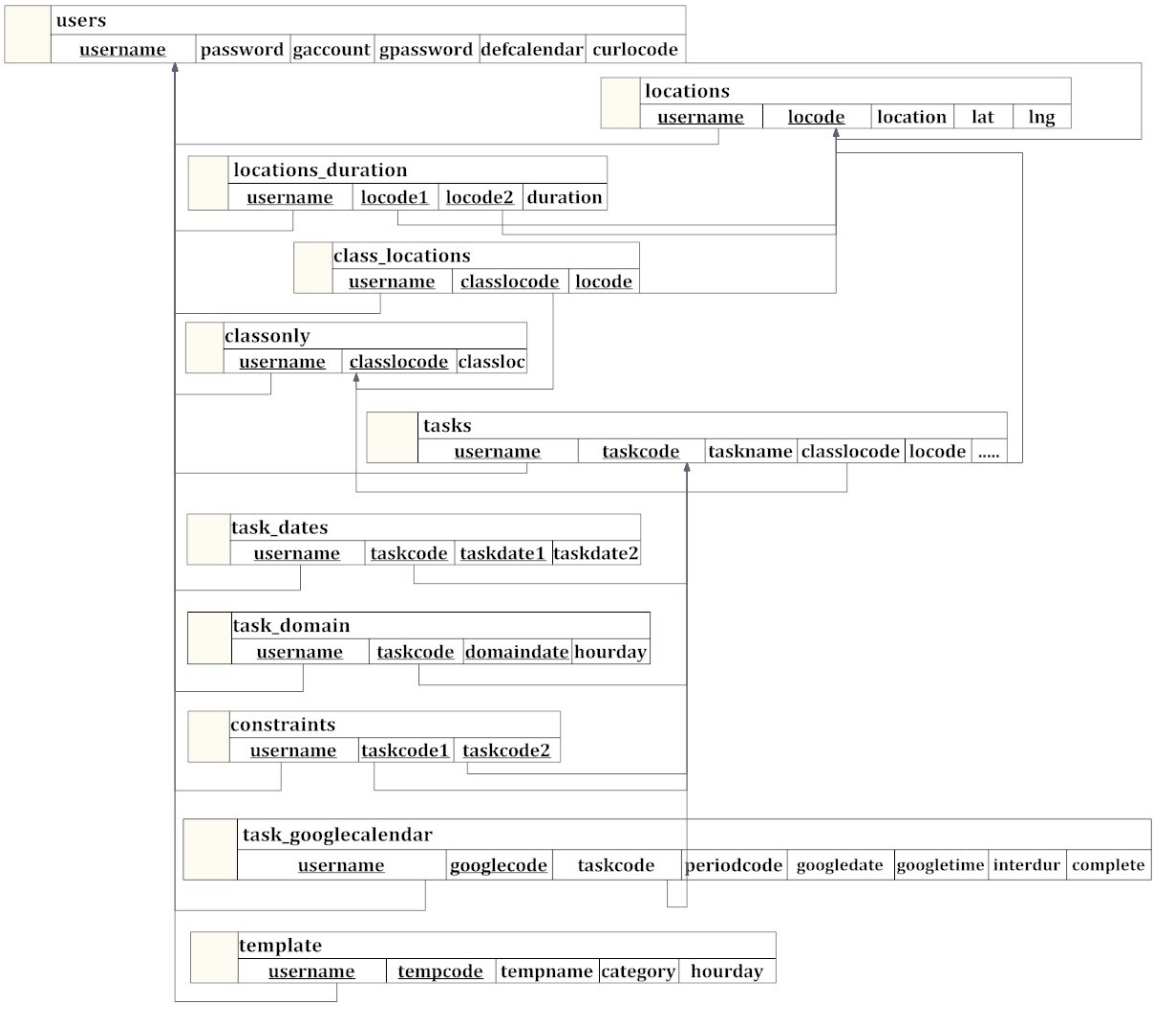
4.3. Εννοιολογικός και Λογικός σχεδιασμός βάσης δεδομένων

Το επόμενο βήμα μετά τη συλλογή και ανάλυση των απαιτήσεων, είναι η δημιουργία ενός εννοιολογικού σχήματος για τη βάση δεδομένων. Το εννοιολογικό σχήμα είναι μια περιεκτική περιγραφή των απαιτήσεων των χρηστών σχετικά με τα δεδομένα. Για τον εννοιολογικό σχεδιασμό της βάσης δεδομένων χρησιμοποιήθηκε το μοντέλο Οντοτήτων-Συσχετίσεων (ΟΣ) (Entity-Relationship (ER)), το οποίο παρουσιάζεται στη συνέχεια.



Εικόνα 3: Μοντέλο Οντοτήτων-Συσχετίσεων

Με βάση τη θεωρία μετάβασης από το μοντέλο ΔΟΣ στο σχεσιακό μοντέλο, στους πίνακες και στη διαδικασία απορρόφησης προκύπτουν οι πίνακες της βάσης δεδομένων. Από το διάγραμμα Οντοτήτων-Συσχετίσεων προκύπτουν πίνακες από τις οντότητες (users, classonly, locations, tasks, task_domain, task_dates, task_googlecalendar, template) και από τις συσχετίσεις N-M (locations_duration, class_locations, constraints), ενώ από τις συσχετίσεις 1-N οι οποίες απορροφούνται δεν προκύπτουν πίνακες. Στη συνέχεια παρουσιάζεται το σχεσιακό μοντέλο της βάσης δεδομένων.

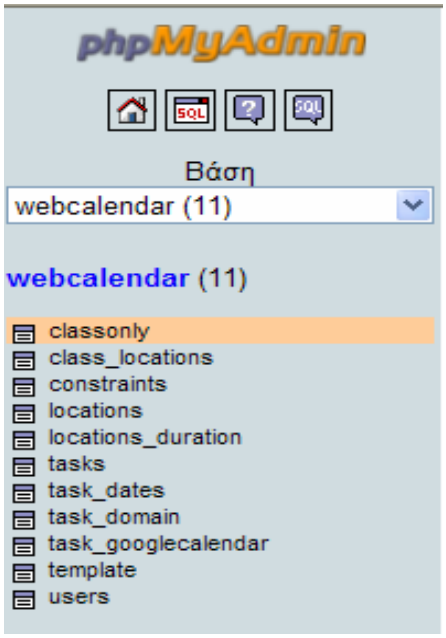


Εικόνα 4: Σχεσιακό Μοντέλο

4.4. Υλοποίηση της βάσης δεδομένων

Το επόμενο βήμα στο σχεδιασμό βάσεων δεδομένων είναι η ίδια η υλοποίηση της βάσης δεδομένων, βασιζόμενη στο σχεσιακό μοντέλο και χρησιμοποιώντας ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων.

Η εφαρμογή SelfPlanner χρησιμοποιεί τη βάση δεδομένων webcalendar (Εικόνα 5) η οποία αποτελείται από 11 πίνακες με τις κατάλληλες σχέσεις. Η διαχείριση της βάσης δεδομένων γίνεται στο περιβάλλον του PhpMyAdmin. Η βάση δεδομένων webcalendar έχει πληροφορίες για τους χρήστες – users που εισέρχονται στην εφαρμογή SelfPlanner, για τις εργασίες – tasks του χρήστη που θέλει να εκτελέσει και να ενημερώσει στο Google Calendar, για τις τοποθεσίες – locations των εργασιών καθώς επίσης και άλλες σημαντικές πληροφορίες που αναλύονται στη συνέχεια. Όταν ένας χρήστης εισέρχεται στην εφαρμογή διαχειρίζεται τα δικά του αρχεία. Για το λόγο αυτό, όλοι οι πίνακες έχουν στο πρωτεύον κλειδί τους το όνομα του χρήστη. Διευκρινίζεται ότι τα Google Subcalendars του χρήστη δεν αποθηκεύονται σε κάποιον πίνακα στη βάση δεδομένων αλλά κάθε φορά που χρειάζονται, διαβάζονται από το Google Calendar. Αυτό γίνεται έτσι ώστε οποιαδήποτε αλλαγή γίνει στα Subcalendars μέσα από το Google Calendar, να φαίνεται στην εφαρμογή SelfPlanner.



Εικόνα 5: Πίνακες της βάσης δεδομένων webcalendar

4.5. Περιγραφή πινάκων της βάσης δεδομένων

4.5.1. Πίνακας users

Στον πίνακα users αποθηκεύονται οι πληροφορίες για τους χρήστες που εισέρχονται στην εφαρμογή. Ένας χρήστης εισέρχεται στην εφαρμογή με το login – όνομα χρήστη και το password. Με την είσοδο είναι γνωστά τα στοιχεία του λογαριασμού της Google με τα οποία γίνεται η σύνδεση της εφαρμογής με το Google Calendar, το εξ ορισμού calendar του χρήστη και η τρέχουσα τοποθεσία που βρίσκεται ο χρήστης.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**password**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**gaccount**	varchar(30)	utf8_general_ci		Ναι	*NULL*
☐	**gpassword**	varchar(30)	utf8_general_ci		Ναι	*NULL*
☐	**defcalendar**	varchar(50)	utf8_general_ci		Ναι	*NULL*
☐	**curlocode**	int(11)			Ναι	*NULL*

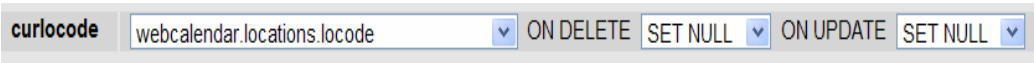
Εικόνα 6: πίνακας users

Πίνακας 1: πεδία του πίνακα users

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη - login.
password	Το password του χρήστη.
gaccount	Το email του λογαριασμού της Google, δηλαδή ο κωδικός εισόδου του χρήστη στο Google Calendar.
gpassword	Το password του λογαριασμού της Google για την είσοδο του χρήστη στο Google Calendar.
defcalendar	Το εξ ορισμού calendar του χρήστη που είναι ένα από τα Google Subcalendars.
curlocode	Η τρέχουσα τοποθεσία που βρίσκεται ο χρήστης.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα users είναι τα εξής:

- username: το πρωτεύον κλειδί – primary key.
- curlocode: ξένο κλειδί – foreign key. Το πεδίο curlocode του πίνακα users (child table) συνδέεται με το πεδίο locode (primary key) του πίνακα locations (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του locode από τον πίνακα locations διαγραφεί ή αλλάξει τότε το curlocode του πίνακα users θα πάρει τιμή null.

4.5.2. Πίνακας locations

Στον πίνακα locations αποθηκεύονται οι πληροφορίες για τις τοποθεσίες του χρήστη στο Google Map. Η κάθε τοποθεσία έχει τις συντεταγμένες του σημείου που βρίσκεται πάνω στο χάρτη. Υπάρχει η τοποθεσία με κωδικό 0 και περιγραφή ANYWHERE που εισάγεται αυτόματα και δεν υπάρχει πάνω στο χάρτη.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	username	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	locode	int(11)			Όχι	*Κανένα*
☐	location	varchar(50)	utf8_general_ci		Όχι	*Κανένα*
☐	lat	double			Όχι	*Κανένα*
☐	lng	double			Όχι	*Κανένα*

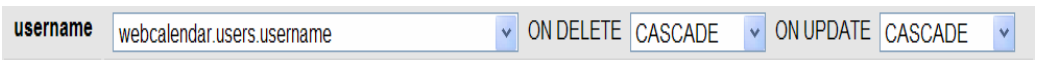
Εικόνα 7: πίνακας locations

Πίνακας 2: πεδία του πίνακα locations

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
locode	Ο κωδικός της τοποθεσίας.
location	Η περιγραφή της τοποθεσίας.
lat	Το γεωγραφικό πλάτος του σημείου που βρίσκεται η τοποθεσία πάνω στο Google Map.
lng	Το γεωγραφικό μήκος του σημείου που βρίσκεται η τοποθεσία πάνω στο Google Map.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα locations είναι τα εξής:

- username + locode: το πρωτεύον κλειδί – primary key.
- username + location: μοναδικό κλειδί – unique key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα locations (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα locations ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα locations.

4.5.3. Πίνακας locations_duration

Στον πίνακα locations_duration αποθηκεύονται οι πληροφορίες για τις χρονικές αποστάσεις μεταξύ των τοποθεσιών του χρήστη στο Google Map. Μεταξύ δύο τοποθεσιών υπολογίζεται ο χρόνος σε λεπτά, που χρειάζεται για να πάει ο χρήστης από τη μία τοποθεσία στην άλλη με αυτοκίνητο.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	username	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	locode1	int(11)			Όχι	*Κανένα*
☐	locode2	int(11)			Όχι	*Κανένα*
☐	duration	double			Όχι	*Κανένα*

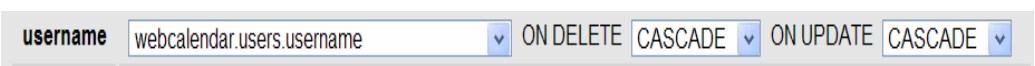
Εικόνα 8: πίνακας locations_duration

Πίνακας 3: πεδία του πίνακα locations_duration

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
locode1	Ο κωδικός της 1ης τοποθεσίας.
locode2	Ο κωδικός της 2ης τοποθεσίας.
duration	Ο χρόνος σε λεπτά που χρειάζεται ο χρήστης για να πάει από τη μία τοποθεσία στην άλλη στο Google Map με αυτοκίνητο.

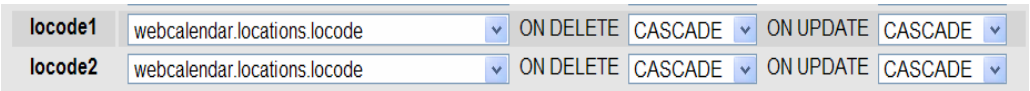
Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα locations_duration είναι τα εξής:

- username + locode1 + locode2: το πρωτεύον κλειδί – primary key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα locations_duration (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα locations_duration ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα locations_duration.

- locode1, locode2: ξένα κλειδιά – foreign keys. Το πεδίο locode1 ή locode2 του πίνακα locations_duration (child table) συνδέεται με το πεδίο locode (primary key) του πίνακα locations (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του locode από τον πίνακα locations διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα locations_duration ενώ αν η τιμή του locode από τον πίνακα locations αλλάξει τότε αλλάζει και η τιμή του locode1 ή locode2 στις αντίστοιχες εγγραφές από τον πίνακα locations_duration.

4.5.4. Πίνακας classonly

Στον πίνακα classonly αποθηκεύονται οι πληροφορίες για τις κλάσεις – ομάδες των τοποθεσιών του χρήστη.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**classlocode**	int(11)			Όχι	*Κανένα*
☐	**classloc**	varchar(50)	utf8_general_ci		Όχι	*Κανένα*

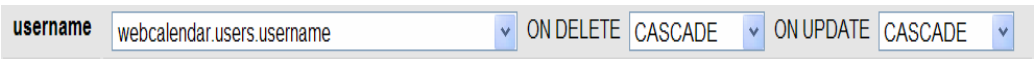
Εικόνα 9: πίνακας classonly

Πίνακας 4: πεδία του πίνακα classonly

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
classlocode	Ο κωδικός της κλάσης τοποθεσιών.
classloc	Η περιγραφή της κλάσης.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα classonly είναι τα εξής:

- username + classlocode: το πρωτεύον κλειδί – primary key.
- username + classloc: μοναδικό κλειδί – unique key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα classonly (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα classonly ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα classonly.

4.5.5. Πίνακας class_locations

Στον πίνακα class_locations αποθηκεύονται οι πληροφορίες για τις κλάσεις με τις τοποθεσίες που περιέχουν. Η κάθε κλάση είναι μία ομάδα τοποθεσιών.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**classlocode**	int(11)			Όχι	*Κανένα*
☐	**locode**	int(11)			Όχι	*Κανένα*

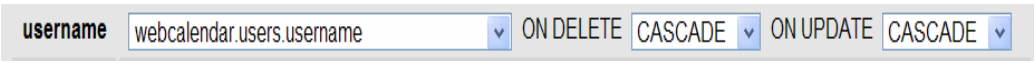
Εικόνα 10: πίνακας class_locations

Πίνακας 5: πεδία του πίνακα class_locations

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
classlocode	Ο κωδικός της κλάσης τοποθεσιών.
locode	Ο κωδικός της τοποθεσίας.

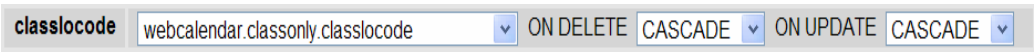
Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα class_locations είναι τα εξής:

- username + classlocode + locode: το πρωτεύον κλειδί – primary key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα class_locations (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



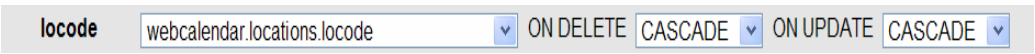
  Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα class_locations ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα class_locations.

- classlocode: ξένο κλειδί – foreign key. Το πεδίο classlocode του πίνακα class_locations (child table) συνδέεται με το πεδίο classlocode (primary key) του πίνακα classonly (parent table) ως εξής:



  Αυτό σημαίνει ότι αν η τιμή του classlocode από τον πίνακα classonly διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα class_locations ενώ αν η τιμή του classlocode από τον πίνακα classonly αλλάξει τότε αλλάζει και η τιμή του classlocode στις αντίστοιχες εγγραφές από τον πίνακα class_locations.

- locode: ξένο κλειδί – foreign key. Το πεδίο locode του πίνακα class_locations (child table) συνδέεται με το πεδίο locode (primary key) του πίνακα locations (parent table) ως εξής:



  Αυτό σημαίνει ότι αν η τιμή του locode από τον πίνακα locations διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα class_locations ενώ αν η τιμή του locode από τον πίνακα locations αλλάξει τότε αλλάζει και η τιμή του locode στις αντίστοιχες εγγραφές από τον πίνακα class_locations.

4.5.6. Πίνακας tasks

Στον πίνακα tasks αποθηκεύονται οι πληροφορίες για τις εργασίες του χρήστη.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	username	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	taskcode	int(11)			Όχι	*Κανένα*
☐	taskname	varchar(100)	utf8_general_ci		Όχι	*Κανένα*
☐	taskgroup	varchar(100)	utf8_general_ci		Ναι	*NULL*
☐	calendar	varchar(100)	utf8_general_ci		Ναι	*NULL*
☐	classlocode	int(11)			Ναι	*NULL*
☐	locode	int(11)			Ναι	*NULL*
☐	start_time_hour	tinyint(4)			Ναι	*NULL*
☐	start_time_min	tinyint(4)			Ναι	*NULL*
☐	start_time_date	date			Ναι	*NULL*
☐	end_time_hour	tinyint(4)			Ναι	*NULL*
☐	end_time_min	tinyint(4)			Ναι	*NULL*
☐	end_time_date	date			Ναι	*NULL*
☐	duration_days	tinyint(4)			Ναι	*NULL*
☐	duration_hours	tinyint(4)			Ναι	*NULL*
☐	duration_mins	tinyint(4)			Ναι	*NULL*
☐	remaining_duration_days	tinyint(4)			Ναι	*NULL*
☐	remaining_duration_hours	tinyint(4)			Ναι	*NULL*
☐	remaining_duration_mins	tinyint(4)			Ναι	*NULL*
☐	description	text	utf8_general_ci		Ναι	*NULL*
☐	interruptible	tinyint(4)			Ναι	*NULL*
☐	minpart_duration_days	tinyint(4)			Ναι	*NULL*
☐	minpart_duration_hours	tinyint(4)			Ναι	*NULL*
☐	minpart_duration_mins	tinyint(4)			Ναι	*NULL*
☐	maxpart_duration_days	tinyint(4)			Ναι	*NULL*
☐	maxpart_duration_hours	tinyint(4)			Ναι	*NULL*
☐	maxpart_duration_mins	tinyint(4)			Ναι	*NULL*
☐	minpart_distance_days	tinyint(4)			Ναι	*NULL*
☐	minpart_distance_hours	tinyint(4)			Ναι	*NULL*
☐	minpart_distance_mins	tinyint(4)			Ναι	*NULL*
☐	pref	tinyint(4)			Ναι	*NULL*
☐	periodic	tinyint(4)			Ναι	*NULL*
☐	deadline	date			Ναι	*NULL*
☐	dwm	varchar(1)	utf8_general_ci		Ναι	*NULL*
☐	end_deadline_after	varchar(1)	utf8_general_ci		Ναι	d
☐	recurrences	tinyint(4)			Ναι	*NULL*
☐	include_first	tinyint(4)			Ναι	*NULL*
☐	include_last	tinyint(4)			Ναι	*NULL*
☐	complete	tinyint(1)			Όχι	*Κανένα*

Εικόνα 11: πίνακας tasks

Πίνακας 6: πεδία του πίνακα tasks

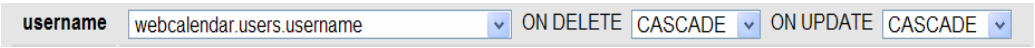
Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
taskcode	Ο κωδικός του task.
taskname	Το όνομα του task.
taskgroup	Η ομάδα του task.
calendar	Το Google subcalendar του task.
classlocode	Η κλάση τοποθεσιών του task.
locode	Η τοποθεσία του task.
start_time_hour	Η ώρα που ξεκινάει το task.
start_time_min	Τα λεπτά που ξεκινάει το task.
start_time_date	Η ημερομηνία έναρξης του task.
end_time_hour	Η ώρα που τελειώνει το task.
end_time_min	Τα λεπτά που τελειώνει το task.
end_time_date	Η ημερομηνία που τελειώνει το task.
duration_days	Η διάρκεια σε ημέρες του task.
duration_hours	Η διάρκεια σε ώρες του task.
duration_mins	Η διάρκεια σε λεπτά του task.
remaining_duration_days	Η υπολειπόμενη διάρκεια σε ημέρες του task.
remaining_duration_hours	Η υπολειπόμενη διάρκεια σε ώρες του task.
remaining_duration_mins	Η υπολειπόμενη διάρκεια σε λεπτά του task.
description	Η περιγραφή του task.
interruptible	Αν είναι διακοπτόμενο ή όχι το task.
minpart_duration_days	Η ελάχιστη διάρκεια σε ημέρες ενός τμήματος διακοπτόμενου task.

minpart_duration_hours	Η ελάχιστη διάρκεια σε ώρες ενός τμήματος διακοπτόμενου task.
minpart_duration_mins	Η ελάχιστη διάρκεια σε λεπτά ενός τμήματος διακοπτόμενου task.
maxpart_duration_days	Η μέγιστη διάρκεια σε ημέρες ενός τμήματος διακοπτόμενου task.
maxpart_duration_hours	Η μέγιστη διάρκεια σε ώρες ενός τμήματος διακοπτόμενου task.
maxpart_duration_mins	Η μέγιστη διάρκεια σε λεπτά ενός τμήματος διακοπτόμενου task.
minpart_distance_days	Η ελάχιστη χρονική απόσταση σε ημέρες μεταξύ των τμημάτων ενός διακοπτόμενου task.
minpart_distance_hours	Η ελάχιστη χρονική απόσταση σε ώρες μεταξύ των τμημάτων ενός διακοπτόμενου task.
minpart_distance_mins	Η ελάχιστη χρονική απόσταση σε λεπτά μεταξύ των τμημάτων ενός διακοπτόμενου task.
pref	Η προτίμηση του χρήστη (preference type) για το πότε θέλει να εκτελεστεί το task. 0: No preference 1: As later as possible -1: As earlier as possible
periodic	Αν είναι περιοδικό ή όχι το task.
deadline	Η ημερομηνία λήξης του task.
dwm	Αν το περιοδικό task είναι ημερήσιο (daily), εβδομαδιαίο (weekly), μηνιαίο (monthly).
end_deadline_after	Αν το τέλος των περιόδων ενός task είναι μέχρι το deadline ή μέχρι την ημερομηνία που υπολογίζεται με συγκεκριμένο πλήθος περιόδων.

recurrences	Το πλήθος περιόδων του task.
include_first	Αν περιλαμβάνεται ή όχι η πρώτη περίοδος του task.
include_last	Αν περιλαμβάνεται ή όχι η τελευταία περίοδος του task.
complete	Αν είναι ολοκληρωμένο ή όχι το task.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα tasks είναι τα εξής:

- username + taskcode: το πρωτεύον κλειδί – primary key.
- username + taskname: μοναδικό κλειδί – unique key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα tasks (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:

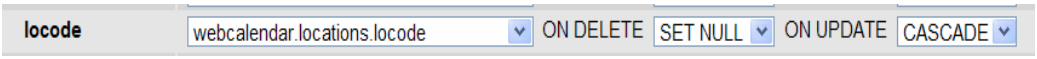


  Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα tasks ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα tasks.

- classlocode: ξένο κλειδί – foreign key. Το πεδίο classlocode του πίνακα tasks (child table) συνδέεται με το πεδίο classlocode (primary key) του πίνακα classonly (parent table) ως εξής:

  Αυτό σημαίνει ότι αν η τιμή του classlocode από τον πίνακα classonly διαγραφεί τότε η τιμή του classlocode γίνεται null στις αντίστοιχες εγγραφές από τον πίνακα tasks ενώ αν η τιμή του classlocode από τον πίνακα classonly αλλάξει τότε αλλάζει και η τιμή του classlocode στις αντίστοιχες εγγραφές από τον πίνακα tasks.

- locode: ξένο κλειδί – foreign key. Το πεδίο locode του πίνακα tasks (child table) συνδέεται με το πεδίο locode (primary key) του πίνακα locations (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του locode από τον πίνακα locations διαγραφεί τότε η τιμή του locode γίνεται null στις αντίστοιχες εγγραφές από τον πίνακα tasks ενώ αν η τιμή του locode από τον πίνακα locations αλλάξει τότε αλλάζει και η τιμή του locode στις αντίστοιχες εγγραφές από τον πίνακα tasks.

4.5.7. Πίνακας task_domain

Στον πίνακα task_domain αποθηκεύονται οι πληροφορίες για το domain ενός task ανά ημερομηνία. Το domain είναι τα ημίωρα ή οι μέρες του μήνα που μπορεί ο χρήστης να εκτελέσει το task.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**taskcode**	int(11)			Όχι	*Κανένα*
☐	**domaindate**	date			Όχι	*Κανένα*
☐	**hourday**	text	utf8_general_ci		Όχι	*Κανένα*

Εικόνα 12: πίνακας task_domain

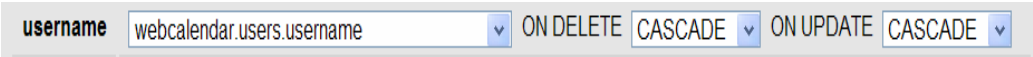
Πίνακας 7: πεδία του πίνακα task_domain

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
taskcode	Ο κωδικός του task.
domaindate	Η ημερομηνία του domain.
hourday	Πεδίο κειμένου που περιέχει τα ημίωρα ή τις ημέρες μήνα που μπορεί να εκτελεστεί το task.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα task_domain είναι τα εξής:

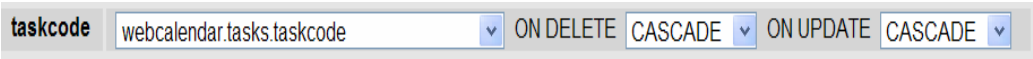
- username + taskcode + domaindate: το πρωτεύον κλειδί – primary key.

- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα task_domain (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_domain ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα task_domain.

- taskcode: ξένο κλειδί – foreign key. Το πεδίο taskcode του πίνακα task_domain (child table) συνδέεται με το πεδίο taskcode (primary key) του πίνακα tasks (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του taskcode από τον πίνακα tasks διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_domain ενώ αν η τιμή του taskcode από τον πίνακα tasks αλλάξει τότε αλλάζει και η τιμή του taskcode στις αντίστοιχες εγγραφές από τον πίνακα task_domain.

4.5.8. Πίνακας task_dates

Στον πίνακα task_dates αποθηκεύονται οι πληροφορίες για τις ημερομηνίες περιόδων ενός task. Ο πίνακας task_dates ενημερώνεται όταν το task είναι περιοδικό.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**taskcode**	int(11)			Όχι	*Κανένα*
☐	**taskdate1**	date			Όχι	*Κανένα*
☐	**taskdate2**	date			Όχι	*Κανένα*

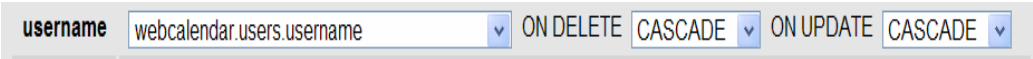
Εικόνα 13: πίνακας task_dates

Πίνακας 8: πεδία του πίνακα task_dates

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
taskcode	Ο κωδικός του task.
taskdate1	Η ημερομηνία έναρξης της περιόδου.
taskdate2	Η ημερομηνία λήξης της περιόδου.

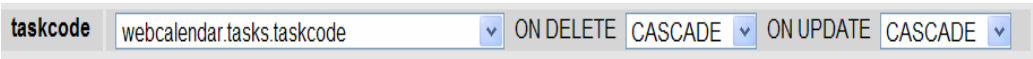
Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα task_dates είναι τα εξής:

- username + taskcode + taskdate1: το πρωτεύον κλειδί – primary key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα task_dates (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



  Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_dates ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα task_dates.

- taskcode: ξένο κλειδί – foreign key. Το πεδίο taskcode του πίνακα task_dates (child table) συνδέεται με το πεδίο taskcode (primary key) του πίνακα tasks (parent table) ως εξής:



  Αυτό σημαίνει ότι αν η τιμή του taskcode από τον πίνακα tasks διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_dates ενώ αν η τιμή του taskcode από τον πίνακα tasks αλλάξει τότε αλλάζει και η τιμή του taskcode στις αντίστοιχες εγγραφές από τον πίνακα task_dates.

4.5.9. Πίνακας task_googlecalendar

Στον πίνακα task_googlecalendar αποθηκεύονται οι πληροφορίες για τα tasks που έχουν ενημερωθεί στο Google Calendar μετά την επίλυση.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	username	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	googlecode	varchar(200)	utf8_general_ci		Όχι	*Κανένα*
☐	taskcode	int(11)			Όχι	*Κανένα*
☐	periodcode	tinyint(4)			Όχι	*Κανένα*
☐	googledate	date			Όχι	*Κανένα*
☐	googletime	varchar(10)	utf8_general_ci		Όχι	*Κανένα*
☐	interdur	tinyint(4)			Όχι	*Κανένα*
☐	complete	tinyint(1)			Όχι	0

Εικόνα 14: πίνακας task_googlecalendar

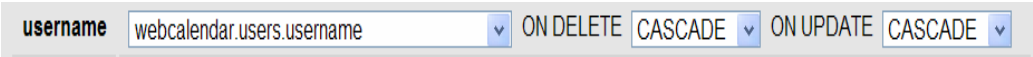
Πίνακας 9: πεδία του πίνακα task_googlecalendar

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
googlecode	Το id event που δημιουργείται σε κάποιο subcalendar του χρήστη όταν ενημερώνεται το task στο Google Calendar.
taskcode	Ο κωδικός του task.
periodcode	Ο αύξοντας αριθμός περιόδου όταν το task είναι περιοδικό, διαφορετικά είναι μηδέν.
googledate	Η ημερομηνία ενημέρωσης του task στο Google Calendar.
googletime	Η ώρα ενημέρωσης του task στο Google Calendar.
interdur	Η διάρκεια του task σε πλήθος ημίωρων.
complete	Αν είναι ολοκληρωμένο ή όχι το task ή το subtask.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα task_googlecalendar είναι τα εξής:

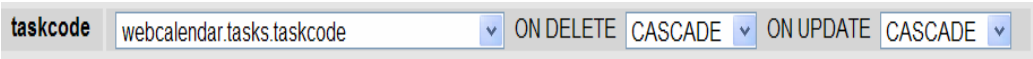
- username + googlecode: το πρωτεύον κλειδί – primary key.

- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα task_googlecalendar (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_googlecalendar ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα task_googlecalendar.

- taskcode: ξένο κλειδί – foreign key. Το πεδίο taskcode του πίνακα task_googlecalendar (child table) συνδέεται με το πεδίο taskcode (primary key) του πίνακα tasks (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του taskcode από τον πίνακα tasks διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα task_googlecalendar ενώ αν η τιμή του taskcode από τον πίνακα tasks αλλάξει τότε αλλάζει και η τιμή του taskcode στις αντίστοιχες εγγραφές από τον πίνακα task_googlecalendar.

4.5.10. Πίνακας template

Στον πίνακα template αποθηκεύονται οι πληροφορίες για τα πρότυπα του τομέα-domain μιας εργασίας του χρήστη. Τα πρότυπα, ανάλογα την κατηγορία τους (day, week, month), περιέχουν ημίωρα ή ημέρες μήνα που μπορεί ο χρήστης να εκτελέσει την εργασία του.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	**username**	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	**tempcode**	int(11)			Όχι	*Κανένα*
☐	**tempname**	varchar(50)	utf8_general_ci		Όχι	*Κανένα*
☐	**category**	varchar(1)	utf8_general_ci		Όχι	*Κανένα*
☐	**hourday**	text	utf8_general_ci		Ναι	*NULL*

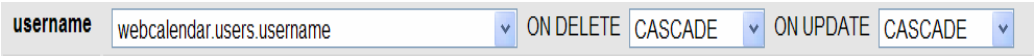
Εικόνα 15: πίνακας template

Πίνακας 10: πεδία του πίνακα template

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
tempcode	Ο κωδικός του προτύπου-template.
tempname	Η περιγραφή του προτύπου-template.
category	Η κατηγορία του προτύπου-template. Οι τιμές είναι d (day), w (week), m (month).
hourday	Πεδίο κειμένου που περιέχει ημίωρα ή ημέρες μήνα του προτύπου-template, τα οποία εμφανίζονται με πράσινο χρώμα στην εφαρμογή.

Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα template είναι τα εξής:

- username + tempcode: το πρωτεύον κλειδί – primary key.
- username + category + tempname: μοναδικό κλειδί – unique key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα template (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα template ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα template.

Ο πίνακας template δεν συνδέεται με τον πίνακα tasks. Ο λόγος είναι ότι ο χρήστης σε μια εργασία-task μπορεί να χρησιμοποιήσει ένα πρότυπο-template μόνο για να ενημερώσει αυτόματα το τομέα της και στη συνέχεια να αλλάξει ή όχι τον τομέα.

4.5.11. Πίνακας constraints

Στον πίνακα constraints αποθηκεύονται οι πληροφορίες για τους περιορισμούς μεταξύ των tasks. Σε έναν περιορισμό, ένα task προηγείται από κάποιο άλλο task. Με βάση τους περιορισμούς αυτούς, τα tasks κατατάσσονται κατά την επίλυση.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο
☐	username	varchar(30)	utf8_general_ci		Όχι	*Κανένα*
☐	taskcode1	int(11)			Όχι	*Κανένα*
☐	taskcode2	int(11)			Όχι	*Κανένα*

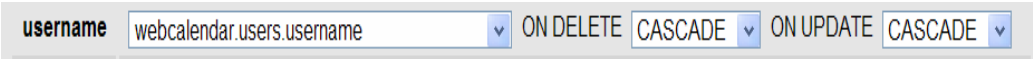
Εικόνα 16: πίνακας constraints

Πίνακας 11: πεδία του πίνακα constraints

Όνομα πεδίου	Περιγραφή
username	Το όνομα του χρήστη.
taskcode1	Ο κωδικός του 1ου task που προηγείται του 2ου task.
taskcode1	Ο κωδικός του 2ου task.

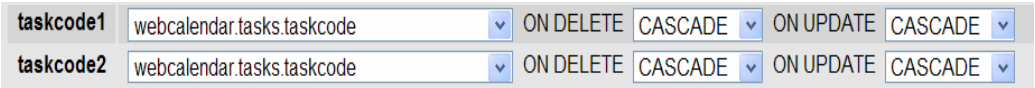
Τα σημαντικότερα ευρετήρια - κλειδιά του πίνακα constraints είναι τα εξής:

- username + taskcode1 + taskcode2: το πρωτεύον κλειδί – primary key.
- username: ξένο κλειδί – foreign key. Το πεδίο username του πίνακα constraints (child table) συνδέεται με το πεδίο username (primary key) του πίνακα users (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του username από τον πίνακα users διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα constraints ενώ αν η τιμή του username από τον πίνακα users αλλάξει τότε αλλάζει και η τιμή του username στις αντίστοιχες εγγραφές από τον πίνακα constraints.

- taskcode1, taskcode2: ξένα κλειδιά – foreign keys. Το πεδίο taskcode1 ή taskcode2 του πίνακα constraints (child table) συνδέεται με το πεδίο taskcode (primary key) του πίνακα tasks (parent table) ως εξής:



Αυτό σημαίνει ότι αν η τιμή του taskcode από τον πίνακα tasks διαγραφεί τότε διαγράφονται οι αντίστοιχες εγγραφές από τον πίνακα constraints ενώ αν η τιμή του taskcode από τον πίνακα tasks αλλάξει τότε αλλάζει και η τιμή του taskcode1 ή taskcode2 στις αντίστοιχες εγγραφές από τον πίνακα constraints.

Κεφάλαιο 5ο

Παρουσίαση εφαρμογής SelfPlanner

Στο κεφάλαιο αυτό στην πρώτη ενότητα παρουσιάζεται ο τρόπος με τον οποίο υλοποιήθηκε η εφαρμογή, δηλαδή ποιες τεχνολογίες χρησιμοποιήθηκαν για να γίνει η υλοποίηση αυτή. Στη δεύτερη ενότητα παρουσιάζεται η λειτουργία της εφαρμογής, από τη μεριά του χρήστη, παραθέτοντας οθόνες και επεξηγήσεις αλλά δίνονται και πληροφορίες που αφορούν τον προγραμματισμό της εφαρμογής. Στη τελευταία ενότητα δίνονται παραδείγματα χρήσης της εφαρμογής.

5.1. Υλοποίηση της εφαρμογής

Για την υλοποίηση της εφαρμογής SelfPlanner χρησιμοποιήθηκαν διάφορες τεχνολογίες, όπως ήδη έχει αναφερθεί. Αρχικά έγινε η εγκατάσταση του Wamp Server. Μέσα από αυτό το περιβάλλον ανάπτυξης χρησιμοποιήθηκε το PhpMyAdmin για τη διαχείριση της βάσης δεδομένων webcalendar και ο Apache Web Server για την παρουσίαση των σελίδων της εφαρμογής.

Για τη συγγραφή των προγραμμάτων χρησιμοποιήθηκαν το σημειωματάριο και το Macromedia DreamWeaver κυρίως για τη σχεδίαση των οθονών. Οι οθόνες σχεδιάστηκαν με βάση τις οθόνες της ήδη υπάρχουσας εφαρμογής SelfPlanner με Java αλλά δημιουργήθηκαν από την αρχή με διαφορετικά εργαλεία ανάπτυξης και έγιναν αλλαγές όπου θεωρήθηκαν απαραίτητες. Για τον κώδικα της εφαρμογής χρησιμοποιήθηκαν Html, Javascript, Css, Xml, Php, Sql, Ajax. Τα περισσότερα προγράμματα υλοποιήθηκαν με html για τη δομή και την μορφοποίηση των σελίδων μαζί με κώδικα javascript και php ενώ τα css χρησιμοποιήθηκαν για τον τρόπο εμφάνισης των στοιχείων της html. Η php χρησιμοποιήθηκε κυρίως για την επικοινωνία με τη βάση δεδομένων webcalendar. Η επικοινωνία αυτή με τις εντολές sql έγινε με τις συναρτήσεις της php. Σε όλα τα προγράμματα χρησιμοποιήθηκε η λειτουργία της Ajax η οποία χρησιμοποιεί JavaScript στον browser και το XMLHttpRequest object για να επικοινωνήσει με το server και χειρίζεται xml ή text δεδομένα που στέλνονται από το server.

Για την εμφάνιση καρτελών και του DatePicker στα προγράμματα χρησιμοποιήθηκε η βιβλιοθήκη JQuery. Τα Google API χρησιμοποιήθηκαν για την επικοινωνία με τα Google Map και Google Calendar και συγκεκριμένα το πακέτο

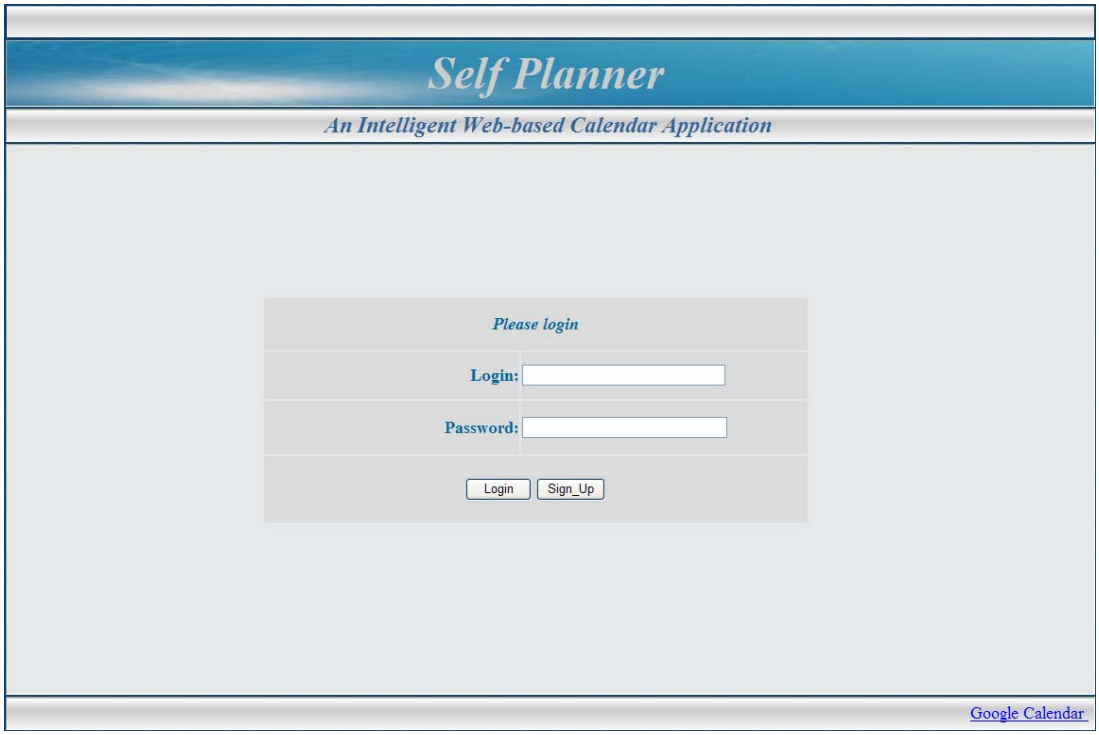
Zend_Gdata χρησιμοποιήθηκε για τη πρόσβαση στα δεδομένα της Google Calendar μέσω της PHP.

Τέλος, χρησιμοποιήθηκε το πρόγραμμα swo.exe, το οποίο εκτελεί τον αλγόριθμο swo για να επιλύσει το πρόβλημα της διαχείρισης προσωπικών εργασιών με χρονικούς περιορισμούς και προτιμήσεις. Τα αποτελέσματα του swo.exe χρησιμοποιήθηκαν από την εφαρμογή για την ενημέρωση των εργασιών στο Google Calendar.

5.2. Περιγραφή λειτουργίας της εφαρμογής

Όλα τα προγράμματα της εφαρμογής βρίσκονται στο φάκελο SelfPlanner, ο οποίος έχει αντιγραφεί κάτω από το φάκελο wamp/www. Ο χρήστης για να εκτελέσει την εφαρμογή SelfPlanner ανοίγει ένα φυλλομετρητή, όπως είναι ο Internet Explorer, και δίνει τη διεύθυνση http://localhost/SelfPlanner/index.htm.

5.2.1. Οθόνες «SelfPlanner Login» και «Sign_Up»



Εικόνα 17: Οθόνη «SelfPlanner Login»

Η πρώτη οθόνη που εμφανίζεται είναι η εισαγωγή του χρήστη στην εφαρμογή. Ο χρήστης επιλέγει Sign_Up για να κάνει εγγραφή ή δίνει κωδικό-Login και Password και επιλέγει Login για να μπει στην εφαρμογή.

Όταν ο χρήστης δώσει το κωδικό-Login και το Password και επιλέξει το κουμπί Login, το πρόγραμμα διαβάζει τα στοιχεία που έδωσε ο χρήστης από τον πίνακα users και αν ο χρήστης υπάρχει εμφανίζεται η οθόνη «Main». Όταν ο χρήστης επιλέξει το κουμπί Sign_Up εμφανίζεται η οθόνη «Sign_Up» για να γίνει εισαγωγή στοιχείων ενός καινούργιου χρήστη.

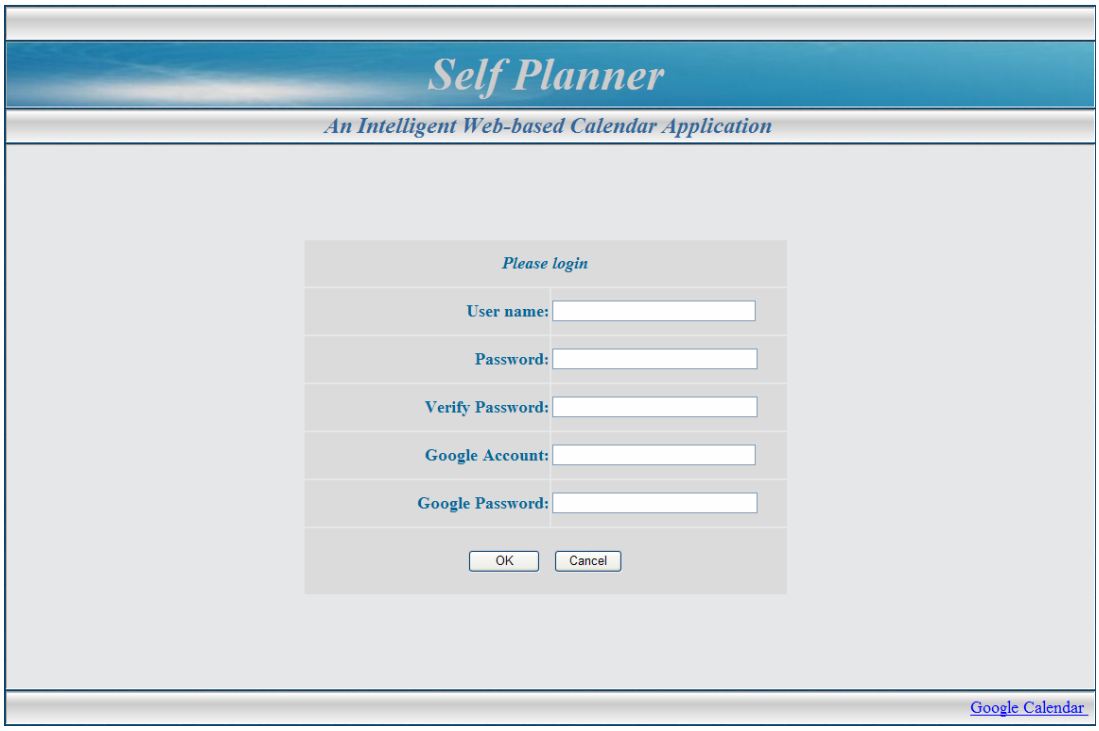
Όλες οι οθόνες έχουν στην κάτω δεξιά γωνία τη σύνδεση του χρήστη με το Google Calendar.

Στη συνέχεια παρουσιάζεται ενδεικτικά κώδικας σε php, που διαβάζει δεδομένα από έναν πίνακα και τα στέλνει στο XMLHttpRequest object με τη μορφή text για να τα διαχειριστεί. Τα πεδία που εμφανίζονται με την εντολή echo στέλνονται στο XMLHttpRequest object και για την ανάκτηση των δεδομένων χρησιμοποιείται η ιδιότητα responseText για απλό κείμενο:

var recs = XMLHttpRequestObject.responseText.

Ο κώδικας που ακολουθεί χρησιμοποιείται παρόμοια από πολλά προγράμματα της εφαρμογής.

```
$sql = "select * from users where users.username = '$username'";
$result = mysql_query($sql);
$row=mysql_fetch_array($result);
echo $row["username"].",".$row["password"];
```

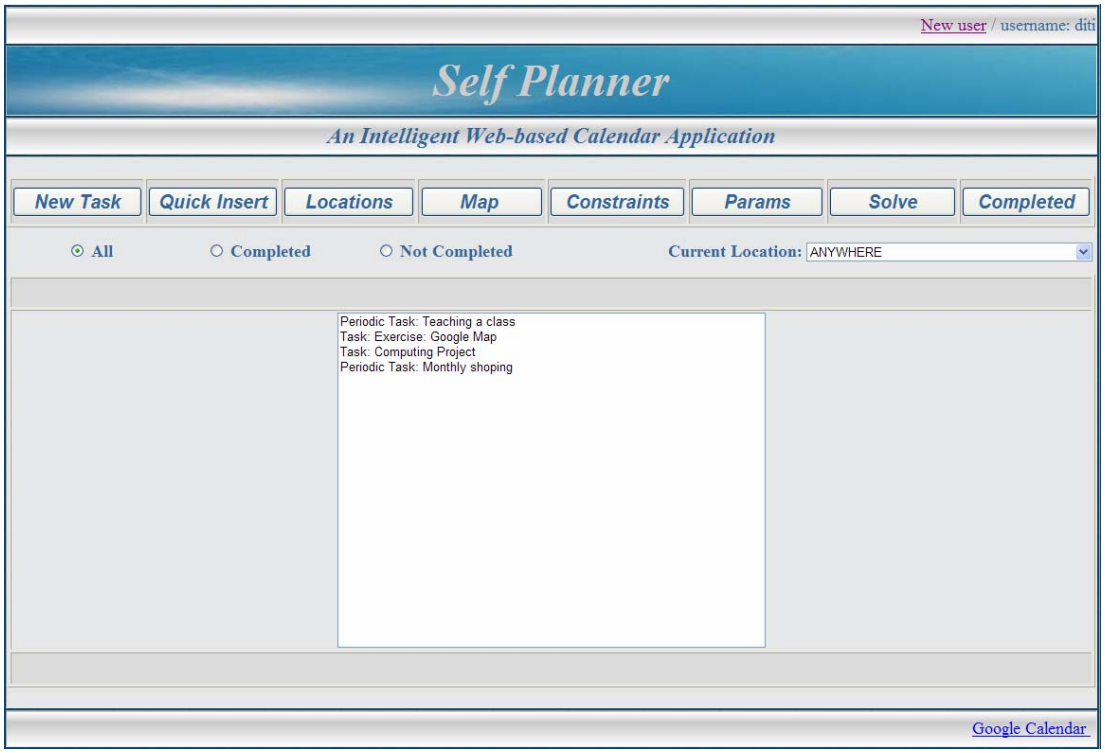


Εικόνα 18: Οθόνη «Sign_Up»

Στην οθόνη «Sign_Up» ο χρήστης εισάγει το username και το password με τα οποία γίνεται η εισαγωγή του στην εφαρμογή και το Google Account και Google Password που είναι τα στοιχεία του λογαριασμού της Google που έχει ο χρήστης. Με τα στοιχεία του λογαριασμού της Google γίνεται η σύνδεση με το Google Calendar.

Τα στοιχεία username, password, Google Account και Google Password εισάγονται στον πίνακα users. Με την εισαγωγή του χρήστη, εισάγεται η τοποθεσία - location με κωδικό 0 και περιγραφή ANYWHERE στον πίνακα locations και ο κωδικός 0 του location ενημερώνεται στον πίνακα users, που είναι η τρέχουσα τοποθεσία που βρίσκεται ο χρήστης.

5.2.2. Οθόνη «Main»



Εικόνα 19: Οθόνη «Main»

Στην οθόνη «Main» υπάρχει το μενού της εφαρμογής, το οποίο αποτελείται από τα εξής Command buttons:

- *New Task*: Ο χρήστης εισάγει μία καινούργια εργασία – task με όλες τις λεπτομέρειες που έχει μια εργασία.
- *Quick Insert*: Ο χρήστης εισάγει μία καινούργια εργασία – task στην οποία υπάρχουν περιορισμένες επιλογές.
- *Locations*: Ο χρήστης διαχειρίζεται τις κλάσεις των τοποθεσιών - class locations.
- *Map*: Ο χρήστης διαχειρίζεται τις τοποθεσίες-locations στο Google Map.
- *Constraints*: Ο χρήστης διαχειρίζεται τους περιορισμούς ταξινόμησης μεταξύ των εργασιών.
- *Params*: Ο χρήστης μπορεί να επιλέξει το default calendar και να αλλάξει το password και τα στοιχεία του λογαριασμού της Google.

- *Solve*: Γίνεται η επίλυση των εργασιών και η τοποθέτησή τους στο Google Calendar.
- *Completed*: Ο χρήστης επιλέγει από τις εργασίες που έχουν ενημερωθεί στο Google Calendar ποιες πραγματικά έχει εκτελέσει.

Υπάρχουν 3 Radio buttons. Όταν ο χρήστης επιλέξει ένα από τα Radio buttons, εμφανίζονται οι κατάλληλες εργασίες στη λίστα που βρίσκεται κάτω από αυτά:

- *All*: εμφανίζονται όλες οι εργασίες.
- *Completed*: εμφανίζονται οι ολοκληρωμένες εργασίες, δηλαδή οι εργασίες που έχουν ενημερωθεί στο Google Calendar και ο χρήστης έχει επιλέξει ότι πράγματι έχουν εκτελεστεί.
- *Not Completed*: εμφανίζονται οι μη ολοκληρωμένες εργασίες.

Το «current location» είναι η τρέχουσα τοποθεσία που βρίσκεται ο χρήστης. Ο χρήστης μπορεί να αλλάξει τη τρέχουσα τοποθεσία επιλέγοντας μια άλλη από τη λίστα που εμφανίζει τις τοποθεσίες που υπάρχουν στο Google Map. Με την επιλογή που κάνει ο χρήστης ενημερώνεται ο πίνακας users με το καινούργιο «current location» του χρήστη.

Στο κέντρο της οθόνης εμφανίζεται μία λίστα με τις υπάρχουσες εργασίες, που βρίσκονται στον πίνακα tasks, όπου ο χρήστης μπορεί να επιλέξει μία εργασία για να τη διαχειριστεί.

Όλες οι οθόνες έχουν στην πάνω δεξιά γωνία το όνομα του χρήστη και τη σύνδεση «New user» που πηγαίνει στην αρχική σελίδα «SelfPlanner Login». Όλα τα αρχεία που διαχειρίζεται ο χρήστης αφορούν τα δικά του αρχεία.

Το όνομα του χρήστη που εμφανίζεται σε όλες τις οθόνες είναι μία session μεταβλητή. Οι μεταβλητές sessions [19] είναι μεταβλητές οι οποίες δημιουργούνται όταν ο χρήστης μπει στην εφαρμογή και αποθηκεύονται στον server, ενώ χάνονται όταν ο χρήστης βγει από την εφαρμογή. Η έναρξη ενός session γίνεται με τη χρήση του session_start() και η ανάκτηση μιας session μεταβλητής γίνεται με το $_SESSION['variable']. Συγκεκριμένα στην εφαρμογή SelfPlanner, μία session μεταβλητή είναι η username που είναι το όνομα του χρήστη που έχει εισέλθει στην

εφαρμογή. Για να ανατεθεί το όνομα του χρήστη που βρίσκεται στον πίνακα users στη session μεταβλητή username χρησιμοποιείται η εντολή: $_SESSION['username'] = $row["username"].

Όταν ο χρήστης εισέρχεται πρώτη φορά στην εφαρμογή θα πρέπει να εισάγει τοποθεσίες-locations στην οθόνη «Map». Χωρίς να είναι υποχρεωτικό, ο χρήστης μπορεί να εισάγει κλάσεις τοποθεσιών στην οθόνη «Locations» και να ορίσει default calendar στην οθόνη «Params» και στη συνέχεια να διαχειριστεί τις εργασίες του. Οι οθόνες θα παρουσιαστούν με αυτή τη σειρά παρακάτω, παρόλο που στην εφαρμογή εμφανίζονται με σειρά που σχετίζεται με τη χρήση τους.

Στη συνέχεια παρουσιάζεται ενδεικτικά κώδικας σε php, που διαβάζει δεδομένα από έναν πίνακα και εμφανίζει πολλές εγγραφές, όπως την εμφάνιση των tasks, και τα στέλνει στο XMLHttpRequest object με τη μορφή XML για να τα διαχειριστεί. Τα πεδία που εμφανίζονται με την εντολή echo στέλνονται στο XMLHttpRequest object. Ο κώδικας που ακολουθεί χρησιμοποιείται παρόμοια από πολλά προγράμματα της εφαρμογής.

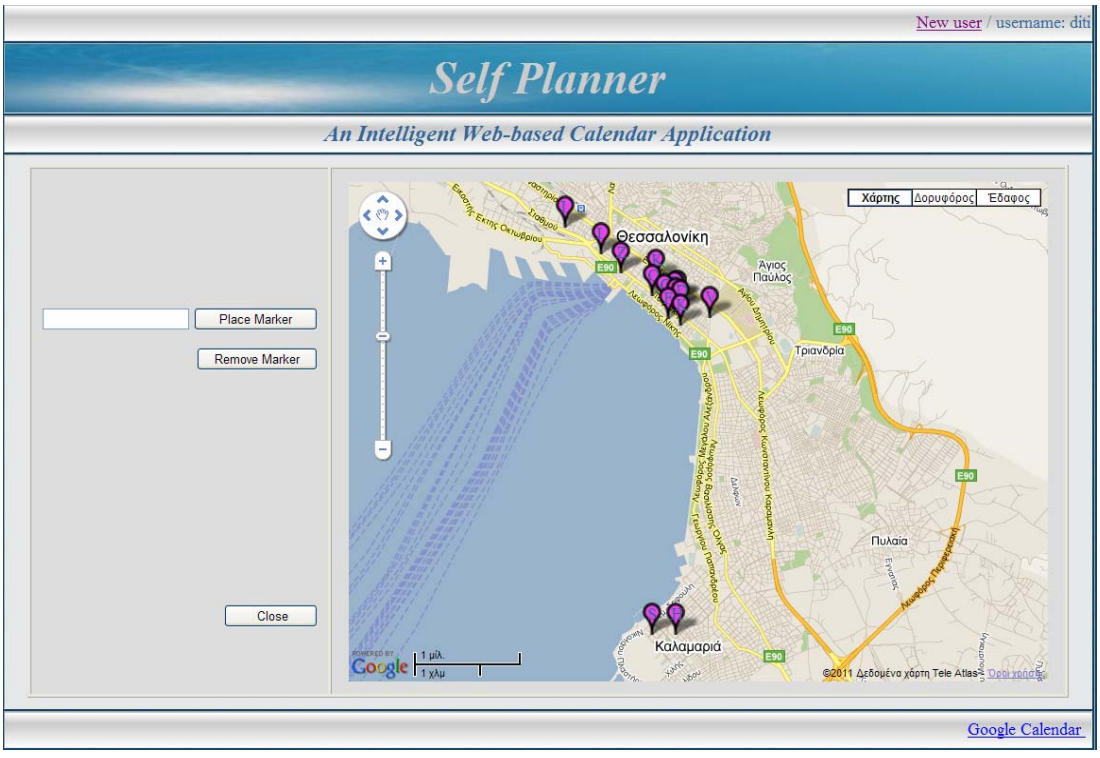
```
$sql = "select * from tasks where tasks.username = '$username' && tasks.complete = '$complete' ";
$result = mysql_query($sql);
echo '<?xml version="1.0" encoding="ISO-8859-7" ?>';
echo '<listask>';
while($row=mysql_fetch_array($result))
{
 echo "<ltcode>" . $row['taskcode'] . "</ltcode>";
 echo "<ltask>" . $row['taskname'] . "</ltask>";
 echo "<perd>" .$row['periodic']. "</perd>";
}
echo '</listask>';
```

Για την ανάκτηση των δεδομένων χρησιμοποιείται η ιδιότητα responseXML για XML δεδομένα και για την εμφάνιση συγκεκριμένου πεδίου χρησιμοποιείται η εντολή: xmldoc.getElementsByTagName("ltask")[i].childNodes[0].nodeValue.

Ακολουθεί ένα παράδειγμα, στο οποίο εμφανίζονται XML δεδομένα σε μία λίστα όπου δημιουργούνται τόσες επιλογές όσες είναι οι εγγραφές:

```
var xmldoc = XMLHttpRequestObject.responseXML;
tleng =xmldoc.getElementsByTagName("ltask").length;
var ld=document.getElementById("listasks");
for (i=0;i<tleng;i++)
{
        var optnew=document.createElement('option');
        optnew.text=xmldoc.getElementsByTagName("ltask")[i].childNodes[0].nodeValue;
        ld.add(optnew);
}
```

5.2.3. Οθόνη «Map»



Εικόνα 20: Οθόνη «Map»

Στην οθόνη «Map» εμφανίζεται ο χάρτης της Google. Για να εμφανιστεί ο χάρτης στο html αρχείο χρειάζονται τα εξής:

- Το javascript της Google μαζί με το Map Api key που δηλώνεται στο <head>:
  <script
  src="http://maps.google.com/maps?file=api&v=2&hl=el&oe=utf-8&key=ABQIAAAAJ_VOu2suojQjVQOf1awTKxSrgP1H14j86luPB6hadaGnNCvvShSjmHMVyawNoJHIrxp0lT4j2ejk8A"
  type="text/javascript"></script>
- Κώδικας javascript με διάφορες παραμέτρους του χάρτη:
  // δημιουργία και εμφάνιση χάρτη
  map = new GMap2(document.getElementById("map"));
  // δημιουργία σημείου με συντεταγμένες της Θεσσαλονίκης, κεντράρισμα του χάρτη στο σημείο αυτό και zoom 13
  map.setCenter(new GLatLng(40.613691819957566, 22.951126098632812), 13);
- Εμφάνιση του χάρτη σε συγκεκριμένη θέση της σελίδας:
  Συνήθως ο χρήστης ορίζει ένα div για να εμφανιστεί εκεί μέσα ο χάρτης. Για να γίνει αυτό πρέπει το id του div να συμφωνεί με το id που έχει δηλωθεί στο new GMap2(document.getElementById("map"):
  .

Στην οθόνη «Map» ο χρήστης μπορεί να διαχειριστεί τις τοποθεσίες. Όταν εμφανίζεται η οθόνη «Map» το πρόγραμμα διαβάζει τις τοποθεσίες από τον πίνακα locations. Τα στοιχεία του πίνακα locations είναι η περιγραφή της τοποθεσίας και οι συντεταγμένες του σημείου στο οποίο βρίσκεται η τοποθεσία. Κάθε φορά που διαβάζεται μία τοποθεσία με τις συντεταγμένες της, το πρόγραμμα δημιουργεί ένα marker στις συντεταγμένες αυτές πάνω στο Google Map.

Ο χρήστης για να εισάγει μία τοποθεσία, πληκτρολογεί το όνομα της στο πλαίσιο κειμένου, επιλέγει το κουμπί «Place Marker» και κάνει κλικ πάνω στο χάρτη στο σημείο που θέλει να μπει η τοποθεσία. Το πρόγραμμα βρίσκει τις συντεταγμένες του σημείου και μαζί με την περιγραφή της τοποθεσίας τις καταχωρεί στον πίνακα locations. Στη συνέχεια για κάθε καινούργια τοποθεσία υπολογίζεται η χρονική απόσταση σε λεπτά της τοποθεσίας αυτής με τις υπόλοιπες τοποθεσίες μέσω

αυτοκινήτου. Οι χρονικές αποστάσεις μεταξύ των τοποθεσιών καταχωρούνται στον πίνακα locations_duration.

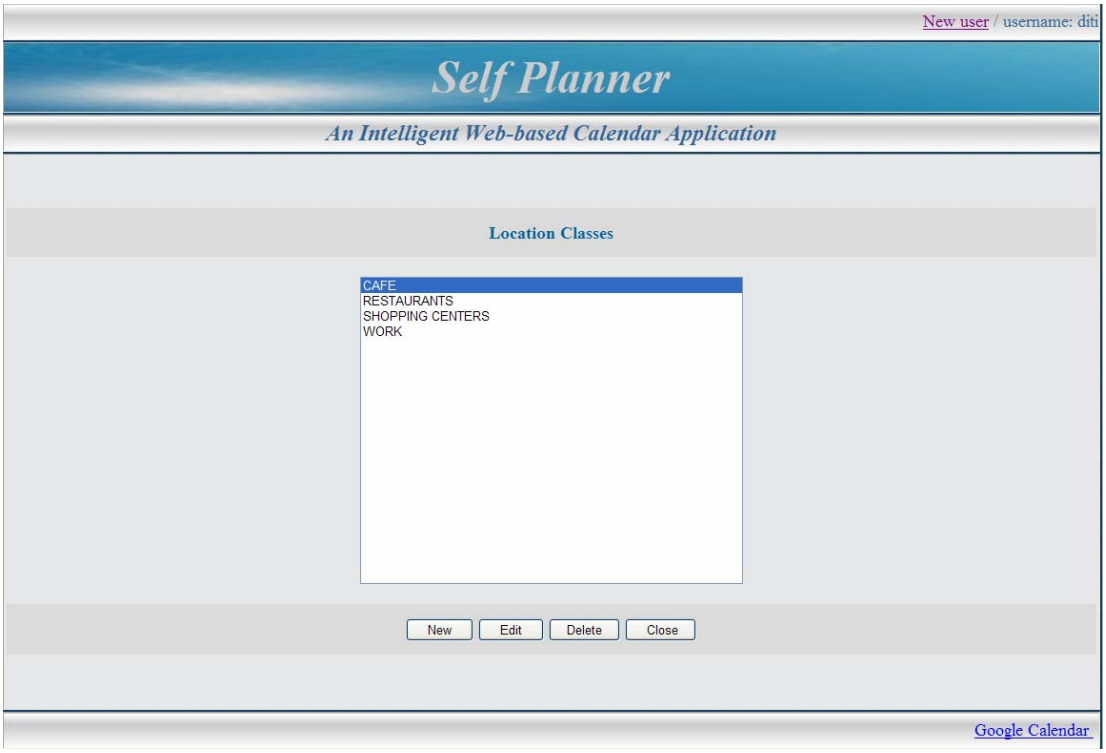
Ο χρήστης για να διαγράψει μία τοποθεσία, επιλέγει το κουμπί «Remove Marker» και κάνει κλικ πάνω στο marker που θέλει να διαγράψει. Διαγράφεται το marker από το χάρτη και από τον πίνακα locations και τους σχετικούς πίνακες μ' αυτόν.

Όταν ο χρήστης κάνει κλικ πάνω σε ένα marker εμφανίζεται ένα σύννεφο διαλόγου με το όνομα της τοποθεσίας.

Η διαχείριση της εφαρμογής με τον χάρτη γίνεται με τις ιδιότητες και τις μέθοδοι των Google Map API [14]. Μερικές από τις ιδιότητες και τις μεθόδους που χρησιμοποιήθηκαν από την εφαρμογή είναι:

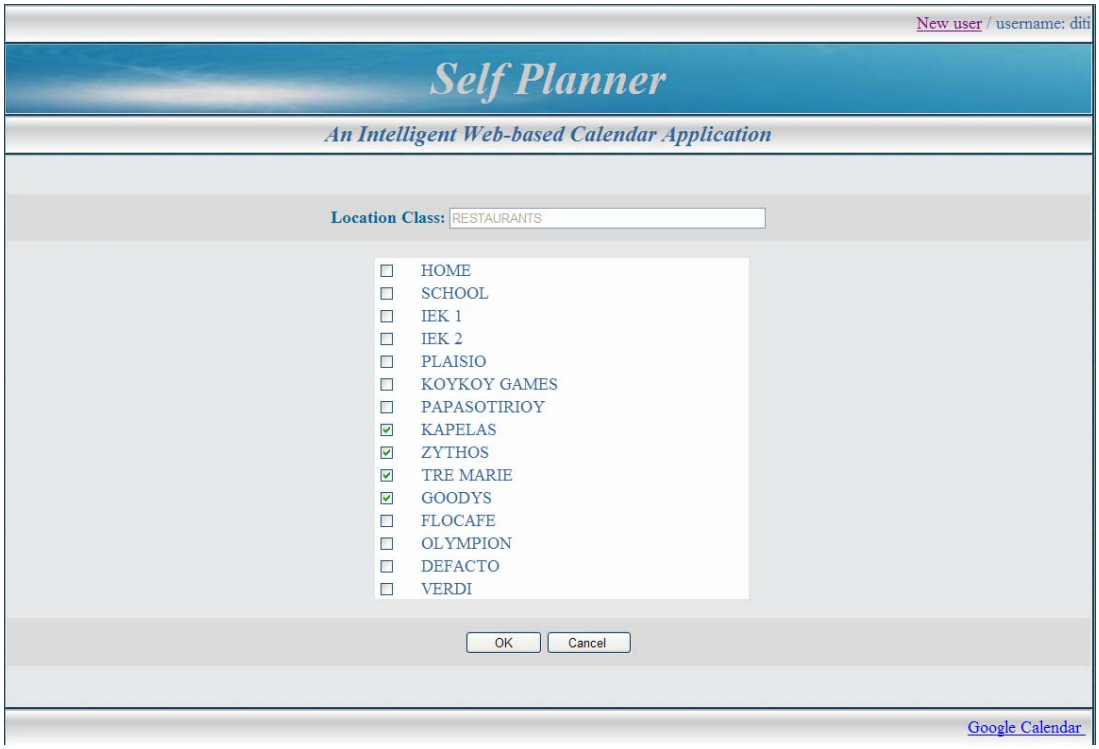
- Η δημιουργία σημείου πάνω στο χάρτη έγινε με τη δημιουργία του αντικειμένου GLatLng, το οποίο περιέχει τις συντεταγμένες του σημείου που είναι το γεωγραφικό πλάτος και γεωγραφικό μήκος.
- Για την εισαγωγή marker χρησιμοποιήθηκε η μέθοδος addOverlay().
- Για την αφαίρεση marker χρησιμοποιήθηκε η μέθοδος removeOverlay()
- Η εμφάνιση σύννεφου διαλόγου – παράθυρο σε συγκεκριμένο σημείο έγινε με τη μέθοδο openInfoWindow().
- Η μέθοδος που χειρίζεται τα γεγονότα είναι η addListener().
- Η χρονική απόσταση σε λεπτά μεταξύ δύο τοποθεσιών υπολογίστηκε με τη μέθοδο directions.getDuration().

5.2.4. Οθόνες «Locations» και «Location Class»



Εικόνα 21: Οθόνη «Locations»

Στην οθόνη «Locations» εμφανίζονται οι κλάσεις τοποθεσιών (Location Classes) που βρίσκονται στον πίνακα classonly. Ο χρήστης μπορεί να εισάγει μια καινούργια κλάση επιλέγοντας το κουμπί «New», να μεταβάλλει μία κλάση επιλέγοντας το κουμπί «Edit» ή να διαγράψει μία κλάση επιλέγοντας το κουμπί «Delete». Όταν ο χρήστης επιλέξει «New» ή «Edit» εμφανίζεται η οθόνη «Location Class».

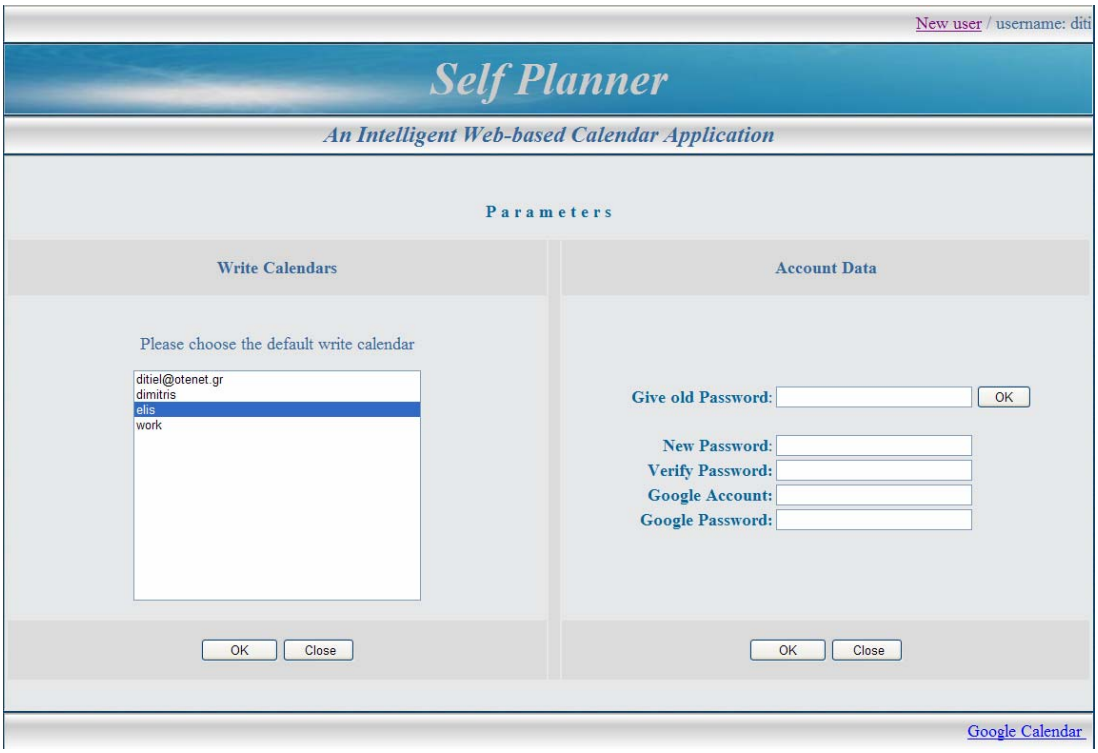


Εικόνα 22: Οθόνη «Location Class»

Στην οθόνη «Location Class» ο χρήστης μπορεί να διαχειριστεί μία κλάση, που είναι μία ομάδα, με τις τοποθεσίες της. Αν ο χρήστης έχει επιλέξει να εισάγει μία καινούργια κλάση τότε μπορεί να πληκτρολογήσει το όνομα της κλάσης και να επιλέξει τις τοποθεσίες που έχει. Αν ο χρήστης έχει επιλέξει να μεταβάλλει μία κλάση τότε μπορεί να επιλέξει τις τοποθεσίες που έχει. Όταν ο χρήστης καταχωρεί μια εργασία, δίνει και την τοποθεσία που θα γίνει αυτή η εργασία ή δίνει μια κλάση-ομάδα τοποθεσιών. Στην περίπτωση που δώσει κλάση τοποθεσιών, κατά την εκτέλεση της εργασίας στο ημερολόγιο, θα γίνει επιλογή τοποθεσίας μεταξύ των τοποθεσιών που υπάρχουν στην κλάση από τον αλγόριθμο SWO.

Το όνομα της κλάσης βρίσκεται στον πίνακα classonly, οι τοποθεσίες βρίσκονται στον πίνακα locations και οι τοποθεσίες μιας συγκεκριμένης κλάσης βρίσκονται στον πίνακα class_locations.

5.2.5. Οθόνη «Params»



Εικόνα 23: Οθόνη «Params»

Στην οθόνη «Params» υπάρχουν 2 επιλογές στις οποίες ο χρήστης ορίζει τις παραμέτρους της εφαρμογής.

Στη 1η επιλογή ο χρήστης επιλέγει το default calendar από τη λίστα των Subcalendars που έχει στο Google Calendar. Με τα στοιχεία του λογαριασμού της Google (account, password), που είναι καταχωρημένα στον πίνακα users, το πρόγραμμα διαβάζει τα calendars που έχει ο χρήστης στο Google Calendar και τα εμφανίζει στη λίστα του προγράμματος. Με αυτόν τον τρόπο ο χρήστης πάντα βλέπει τα calendars που έχει αυτή τη στιγμή. Δηλαδή οποιαδήποτε αλλαγή γίνει μέσα από το Google Calendar θα φανεί άμεσα στην εφαρμογή. Το default calendar του χρήστη αποθηκεύεται στον πίνακα users.

Στη συνέχεια παρουσιάζεται ενδεικτικά κώδικας php για την εμφάνιση subcalendars της Google ενός συγκεκριμένου χρήστη. Για να υπάρχει πρόσβαση στα δεδομένα της Google Calendar χρησιμοποιείται η κλάση Zend_Gdata_Calendar. Για

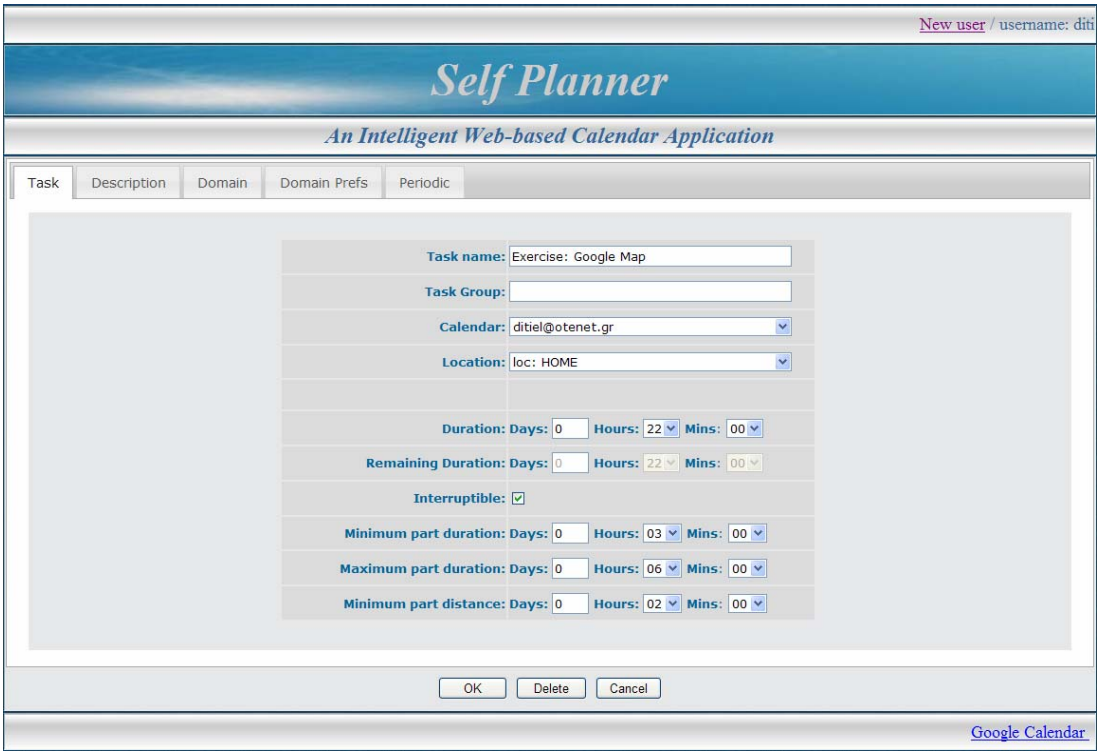
να δημιουργηθεί μια σύνδεση με τους διακομιστές της Google πρέπει να δημιουργηθεί ένας πελάτης-client και γι' αυτόν τον πελάτη να δεσμευτεί μια υπηρεσία Zend_Gdata_Calendar [22]. Στο κώδικα που ακολουθεί, για τη δημιουργία του πελάτη δίνονται από την εφαρμογή τα στοιχεία του λογαριασμού της Google (account, password). Στη συνέχεια, γι' αυτόν τον πελάτη διαβάζονται τα subcalendars της Google, τα οποία στέλνονται στο XMLHttpRequest object με τη μορφή XML για να τα διαχειριστεί.

```
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Gdata');
Zend_Loader::loadClass('Zend_Gdata_ClientLogin');
Zend_Loader::loadClass('Zend_Gdata_Calendar');
$client = getGDataClient($USER, $PASSWORD);
if ($client!=null)
        printCalendarList($client);

function getGDataClient($user, $pass) {
$service = Zend_Gdata_Calendar::AUTH_SERVICE_NAME;
$client = Zend_Gdata_ClientLogin::getHttpClient($user, $pass, $service);
return $client;
}
function printCalendarList($client) {
echo '<?xml version="1.0" encoding="utf-8" ?>';
echo '<calends>';
$gdataCal = new Zend_Gdata_Calendar($client);
$calFeed = $gdataCal->getCalendarListFeed();
foreach ($calFeed as $calendar) {
  echo "<calend>" . $calendar->title->text . "</calend>";
}
echo '</calends>';
}
```

Στη 2η επιλογή ο χρήστης πληκτρολογώντας το παλιό του password σωστά, το πρόγραμμα διαβάζει τα στοιχεία από τον πίνακα users και τα εμφανίζει για να μπορέσει ο χρήστης αν θέλει να τα αλλάξει.

5.2.6. Οθόνη «New Task»



Εικόνα 24: Οθόνη «New Task»: Καρτέλα «Task»

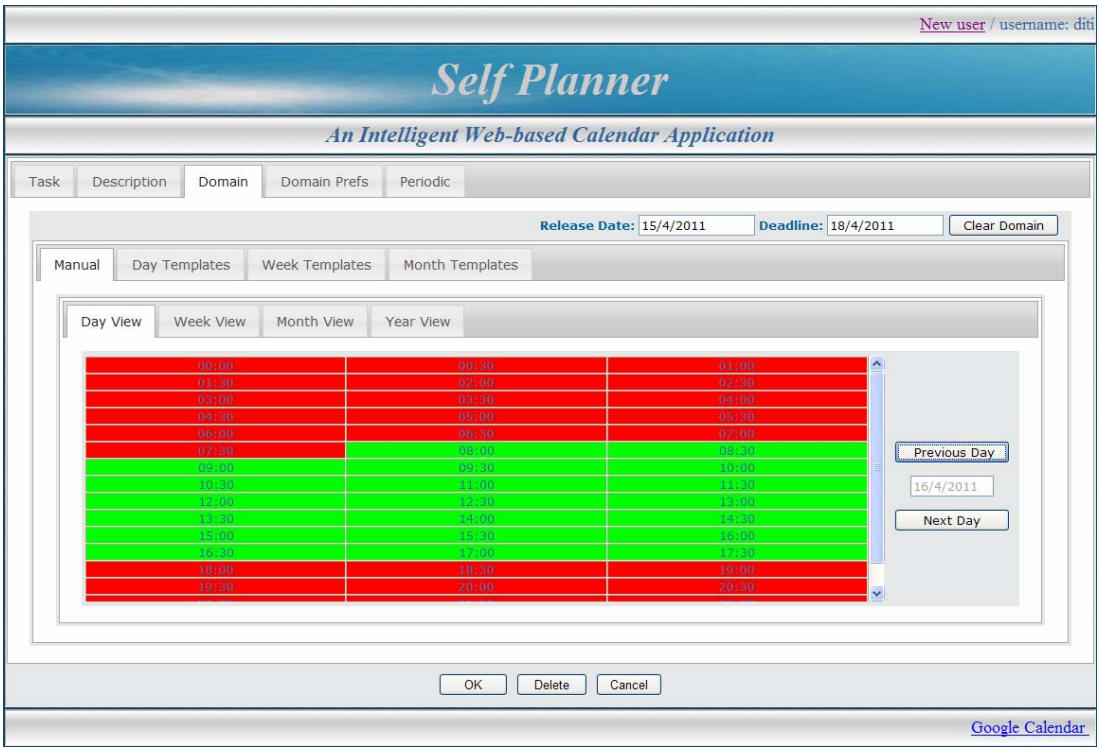
Στην οθόνη «New Task» ο χρήστης εισάγει, μεταβάλλει ή διαγράφει μία εργασία. Όταν ο χρήστης καταχωρεί μία εργασία το πλήκτρο «Delete» είναι ανενεργό. Τα στοιχεία που μπορεί ο χρήστης να καταχωρήσει σε μια εργασία χωρίζονται σε 5 καρτέλες. Στη συνέχεια δίνεται επεξήγηση των πεδίων ανά καρτέλα.

Πίνακας 12: Τα πεδία της καρτέλας «Task»

Πεδίο	Περιγραφή
Task name	Το όνομα της εργασίας – task.
Task Group	Η ομάδα της εργασίας.
Calendar	Ένα από τα subcalendars που έχει ο χρήστης στο Google Calendar. Όταν εκτελείται η επίλυση-solve, τότε η εργασία τοποθετείται σε αυτό το subcalendar στο Google Calendar του χρήστη. Στο πεδίο Calendar υπάρχει μια λίστα που εμφανίζει όλα τα subcalendars της Google και ο χρήστης μπορεί να επιλέξει ένα από αυτά. Με την είσοδο στο πρόγραμμα «New Task» διαβάζονται τα subcalendars της Google και ενημερώνουν τη λίστα. Στη συνέχεια διαβάζεται το default calendar του χρήστη από τον πίνακα users, και ενημερώνει την τρέχουσα θέση της λίστας.
Location	Η τοποθεσία-location που ο χρήστης θέλει να εκτελέσει την εργασία. Στο πεδίο Location υπάρχει μια λίστα που εμφανίζει όλες τις τοποθεσίες-locations και όλες τις κλάσεις τοποθεσιών-class locations και ο χρήστης μπορεί να επιλέξει μία από αυτές. Οι τοποθεσίες διαβάζονται από τον πίνακα locations και οι κλάσεις τοποθεσιών διαβάζονται από τον πίνακα classonly.
Duration Days, Hours, Mins	Ο χρόνος που διαρκεί μία εργασία. Ο χρήστης μπορεί να πληκτρολογήσει ημέρες-Days, να επιλέξει ώρες-Hours από μια λίστα με τιμές από 00 ως 23 ή να επιλέξει λεπτά-Mins από μια λίστα με τιμές 00 ή 30.
Remaining Duration	Ο χρόνος που υπολείπεται για να ολοκληρωθεί μία

Days, Hours, Mins	εργασία σε ημέρες, ώρες και λεπτά. Μία εργασία αφού έχει ενημερωθεί στο Google Calendar, ο χρήστης επιλέγει αν ένα μέρος της εργασίας ή ολόκληρη η εργασία έχει πραγματοποιηθεί. Στη περίπτωση αυτή ο χρόνος-duration που έχει πραγματοποιηθεί αφαιρείται από το Remaining Duration. Τα πεδία του Remaining Duration ενημερώνονται στο πρόγραμμα «Completed» και γι' αυτό είναι ανενεργά και δεν μπορεί να τα αλλάξει ο χρήστης.
Interruptible	Ο χρήστης επιλέγει αν η εργασία είναι διακοπτόμενη ή όχι. Αν η εργασία είναι διακοπτόμενη ο χρήστης επιλέγει το checkbox και τα πεδία Minimum part duration, Maximum part duration, Minimum part distance γίνονται ενεργά.
Minimum part duration Days, Hours, Mins	Για μία διακοπτόμενη εργασία είναι το ελάχιστο χρονικό διάστημα σε ημέρες, ώρες, λεπτά που μπορεί να έχει το μέρος της εργασίας που διακόπτεται. Ο χρήστης μπορεί να πληκτρολογήσει ημέρες-Days, να επιλέξει ώρες-Hours από μια λίστα με τιμές από 00 ως 23 ή να επιλέξει λεπτά-Mins από μια λίστα με τιμές 00 ή 30.
Maximum part duration Days, Hours, Mins	Για μία διακοπτόμενη εργασία είναι το μέγιστο χρονικό διάστημα σε ημέρες, ώρες, λεπτά που μπορεί να έχει το μέρος της εργασίας που διακόπτεται. Ο χρήστης μπορεί να πληκτρολογήσει ημέρες-Days, να επιλέξει ώρες-Hours από μια λίστα με τιμές από 00 ως 23 ή να επιλέξει λεπτά-Mins από μια λίστα με τιμές 00 ή 30.
Minimum part distance Days, Hours, Mins	Για μία διακοπτόμενη εργασία είναι το ελάχιστο χρονικό διάστημα σε ημέρες, ώρες, λεπτά που μπορεί να απέχουν μεταξύ τους τα μέρη της εργασίας που διακόπτεται. Ο χρήστης μπορεί να πληκτρολογήσει ημέρες-Days, να επιλέξει ώρες-Hours από μια λίστα με τιμές από 00 ως 23 ή να επιλέξει λεπτά-Mins από μια λίστα με τιμές 00 ή 30.

Στην καρτέλα «Description» ο χρήστης εισάγει την περιγραφή της εργασίας, που είναι ένα μεγάλο πεδίο κειμένου.



Εικόνα 25: Οθόνη «New Task»: Καρτέλα «Domain»

Στην καρτέλα «Domain» ο χρήστης μπορεί να επιλέξει το χρόνο που είναι διαθέσιμος για την εκτέλεση της εργασίας. Η καρτέλα «Domain» χωρίζεται σε καρτέλες ώστε ο χρήστης να επιλέξει τον τρόπο που θα δώσει το χρόνο εκτέλεσης της εργασίας. Οι καρτέλες έχουν κελιά με τιμές ημίωρα ή ημέρες όπου ο χρήστης κάνει κλικ πάνω τους. Αρχικά όλα τα κελιά έχουν κόκκινο χρώμα. Όταν ο χρήστης κάνει κλικ σε ένα κελί αυτό αλλάζει και γίνεται πράσινο αν είναι κόκκινο ή γίνεται κόκκινο αν είναι πράσινο. Το κόκκινο χρώμα σημαίνει ότι δεν ανήκει στο χρόνο εκτέλεσης της εργασίας ενώ το πράσινο χρώμα ότι ανήκει στο χρόνο εκτέλεσης της εργασίας. Τα κελιά που είναι πριν την ημερομηνία έναρξης της εργασίας – Release Date ή μετά την ημερομηνία λήξης της εργασίας –Deadline έχουν μαύρο χρώμα.

Στον κώδικα του προγράμματος, για να υπάρχουν όλες αυτές οι πληροφορίες που αφορούν τον τομέα ακολουθείται το εξής: για κάθε ημερομηνία που βρίσκεται στο διάστημα από Release Date μέχρι και Deadline, δημιουργείται αντικείμενο με πεδία την ημερομηνία και έναν πίνακα με ώρες και χρώμα.

Στη συνέχεια παρουσιάζονται τα κοινά πεδία για όλες τις καρτέλες.

Πίνακας 13: Τα κοινά πεδία της καρτέλας «Domain»

Πεδίο	Περιγραφή
Release Date	Η ημερομηνία έναρξης της εργασίας – task. Όταν ο χρήστης καταχωρεί μία καινούργια εργασία, το Release Date παίρνει τη τρέχουσα ημερομηνία του συστήματος. Όταν ο χρήστης κάνει κλικ στο πεδίο του Release Date που είναι ένα πλαίσιο κειμένου, εμφανίζεται ένα ημερολόγιο ανά μήνα (Datepicker) όπου μπορεί να επιλέξει την ημερομηνία που επιθυμεί ή να πληκτρολογήσει την ημερομηνία στο πλαίσιο κειμένου. Αν ο χρήστης δώσει Release Date μεγαλύτερη από το Deadline υπολογίζεται ξανά το Deadline με τον ίδιο τρόπο, δηλαδή το Deadline παίρνει την τιμή του Release Date συν 7 ημέρες.
Deadline	Η ημερομηνία λήξης της εργασίας – task. Όταν ο χρήστης καταχωρεί μία καινούργια εργασία, το Deadline παίρνει την τιμή του Release Date συν 7 ημέρες. Όταν ο χρήστης κάνει κλικ στο πεδίο του Deadline που είναι ένα πλαίσιο κειμένου, εμφανίζεται ένα ημερολόγιο ανά μήνα (Datepicker) όπου μπορεί να επιλέξει την ημερομηνία που επιθυμεί ή να πληκτρολογήσει την ημερομηνία στο πλαίσιο κειμένου. Αν ο χρήστης δώσει Deadline μικρότερη από το Release Date υπολογίζεται ξανά το Deadline με τον ίδιο τρόπο, δηλαδή το Deadline παίρνει την τιμή του Release Date συν 7 ημέρες.
Clear Domain	Όταν ο χρήστης επιλέξει το κουμπί «Clear Domain» τότε όλα τα κελιά παίρνουν κόκκινο χρώμα. Με αυτόν τον τρόπο, ο χρήστης μπορεί από την αρχή να επιλέξει τον χρόνο εκτέλεσης της εργασίας.

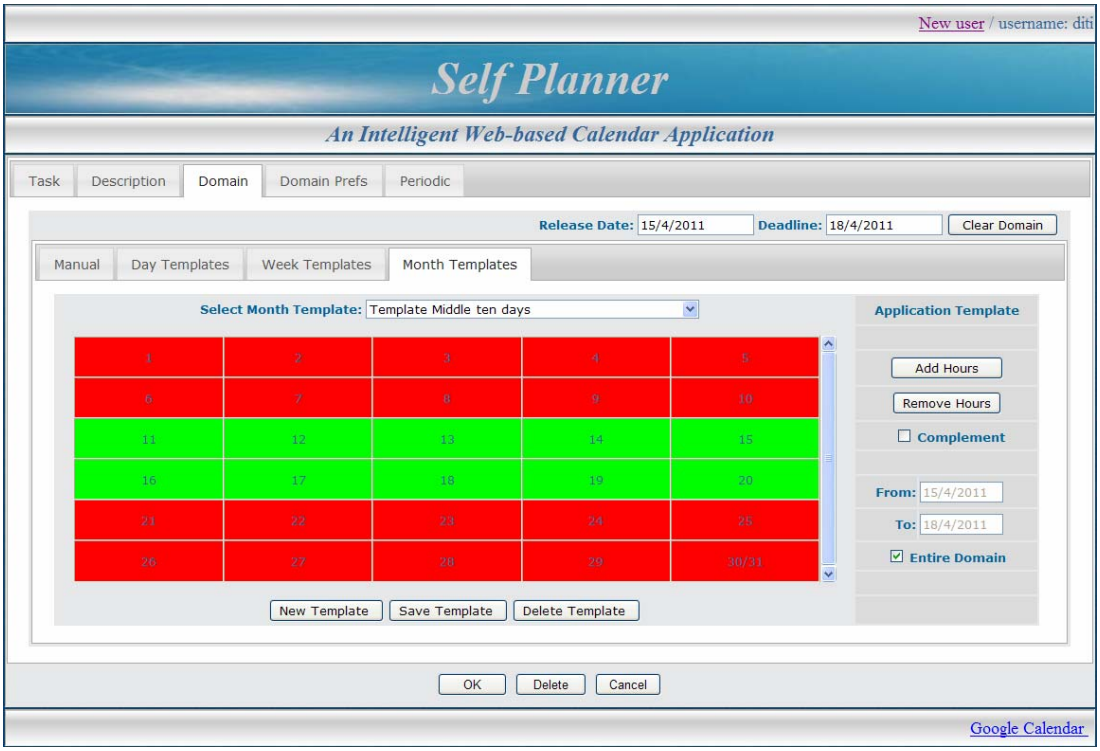
Ο χρήστης μπορεί να επιλέξει το Domain που θέλει, δηλαδή το χρόνο εκτέλεσης της εργασίας είτε Manual είτε μέσω Templates.

Στην καρτέλα «Manual» ο χρήστης επιλέγει τα ημίωρα ή τις ημέρες που θέλει να εκτελεστεί η εργασία. Το domain της εργασίας φαίνεται μόνο στην καρτέλα «Manual». Η καρτέλα «Manual» χωρίζεται στις καρτέλες «Day View», «Week View», «Month View», «Year View», οι οποίες δείχνουν το domain μιας εργασίας με διαφορετικό τρόπο.

- Η καρτέλα «*Day View*» εμφανίζει ανά ημέρα, κελιά που έχουν ημίωρα και τιμές από 00:00 έως 23:30. Ο χρήστης επιλέγει τα ημίωρα που θέλει να εκτελεστεί η εργασία, τα οποία γίνονται πράσινα. Υπάρχει ένα πλαίσιο κειμένου που εμφανίζει την τρέχουσα ημερομηνία και δύο κουμπιά «Previous Day» και «Next Day» που όταν τα επιλέξει ο χρήστης δείχνουν το domain της προηγούμενης ημέρας ή της επόμενης ημέρας αντίστοιχα.
- Η καρτέλα «*Week View*» εμφανίζει ανά εβδομάδα, κελιά που έχουν ημίωρα. Εμφανίζονται 7 στήλες όπου η κάθε στήλη είναι μία ημέρα της εβδομάδας με κελιά που έχουν ημίωρα και τιμές από 00:00 έως 23:30. Ο χρήστης επιλέγει τα ημίωρα που θέλει να εκτελεστεί η εργασία, τα οποία γίνονται πράσινα. Υπάρχουν δύο πλαίσια κειμένου που εμφανίζουν την πρώτη ημέρα και την τελευταία ημέρα της τρέχουσας εβδομάδας και δύο κουμπιά «Previous Week» και «Next Week» που όταν τα επιλέξει ο χρήστης δείχνουν το domain της προηγούμενης εβδομάδας ή της επόμενης εβδομάδας αντίστοιχα.
- Η καρτέλα «*Month View*» εμφανίζει ανά μήνα, κελιά που έχουν ημέρες και τιμές από 1 έως 31, ανάλογα τις ημέρες του μήνα. Τα κελιά μπορεί να έχουν κόκκινο, πράσινο ή γκρι χρώμα. Όταν ένα κελί έχει γκρι χρώμα, αυτό σημαίνει ότι υπάρχουν ημίωρα στη συγκεκριμένη ημέρα με κόκκινο και με πράσινο χρώμα. Όταν ένα κελί έχει κόκκινο χρώμα, αυτό σημαίνει ότι όλα τα ημίωρα στη συγκεκριμένη ημέρα έχουν κόκκινο χρώμα ενώ όταν ένα κελί έχει πράσινο χρώμα, αυτό σημαίνει ότι όλα τα ημίωρα στη συγκεκριμένη ημέρα έχουν πράσινο χρώμα. Ο χρήστης επιλέγει τα κελιά με τις ημέρες που θέλει να εκτελεστεί η εργασία, τα οποία γίνονται πράσινα. Ο χρήστης μπορεί να κάνει ένα κελί κόκκινο ή πράσινο χρώμα αλλά όχι γκρι το οποίο γίνεται αυτόματα

επειδή στις προηγούμενες καρτέλες υπάρχουν κόκκινα και πράσινα ημίωρα. Υπάρχει ένα πλαίσιο κειμένου που εμφανίζει τον τρέχοντα μήνα και δύο κουμπιά «Previous Month» και «Next Month» που όταν τα επιλέξει ο χρήστης δείχνουν το domain του προηγούμενου μήνα ή του επόμενου μήνα αντίστοιχα.

- Η καρτέλα «*Year View*» εμφανίζει ανά έτος, κελιά που έχουν ημέρες και τιμές από 1 έως 31, ανάλογα τις ημέρες του μήνα. Εμφανίζονται 12 γραμμές όπου η κάθε γραμμή είναι ένας μήνας του έτους. Τα κελιά μπορεί να έχουν κόκκινο, πράσινο ή γκρι χρώμα. Όταν ένα κελί έχει γκρι χρώμα, αυτό σημαίνει ότι υπάρχουν ημίωρα στη συγκεκριμένη ημέρα με κόκκινο και με πράσινο χρώμα. Όταν ένα κελί έχει κόκκινο χρώμα, αυτό σημαίνει ότι όλα τα ημίωρα στη συγκεκριμένη ημέρα έχουν κόκκινο χρώμα ενώ όταν ένα κελί έχει πράσινο χρώμα, αυτό σημαίνει ότι όλα τα ημίωρα στη συγκεκριμένη ημέρα έχουν πράσινο χρώμα. Ο χρήστης επιλέγει τα κελιά με τις ημέρες που θέλει να εκτελεστεί η εργασία, τα οποία γίνονται πράσινα. Ο χρήστης μπορεί να κάνει ένα κελί κόκκινο ή πράσινο χρώμα αλλά όχι γκρι το οποίο γίνεται αυτόματα επειδή στις προηγούμενες καρτέλες υπάρχουν κόκκινα και πράσινα ημίωρα. Υπάρχει ένα πλαίσιο κειμένου που εμφανίζει το τρέχον έτος και δύο κουμπιά «Previous Year» και «Next Year» που όταν τα επιλέξει ο χρήστης δείχνουν το domain του προηγούμενου έτους ή του επόμενου έτους αντίστοιχα.



Εικόνα 26: Οθόνη «New Task»: Καρτέλα «Domain - Month Templates»

Εκτός από την καρτέλα «Manual» υπάρχουν οι καρτέλες «Day Templates», «Week Templates», «Month Templates». Μερικές φορές η χειροκίνητη επεξεργασία – Manual δεν είναι τόσο αποτελεσματική και ο χρήστης μπορεί να χρησιμοποιήσει πρότυπα – Templates ώστε να ενημερώνεται αυτόματα ο τομέας – Domain, δηλαδή ο χρόνος εκτέλεσης της εργασίας.

Για κάθε είδος Templates υπάρχουν αρχικές τιμές. Την πρώτη φορά που μπαίνει ο χρήστης στο «New Task» αν δεν υπάρχουν Templates το πρόγραμμα τα δημιουργεί.

Για το «Day Templates» υπάρχουν οι τιμές:

- «Template Daily non sleep day»
- «Template Daily Lunch hours»
- «Template Daily All day»

Για το «Week Templates» υπάρχουν οι τιμές:

- «Template Store Working hours»
- «Template Office Working hours»

Για το «Month Templates» υπάρχουν οι τιμές:

- «Template First ten days»
- «Template Middle ten days»
- «Template Last ten days»

Ο χρήστης για να ορίσει ένα καινούργιο template, μπορεί να ξεκινήσει την αλλαγή ενός υπάρχοντος κάνοντας κλικ στα κελιά του και τελικά να το αποθηκεύσει χρησιμοποιώντας το κουμπί «Save Template» και δίνοντας ένα νέο όνομα. Εναλλακτικά, μπορεί να δημιουργήσει ένα template από την αρχή πατώντας το κουμπί «New Template». Ένα υπάρχον template μπορεί να το διαγράψει πατώντας το κουμπί «Delete Template».

- Η καρτέλα «*Day Templates*» εμφανίζει κελιά που έχουν ημίωρα και τιμές από 00:00 έως 23:30. Για να διαβάσει ο χρήστης ένα template επιλέγει από μία λίστα που εμφανίζει μόνο τα «Day Templates». Όταν ο χρήστης επιλέξει ένα template εμφανίζονται τα ημίωρα του, τα οποία έχουν πράσινο χρώμα.
- Η καρτέλα «*Week Templates*» εμφανίζει κελιά που έχουν ημίωρα. Εμφανίζονται 7 στήλες όπου η κάθε στήλη είναι μία ημέρα της εβδομάδας με κελιά που έχουν ημίωρα και τιμές από 00:00 έως 23:30. Για να διαβάσει ο χρήστης ένα template επιλέγει από μία λίστα που εμφανίζει μόνο τα «Week Templates». Όταν ο χρήστης επιλέξει ένα template εμφανίζονται τα ημίωρα του, τα οποία έχουν πράσινο χρώμα.
- Η καρτέλα «*Month Templates*» εμφανίζει κελιά που έχουν ημέρες και τιμές από 1 έως 31 που είναι οι ημέρες ενός μήνα. Για να διαβάσει ο χρήστης ένα template επιλέγει από μία λίστα που εμφανίζει μόνο τα «Month Templates». Όταν ο χρήστης επιλέξει ένα template εμφανίζονται οι ημέρες του, οι οποίες έχουν πράσινο χρώμα.

Για να εφαρμοστεί το template στον τομέα-domain της εργασίας ο χρήστης θα πρέπει να πατήσει το κουμπί «Add Hours» ή το κουμπί «Remove Hours». Πριν όμως από αυτή την επιλογή, έχει τη δυνατότητα να εφαρμόσει το template σε ολόκληρο τον τομέα ή μέρος αυτού. Για να εφαρμόσει το template σε ολόκληρο τον τομέα πρέπει το checkbox «Entire Domain» να είναι επιλεγμένο. Σε αντίθετη περίπτωση, ο χρήστης μπορεί να επιλέξει το χρονικό διάστημα κατά το οποίο το template θα εφαρμοστεί,

χρησιμοποιώντας τις ημερομηνίες «from» και «to». Οι ημερομηνίες «from» και «to» δεν μπορεί να είναι εκτός των ημερομηνιών «Release Date» και «Deadline».

Μετά την επιλογή του template για να το εφαρμόσει ο χρήστης στο τομέα-domain της εργασίας υπάρχουν 4 τρόποι:

1. Πρόσθεση των πράσινων κελιών του template στον τομέα: Ο χρήστης πατάει το κουμπί «Add Hours» ενώ το checkbox «Complement» δεν είναι επιλεγμένο.
2. Πρόσθεση των κόκκινων κελιών του template στον τομέα: Ο χρήστης πατάει το κουμπί «Remove Hours» ενώ το checkbox «Complement» δεν είναι επιλεγμένο.
3. Πρόσθεση των πράσινων κελιών του template στον τομέα: Ο χρήστης πατάει το κουμπί «Add Hours» ενώ το checkbox «Complement» είναι επιλεγμένο που σημαίνει ότι έγινε ανταλλαγή χρωμάτων.
4. Πρόσθεση των κόκκινων κελιών του template στον τομέα: Ο χρήστης πατάει το κουμπί «Remove Hours» ενώ το checkbox «Complement» είναι επιλεγμένο που σημαίνει ότι έγινε ανταλλαγή χρωμάτων.

Ο χρήστης μπορεί να εφαρμόσει πολλά templates στον τομέα της εργασίας αλλά δεν μπορεί να ακυρώσει κάποιο από αυτά. Μπορεί όμως να πατήσει το κουμπί «Clear Domain» και να τα καταργήσει όλα και να αρχίσει από την αρχή. Πριν ή μετά την εφαρμογή του template, ο χρήστης μπορεί να πάει στην καρτέλα «Manual» και να κάνει τις αλλαγές που θέλει.

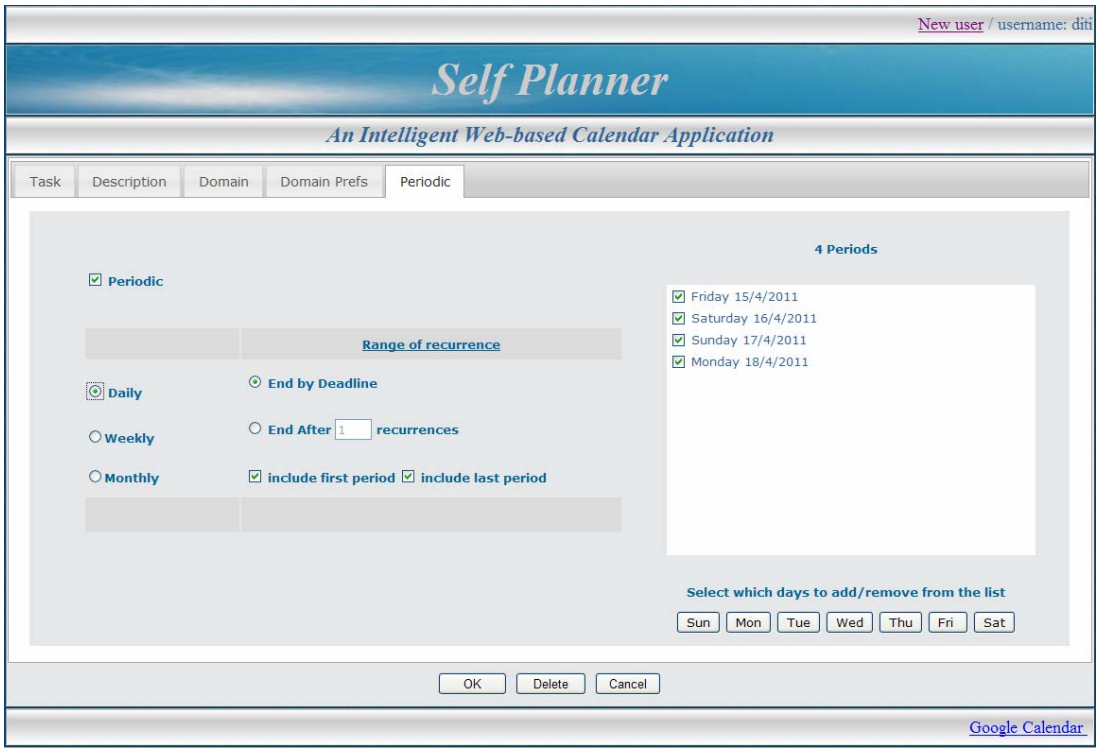
Τα templates εφαρμόζονται στον τομέα-domain της εργασίας και δεν χρειάζεται να αποθηκεύονται σαν πληροφορία στην εργασία.

Τα templates αποθηκεύονται στον πίνακα «template», όπου τα ημίωρα ή οι ημέρες, που έχουν πράσινο χρώμα, αποθηκεύονται σε ένα πεδίο χωρισμένα μεταξύ τους με κόμμα.

Ο τομέας-domain της εργασίας αποθηκεύεται στον πίνακα «task_domain» με τον ίδιο τρόπο όπως στον πίνακα «template», δηλαδή τα ημίωρα ή οι ημέρες, που έχουν πράσινο χρώμα, αποθηκεύονται σε ένα πεδίο χωρισμένα μεταξύ τους με

κόμμα. Για κάθε εργασία υπάρχουν τόσες εγγραφές στον πίνακα «task_domain» όσες είναι οι ημέρες που έχουν ημίωρα ή ημέρες με πράσινο χρώμα.

Στην καρτέλα «Domain Prefs» ο χρήστης επιλέγει την προτίμησή του για το πότε πρέπει να εκτελεστεί η εργασία στο χρονικό διάστημα που έχει επιλέξει στο «Domain». Το πεδίο pref στον πίνακα tasks παίρνει τη τιμή 0 αν ο χρήστης επιλέξει καμία προτίμηση – No preference, τη τιμή 1 αν ο χρήστης επιλέξει όσο το δυνατόν αργότερα – As later as possible ή τη τιμή -1 αν ο χρήστης επιλέξει όσο το δυνατόν νωρίτερα – As earlier as possible.



Εικόνα 27: Οθόνη «New Task»: Καρτέλα «Periodic»

Στην καρτέλα «Periodic» ο χρήστης επιλέγει αν θέλει η εργασία να είναι περιοδική, δηλαδή να επαναλαμβάνεται σε τακτά χρονικά διαστήματα.

Αρχικά όλα τα πεδία, εκτός από το «Periodic» είναι ανενεργά. Ο χρήστης επιλέγει το checkbox «Periodic» για να είναι η εργασία περιοδική και όλα τα πεδία ενεργοποιούνται ενώ στη λίστα «Periods» εμφανίζονται οι ημερήσιες περίοδοι.

Ο χρήστης μπορεί να επιλέξει η κάθε περίοδος να είναι Daily-ημερήσια ή weekly-εβδομαδιαία ή Monthly-μηνιαία. Σε όλες τις επιλογές η πρώτη περίοδος είναι η Release Date ενώ η τελευταία περίοδος είναι η Deadline.

Μόνο στην επιλογή «Daily», που κάθε περίοδος είναι και μία ημέρα, ενεργοποιούνται τα κουμπιά Sun, Mon, Tue, Wed, Thu, Fri, Sat. Ο χρήστης επιλέγει ποια μέρα από τις περιόδους θα αφαιρέσει ή θα προσθέσει. Παράδειγμα: αν ο χρήστης επιλέξει «Sun» τότε όλες οι Κυριακές θα αφαιρεθούν από τις περιόδους ενώ αν το επιλέξει ξανά τότε θα προστεθούν.

Στην επιλογή «Weekly» η κάθε περίοδος είναι μία εβδομάδα. Η εβδομάδα της πρώτης περιόδου ξεκινάει από το Release Date και τελειώνει στην Κυριακή, η τελευταία εβδομάδα ξεκινάει Δευτέρα και τελειώνει στο Deadline ενώ οι ενδιάμεσες εβδομάδες ξεκινούν Δευτέρα και τελειώνουν Κυριακή.

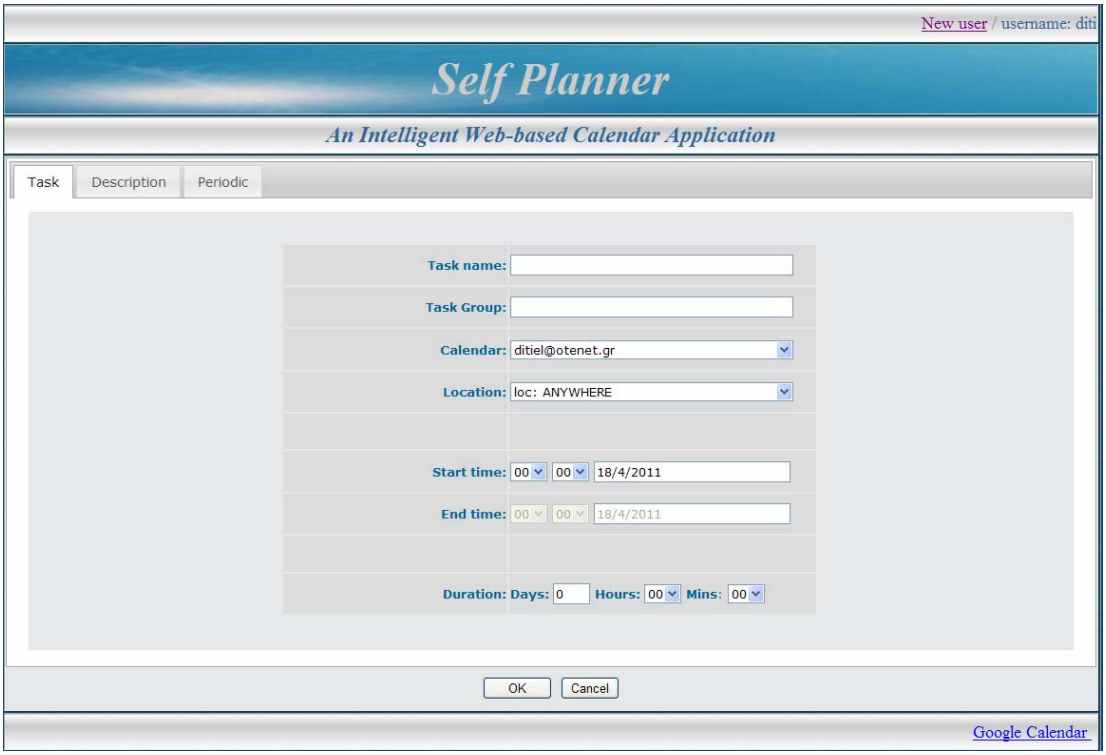
Στην επιλογή «Monthly» η κάθε περίοδος είναι ένας μήνας. Ο μήνας της πρώτης περιόδου ξεκινάει από το Release Date ενώ ο τελευταίος μήνας τελειώνει στο Deadline.

Όταν ο χρήστης επιλέξει το «End by Deadline», η τελευταία περίοδος είναι το Deadline. Όταν ο χρήστης επιλέξει το «End After recurrences» πληκτρολογεί τον αριθμό των περιόδων και το Deadline αλλάζει και παίρνει την τιμή της τελευταίας περιόδου.

Ο χρήστης επιλέγει το «Include first period» αν θέλει να συμπεριλαμβάνεται η πρώτη περίοδος ενώ επιλέγει το «Include last period» αν θέλει να συμπεριλαμβάνεται η τελευταία περίοδος.

Οι περίοδοι μιας εργασίας αποθηκεύονται στον πίνακα task_dates.

5.2.7. Οθόνη «Quick Insert»



Εικόνα 28: Οθόνη «Quick Insert»: Καρτέλα «Task»

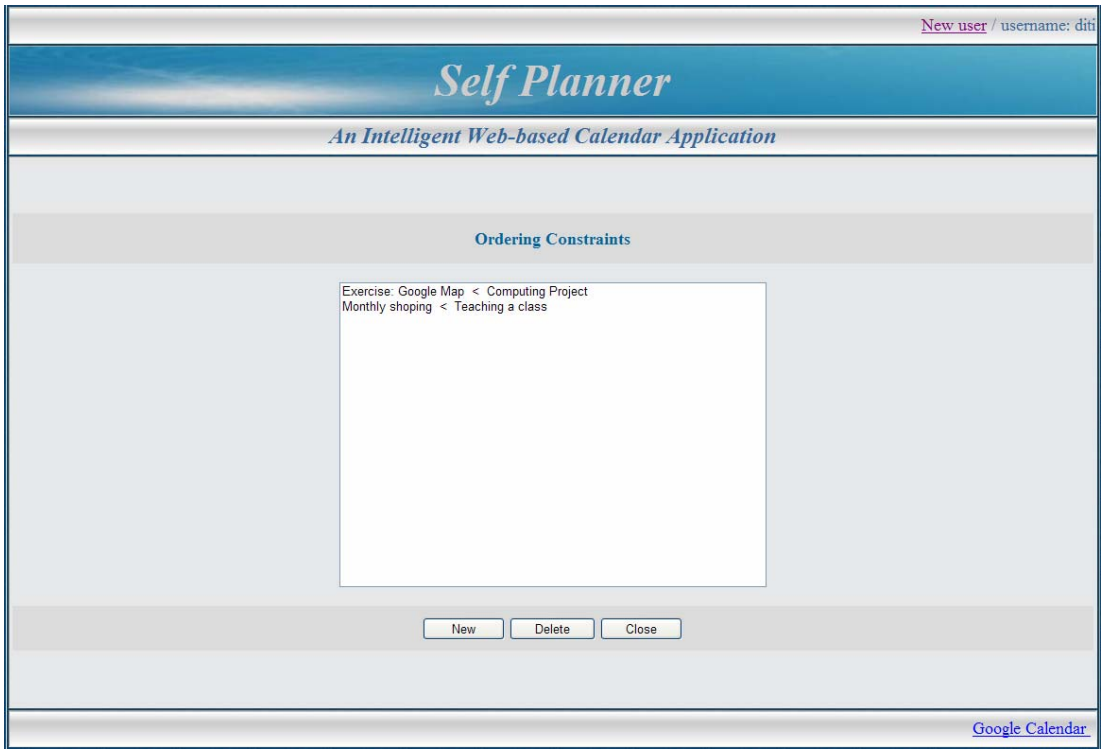
Στην οθόνη «Quick Insert» ο χρήστης εισάγει μία εργασία δίνοντας λιγότερες πληροφορίες. Τα στοιχεία που μπορεί ο χρήστης να καταχωρήσει σε μια εργασία χωρίζονται σε 3 καρτέλες. Οι καρτέλες «Description» και «Periodic» είναι ίδιες με την οθόνη «New Task» και δεν γίνεται ανάλυσή τους. Στη συνέχεια δίνεται επεξήγηση των πεδίων της καρτέλας «Task» μόνο αυτών που δεν υπάρχουν στην οθόνη «New Task».

Πίνακας 14: Τα πεδία της καρτέλας «Task»

Πεδίο	Περιγραφή
Start time	Η Start time που είναι η ημερομηνία έναρξης της εργασίας, αποτελείται από την ώρα, τα λεπτά και την ημερομηνία. Ο χρήστης μπορεί να επιλέξει ώρα από μια λίστα με τιμές από 00 ως 23 ή να επιλέξει λεπτά από μια λίστα με τιμές 00 ή 30. Όταν ο χρήστης κάνει κλικ στο πεδίο της ημερομηνίας που είναι ένα πλαίσιο κειμένου, εμφανίζεται ένα ημερολόγιο ανά μήνα (Datepicker) όπου μπορεί να επιλέξει την ημερομηνία που επιθυμεί ή να πληκτρολογήσει την ημερομηνία στο πλαίσιο κειμένου. Όταν ο χρήστης δώσει τιμές στο Start time, το πρόγραμμα προσθέτει τις αντίστοιχες τιμές του Duration με τις αντίστοιχες τιμές του Start time και υπολογίζεται το End time. Το ίδιο ισχύει όταν ο χρήστης δώσει τιμές στο Duration, δηλαδή υπολογίζεται το End time.
End time	Η End time αποτελείται από την ώρα, τα λεπτά και την ημερομηνία. Και τα 3 πεδία είναι ανενεργά και υπολογίζονται.

Ο τομέας-domain δηλαδή ο χρόνος που είναι διαθέσιμος για την εκτέλεση της εργασίας δεν τον επιλέγει ο χρήστης όπως στην οθόνη «New Task» αλλά υπολογίζεται αυτόματα. Το domain είναι οι ώρες από την ημερομηνία, την ώρα και τα λεπτά του Start time μέχρι την ημερομηνία, την ώρα και τα λεπτά του End time. Όταν η εργασία δεν είναι περιοδική το End time είναι η ημερομηνία λήξης – Deadline της εργασίας. Όταν η εργασία είναι περιοδική, το domain που υπολογίστηκε επαναλαμβάνεται σε κάθε περίοδο και η ημέρα της τελευταίας περιόδου είναι η ημερομηνία λήξης – Deadline της εργασίας.

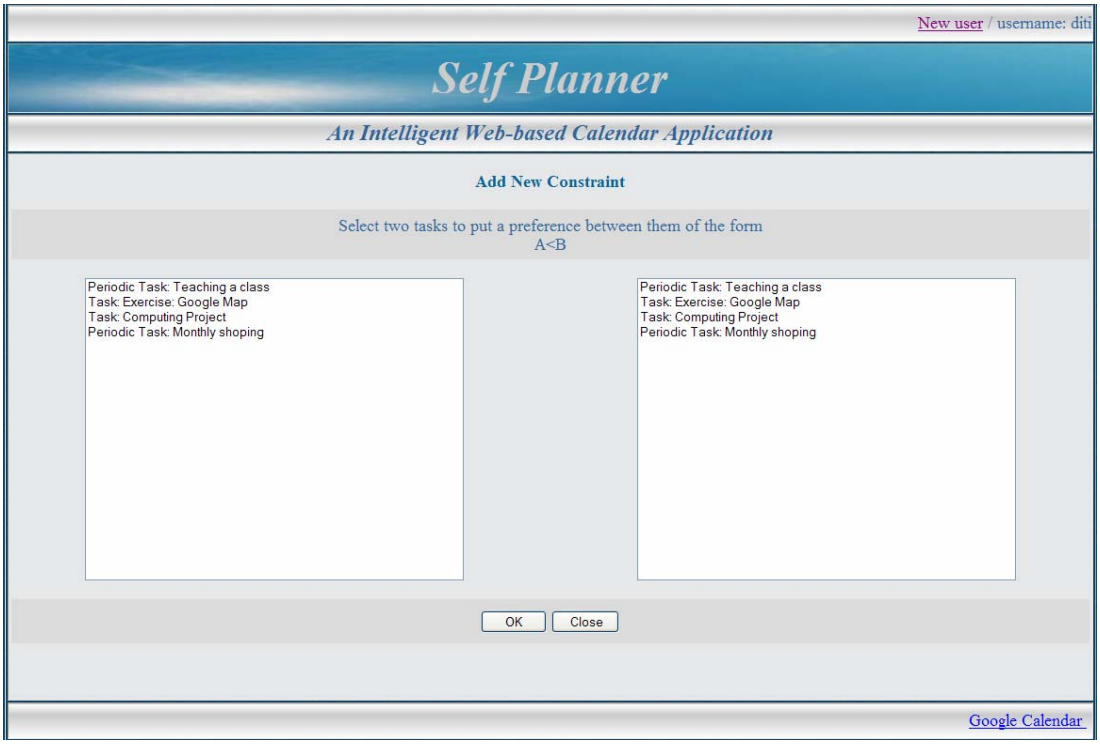
5.2.8. Οθόνες «Constraint» και «New_Constraint»



Εικόνα 29: Οθόνη «Constraint»

Στην οθόνη «Constraint» εμφανίζονται οι περιορισμοί, που στη περίπτωση αυτή είναι ταξινομήσεις, που έχουν γίνει μεταξύ των εργασιών ανά ζεύγη. Οι ταξινομήσεις γίνονται ώστε με τη σειρά αυτή να συμμετέχουν οι εργασίες στην επίλυση-solve. Αυτό σημαίνει ότι κάποια εργασία πρέπει να εκτελεστεί πριν από κάποια άλλη.

Ο χρήστης μπορεί να επιλέξει έναν περιορισμό και να πατήσει το κουμπί «Delete» για να τον διαγράψει ή να πατήσει το κουμπί «New» για να καταχωρήσει έναν καινούργιο περιορισμό. Όταν ο χρήστης πατήσει το κουμπί «New», εμφανίζεται η οθόνη «New_Constraint».



Εικόνα 30: Οθόνη «New_ Constraint »

Στην οθόνη «New_Constraint» εμφανίζονται δύο λίστες, όπου και οι δύο εμφανίζουν τις εργασίες-tasks που δεν είναι ολοκληρωμένες. Ο χρήστης επιλέγει μία εργασία από την αριστερή λίστα και μία εργασία από την δεξιά λίστα, που δεν επιτρέπεται να είναι ίδια, και πατάει το πλήκτρο «OK» για να γίνει η καταχώρηση. Ακόμη δεν επιτρέπεται να επιλεγούν δύο εργασίες που είναι περιοδικές αλλά διαφορετικού τύπου (π.χ. ημερήσια και μηνιαία) καθώς επίσης δεν επιτρέπεται να επιλεγούν δύο εργασίες όταν η μία είναι περιοδική και η άλλη δεν είναι. Μετά την επιλογή και την καταχώρηση, σημαίνει ότι η εργασία από την αριστερή λίστα είναι μικρότερη (δηλαδή θα εκτελεστεί πριν) από την εργασία από την δεξιά λίστα.

Οι περιορισμοί των εργασιών καταχωρούνται στον πίνακα «constraints»

5.2.9. Οθόνη «Solve»

Στην οθόνη «Main», όταν ο χρήστης επιλέξει το κουμπί «Solve» δεν εμφανίζεται καινούργια οθόνη αλλά καλείται ένα πρόγραμμα για την επίλυση-solve των εργασιών. Αρχικά εμφανίζεται ένα παράθυρο διαλόγου με το μήνυμα «Are you sure you want to solve the tasks?». Αν ο χρήστης επιλέξει «OK» τότε εκτελείται η επίλυση των εργασιών.

Διαβάζονται και επιλύονται οι εργασίες που δεν έχουν ολοκληρωθεί και η ημερομηνία έναρξής τους είναι μεγαλύτερη ή ίση της τρέχουσας ημερομηνίας ή οι εργασίες που δεν έχουν ολοκληρωθεί, η ημερομηνία έναρξης τους είναι μικρότερη της τρέχουσας ημερομηνίας και η ημερομηνία λήξης τους είναι μεγαλύτερη ή ίση της τρέχουσας ημερομηνίας. Οι εργασίες που η ημερομηνία έναρξής τους είναι μεγαλύτερη ή ίση της τρέχουσας ημερομηνίας είναι καινούργιες εργασίες. Οι εργασίες που η ημερομηνία έναρξής τους είναι μικρότερη της τρέχουσας ημερομηνίας και η ημερομηνία λήξης τους είναι μεγαλύτερη ή ίση της τρέχουσας ημερομηνίας είναι παλιές εργασίες που ίσως να έχουν εκτελεστεί και επειδή λήγουν από την τρέχουσα ημερομηνία και μετά, εκτελείται ξανά ο τομέας-domain της εργασίας που η ημερομηνία είναι μεγαλύτερη ή ίση της τρέχουσας ημερομηνίας.

Στη συνέχεια το πρόγραμμα διαβάζει από τον πίνακα constraints τις ταξινομήσεις των εργασιών και παίρνει μόνο εκείνες στις οποίες και οι δύο εργασίες βρίσκονται στη λίστα που πρόκειται να επιλυθούν. Ταξινομούνται οι εργασίες σύμφωνα με τον πίνακα constraints και τοποθετούνται οι κωδικοί τους σε έναν πίνακα. Για τις εργασίες που δεν εμφανίζονται στον πίνακα constraints, οι κωδικοί τους τοποθετούνται στη συνέχεια στον ίδιο πίνακα, με τη σειρά που διαβάζονται ώστε όλες οι εργασίες να επιλυθούν. Αν μια εργασία είναι περιοδική τότε επαναλαμβάνεται τόσες φορές στον πίνακα όσες είναι οι περίοδοι που έχουν τομέα.

Στην πρώτη θέση του πίνακα τοποθετείται το εικονικό task με κωδικό 0. Το εικονικό task χρησιμοποιείται για να δηλωθεί η τοποθεσία-location που βρίσκεται ο χρήστης κατά τη χρονική στιγμή εκκίνησης της επίλυσης.

Στη συνέχεια το πρόγραμμα διαβάζει τον πίνακα με τους κωδικούς εργασιών που πρόκειται να επιλυθούν και δημιουργεί και ενημερώνει ένα αρχείο κειμένου με όνομα το όνομα του χρήστη – username και επέκταση .ecl, δηλαδή «username».ecl. Αν π.χ. το όνομα του χρήστη είναι «diti» τότε το αρχείο που δημιουργείται είναι το

diti.ecl. Κάθε φορά που διαβάζεται ένας κωδικός εργασίας διαβάζονται και οι υπόλοιπες πληροφορίες από τους πίνακες της βάσης δεδομένων και ενημερώνουν το αρχείο «username».ecl.

Στον κώδικα του προγράμματος, ανοίγει το αρχείο «username».ecl για γράψιμο με την εντολή fopen(). Αν το αρχείο δεν υπάρχει, δημιουργείται. Για την εγγραφή στο αρχείο, χρησιμοποιείται η εντολή fprintf(). Στη συνέχεια δίνεται ένα μικρό κομμάτι κώδικα που αφορά τις εντολές αυτές:

```
$filename=$username;
$filename = $filename.".ecl";
$fp = fopen($filename,"w");
fprintf($fp,"nof_tasks(%d).\n", count($sortasks));
```

Στη συνέχεια δίνεται η δομή ενός αρχείου «username».ecl:

```
nof_tasks(3).
task(id(0), task_locs(1), loc([2]), dur(1,1,0.000000,0), smin(1), smax(1),
prox_cons(0,2147483647), prox_prefs(0,0.000000,2147483647,0.000000),
temporal_pref(0,0,1.000000), utility(50.000000), utilization(1.000000), domain([0..1])).
task(id(1), task_locs(3), loc([0,2,3]), dur(12,12,0.000000,1), smin(2), smax(6),
prox_cons(2,2147483647), prox_prefs(0,0.000000,2147483647,0.000000),
temporal_pref(0,0,1.000000), utility(5.000000), utilization(1.000000), domain([16..34, 4..82])).
task(id(2), task_locs(1), loc([1]), dur(20,20,0.000000,0), smin(20), smax(20),
prox_cons(0,2147483647), prox_prefs(0,0.000000,2147483647,0.000000),
temporal_pref(0,0,1.000000), utility(5.000000), utilization(1.000000), domain([130..139,
159..184, 379..476])).
nof_locations(3).
loc_dist(0,0,0).
loc_dist(0,1,0).
loc_dist(0,2,0).
loc_dist(1,0,0).
loc_dist(1,1,0).
loc_dist(1,2,1).
loc_dist(2,0,0).
loc_dist(2,1,1).
loc_dist(2,2,0).
ordering_constraints(1).
before(1,2).
min_dist_constraints(0).
max_dist_constraints(0).
implication_constraints(0).
ordering_preferences(0).
min_dist_preferences(0).
max_dist_preferences(0).
implication_preferences(0).
```

Πίνακας 15: Τα πεδία του αρχείου «username».ecl

Πεδίο	Περιγραφή
nof_tasks(8)	Η πρώτη γραμμή του αρχείου έχει τη λέξη nof_tasks(8) όπου στην παρένθεση είναι το πλήθος των εργασιών που είναι για επίλυση. Στο παράδειγμα έχουμε 8 εργασίες. Μετά το πλήθος των εργασιών εμφανίζεται σε κάθε γραμμή και μία εργασία με τις πληροφορίες της, όπου οι πληροφορίες εμφανίζονται μέσα στη λέξη task(). Στη πρώτη θέση είναι το εικονικό task.
Παράδειγμα εργασίας-task: task(id(1), task_locs(3), loc([0,2,3]), dur(12,12,0.000000,1), smin(2), smax(6), prox_cons(2,2147483647), prox_prefs(0,0.000000,2147483647,0.000000), temporal_pref(0,0,1.000000), utility(5.000000), utilization(1.000000), domain([16..34, 64..82])).	
id(1)	Είναι ο κωδικός του task. Εδώ ο κωδικός είναι 1. Ο κωδικός είναι ένας αύξοντας αριθμός που έχει να κάνει με τη θέση που εμφανίζεται στο αρχείο.
task_locs(3)	Είναι το πλήθος των τοποθεσιών-locations που έχει ένα task, δηλαδή σε ποιες τοποθεσίες μπορεί να εκτελεστεί. Όταν το πλήθος των locations είναι μεγαλύτερο από 1, αυτό σημαίνει ότι η εργασία έχει κλάση τοποθεσιών – class locations, δηλαδή πολλές τοποθεσίες και μπορεί να γίνει επιλογή για να εκτελεστεί η εργασία.
loc([0,2,3])	Είναι οι κωδικοί των locations.
dur(12,12,0.000000,1)	Η πρώτη τιμή είναι η ελάχιστη διάρκεια του task ενώ η δεύτερη τιμή είναι η μέγιστη

	διάρκεια του task. Σε αυτή την έκδοση του SelfPlanner δεν έχουμε ελάχιστη και μέγιστη διάρκεια αλλά μία διάρκεια και οι δύο τιμές συμπίπτουν. Η διάρκεια-duration του task είναι σε πλήθος ημίωρων. Εδώ το πλήθος των ημίωρων που διαρκεί η εργασία είναι 12. Το πρόγραμμα μετατρέπει τις ημέρες, τις ώρες και τα λεπτά της διάρκειας-duration του task σε ημίωρα και τα προσθέτει για να βρει το πλήθος των ημίωρων. Η τρίτη τιμή είναι το extra utility που παίρνει ο χρήστης αν πετύχει τη μέγιστη διάρκεια. Εδώ η τιμή είναι 0.000000 αφού δεν υπάρχει μέγιστη διάρκεια. Η τέταρτη τιμή δηλώνει αν το task είναι διακοπτόμενο-interruptible και παίρνει την τιμή 1 ή αν το task δεν είναι διακοπτόμενο και παίρνει την τιμή 0.
smin(2), smax(6)	Είναι το πλήθος των ημίωρων στο ελάχιστο χρονικό διάστημα-minimum duration και στο μέγιστο χρονικό διάστημα-maximum duration που έχει ένα task που είναι διακοπτόμενο. Εδώ το πλήθος των ημίωρων στο minimum duration είναι 2 ενώ το πλήθος των ημίωρων στο maximum duration είναι 6. Αν το task δεν είναι διακοπτόμενο τότε το dur=smin=smax. Το πλήθος των ημίωρων υπολογίζεται όπως αναφέρθηκε παραπάνω.
prox_cons(2,2147483647)	Η πρώτη τιμή είναι η ελάχιστη απόσταση-minimum distance σε ημίωρα ενός διακοπτόμενου task. Δηλαδή πόσα ημίωρα απέχουν μεταξύ τους τα μέρη της εργασίας που

	διακόπτεται. Εδώ το πλήθος των ημίωρων είναι 2. Η δεύτερη τιμή είναι η μέγιστη απόσταση ενός διακοπτόμενου task. Σε αυτή την έκδοση του SelfPlanner δεν έχουμε μέγιστη απόσταση και δίνουμε τη σταθερή τιμή 2147483647.
prox_prefs (0,0.000000,2147483647,0.000000)	Σε αυτή την έκδοση του SelfPlanner δεν χρησιμοποιείται και δίνουμε πάντα αυτές τις τιμές.
temporal_pref(0,0,1.000000)	Η πρώτη τιμή είναι η προτίμηση του χρήστη, δηλαδή η τιμή του πεδίου pref στον πίνακα tasks, για το πότε πρέπει να εκτελεστεί η εργασία στο χρονικό διάστημα που έχει επιλέξει στο domain. Παίρνει τη τιμή 0 αν ο χρήστης έχει επιλέξει καμία προτίμηση – No preference, τη τιμή 1 αν ο χρήστης έχει επιλέξει όσο το δυνατόν αργότερα – As later as possible ή τη τιμή -1 αν ο χρήστης έχει επιλέξει όσο το δυνατόν νωρίτερα – As earlier as possible. Η δεύτερη τιμή είναι 0 και η τρίτη τιμή είναι 1.000000.
utility(5.000000), utilization(1.000000)	Δίνονται αυτές οι σταθερές τιμές.
domain([16..34, 64..82])	Στην καταχώρηση μιας εργασίας, ο χρήστης επιλέγει τα ημίωρα, δηλαδή το domain που μπορεί να εκτελεστεί η εργασία. Εδώ το domain εμφανίζεται με διαφορετικό τρόπο. Ο υπολογισμός του domain έχει σχέση με την ημέρα και την ώρα που ο χρήστης επιλέγει την επίλυση-solve. Κατά συνέπεια το domain κάθε

	φορά αλλάζει ανάλογα πότε γίνεται η επίλυση. Ο υπολογισμός γίνεται ως εξής: γίνεται μέτρηση από το πρώτο ημίωρο μετά το «τώρα», όπου το «τώρα» είναι η τρέχουσα ημέρα και ώρα. Αν π.χ. τώρα είναι 12:45 που ο χρήστης επέλεξε solve, η αρίθμηση ξεκινάει από το επόμενο ημίωρο που είναι το 13:00. Η αρίθμηση ξεκινάει από το 0, δηλαδή το ημίωρο 13:00 είναι το 0, το 13:30 είναι το 1, το 14:00 είναι το 2 κ.λ.π. Αν το ημίωρο αυτό, εδώ 13:00 το οποίο έχει αρίθμηση 0, είναι μέσα στο domain της εργασίας, τότε και το πρώτο διάστημα του domain θα ξεκινά με το 0. Αν το ημίωρο αυτό, δεν είναι μέσα στο domain της εργασίας, τότε συνεχίζουμε την αρίθμηση και όταν φτάσουμε στο πρώτο ημίωρο που ανήκει στο domain, παίρνουμε αυτή την αρίθμηση και τη βάζουμε στο πρώτο διάστημα του domain. Η αρίθμηση συνεχίζεται σε όλες τις ημέρες που μπορεί να έχει μία εργασία. Με αυτόν τον τρόπο δημιουργείται το domain με τα διαστήματα του και το ίδιο ημίωρο πρέπει να έχει την ίδια αρίθμηση σε όλες τις εργασίες που εκτελούνται. Μετά την εμφάνιση των εργασιών, υπάρχουν πληροφορίες για τις τοποθεσίες-locations που έχουν οι εργασίες.
nof_locations(4)	Είναι το πλήθος των τοποθεσιών-locations των εργασιών που είναι για επίλυση. Στο παράδειγμα έχουμε 4 τοποθεσίες. Μετά το πλήθος των τοποθεσιών εμφανίζεται σε κάθε γραμμή η χρονική απόσταση σε ημίωρα μεταξύ

	των τοποθεσιών.
loc_dist(1,3,2).	Η πρώτη τιμή είναι κωδικός τοποθεσίας, η δεύτερη τιμή είναι κωδικός τοποθεσίας και η τρίτη τιμή είναι η χρονική απόσταση σε ημίωρα μεταξύ των δύο τοποθεσιών. Στο παράδειγμα, ο χρόνος που χρειάζεται ο χρήστης για να πάει από την τοποθεσία με κωδικό 1 στην τοποθεσία με κωδικό 3 είναι 2 ημίωρα. Για κάθε τοποθεσία υπάρχουν όλοι οι συνδυασμοί με τις τοποθεσίες που εμφανίζονται στις συγκεκριμένες εργασίες. Η τοποθεσία ANYWHERE με κωδικό 0 έχει χρονική απόσταση 0 με τις υπόλοιπες τοποθεσίες. Οι χρονικές αποστάσεις μεταξύ των τοποθεσιών που βρίσκονται στον πίνακα locations_duration είναι σε λεπτά. Για να υπολογιστούν τα ημίωρα ακολουθείται η εξής διαδικασία: Αν τα λεπτά είναι λιγότερα από 5 τότε δεν υπολογίζεται ημίωρο. Αν τα λεπτά είναι από 5 λεπτά και πάνω και λιγότερα από 30 λεπτά, τότε υπολογίζεται 1 ημίωρο. Στις υπόλοιπες περιπτώσεις τα λεπτά διαιρούνται με το 30 για να βρεθεί το πλήθος των ημίωρων και γίνεται στρογγυλοποίηση προς τα πάνω. Παράδειγμα: αν η χρονική απόσταση είναι 45 λεπτά, αυτό σημαίνει ότι υπάρχουν 2 ημίωρα.
ordering_constraints(2)	Είναι το πλήθος των ταξινομήσεων-constraints των εργασιών που είναι για επίλυση. Στο παράδειγμα έχουμε 2 constraints. Στη συνέχεια ακολουθούν οι ταξινομήσεις με τη λέξη before().

before(4,7)	Και οι δύο αριθμοί είναι κωδικοί εργασίας και δείχνει ποιος κωδικός είναι πριν από τον άλλον. Στο παράδειγμα, ο κωδικός εργασίας 4 εμφανίζεται, δηλαδή θα εκτελεστεί, πριν από τον κωδικό εργασίας 7.
min_dist_constraints(0). max_dist_constraints(0). implication_constraints(0). ordering_preferences(0). min_dist_preferences(0). max_dist_preferences(0). implication_preferences(0).	Σε αυτή την έκδοση του SelfPlanner δεν χρησιμοποιούνται και δίνονται πάντα αυτές οι τιμές.

Όπως αναφέρθηκε παραπάνω, αν μια εργασία είναι περιοδική τότε επαναλαμβάνεται η εργασία τόσες φορές όσες είναι οι περίοδοι της εργασίας. Στο αρχείο «username».ecl, η κάθε περίοδος της ίδιας εργασίας έχει διαφορετικό κωδικό με τα υπόλοιπα στοιχεία ίδια, εκτός από το domain που αλλάζει αφού έχει να κάνει με διαφορετικές χρονικές περιόδους.

Μετά τη δημιουργία του αρχείου «username».ecl, εκτελείται το πρόγραμμα swo.exe για να επιλύσει τις εργασίες ανάλογα με τις ταξινομήσεις και τους χρονικούς περιορισμούς και να τις τοποθετήσει σε μια σειρά για να είναι έτοιμες να εισαχθούν στο Calendar. Το πρόγραμμα swo.exe δέχεται ως παράμετρο το αρχείο «username».ecl, διαβάζει τα στοιχεία του και τα αποτελέσματά του στέλνονται σε ένα αρχείο με όνομα «username».solve. Όπως με το αρχείο .ecl, αν π.χ. το όνομα του χρήστη είναι «diti» τότε το αρχείο που δημιουργείται είναι το diti.solve.

Η εκτέλεση των εξωτερικών εντολών μέσα στο πρόγραμμα γίνεται με την εντολή shell_exec:

```
$fsolve = $username.".solve";
$fstr= " swo ".$filename." > ".$fsolve;
$rescom = shell_exec($fstr);
```

Στη συνέχεια δίνεται η δομή ενός αρχείου «username».solve:

```
NotScheduled(1)
Tasks(4)
TaskID(0)
SubtasksNo(0)
TaskID(1)
SubtasksNo(3)
[16, 6]@0
[24, 4]@0
[64, 2]@0
TaskID(2)
SubtasksNo(1)
[159, 20]@2
TaskID(3)
SubtasksNo(1)
[60, 4]@1
```

Πίνακας 16: Τα πεδία του αρχείου «username».solve

Πεδίο	Περιγραφή
NotScheduled(1)	Είναι το πλήθος των εργασιών που δεν επιλύθηκαν. Στο παράδειγμα, το πλήθος των εργασιών που δεν επιλύθηκαν είναι 1.
Tasks(4)	Είναι το πλήθος των εργασιών που επιλύθηκαν. Στο παράδειγμα, το πλήθος των εργασιών που επιλύθηκαν είναι 4.
TaskID(1)	Είναι ο κωδικός του task.
SubtasksNo(3)	Είναι το πλήθος των τμημάτων αν η εργασία είναι διακοπτόμενη, διαφορετικά είναι 1 αν η εργασία δεν είναι διακοπτόμενη. Αν η τιμή είναι 0, αυτό σημαίνει ότι η εργασία δεν έχει επιλυθεί.

[159, 20]@2	Η πρώτη τιμή είναι ο χρόνος έναρξης ενός τμήματος του domain σε ημίωρα, η δεύτερη τιμή είναι η διάρκεια σε ημίωρα και η τρίτη τιμή είναι ο κωδικός τοποθεσίας-Location που πρόκειται να εκτελεστεί η εργασία. Στο παράδειγμα, το διάστημα του domain ξεκινάει στο 159 ημίωρο, έχει διάρκεια 20 ημίωρα και εκτελείται στο location με κωδικό 2.

Μετά τη δημιουργία του αρχείου «username».solve, το πρόγραμμα ανοίγει το αρχείο για διάβασμα με την εντολή $fs = fopen($fsolve, "r") και διαβάζει το αρχείο, με την εντολή $line = fgets($fs), για να ενημερώσει το Google Calendar με τις εργασίες. Πριν γίνει η ενημέρωση στο Google Calendar, το πρόγραμμα διαγράφει τα γεγονότα στο Calendar από την τρέχουσα ημερομηνία και μετά, αλλά μόνο των γεγονότων που έχουν δημιουργηθεί μέσα από την εφαρμογή SelfPlanner. Το πρόγραμμα διαβάζει τα στοιχεία από το αρχείο «username».solve και μαζί με τους πίνακες της βάσης δεδομένων δημιουργεί γεγονότα-events στο Calendar με τα εξής πεδία:

- *Πεδίο title*: έχει το όνομα του task, τη λέξη «Period:» και τον αριθμό περιόδου αν η εργασία είναι περιοδική, τη λέξη «Part:» και τον αριθμό του μέρους που έχει διασπαστεί η εργασία αν είναι διακοπτόμενη και τη λέξη «Location:» και την περιγραφή της τοποθεσίας-location.
- *Πεδίο content*: έχει τη πρόταση «Automatically created by Self Planner» και την περιγραφή-description της εργασίας.
- *Πεδίο where*: έχει τις συντεταγμένες της τοποθεσίας και την περιγραφή της τοποθεσίας. Όταν ο χρήστης, μέσα από το Google Calendar, σε ένα γεγονός κάνει κλικ στις συντεταγμένες, ανοίγει η σελίδα του Google Map και δείχνει ένα marker με την περιγραφή της τοποθεσίας στο συγκεκριμένο σημείο.
- *Πεδίο when*: έχει την ημερομηνία και ώρα έναρξης και την ημερομηνία και ώρα λήξης της εργασίας.

- *Πεδίο calendar*: έχει το ημερολόγιο-calendar της εργασίας. Το γεγονός δημιουργείται στο συγκεκριμένο calendar που έχει η εργασία.
- *Πεδίο user*: έχει το χρήστη που δημιούργησε το γεγονός στο Google Calendar που είναι ο κωδικός google account στον πίνακα users.

Η κλάση Zend_Gdata_Calendar χρησιμοποιείται για να προβάλει, να δημιουργήσει, να ενημερώσει και να διαγράψει τα γεγονότα στην online υπηρεσία Google Calendar. Πρώτα απ' όλα για να δημιουργηθεί μια σύνδεση με τους διακομιστές της Google πρέπει να δημιουργηθεί ένας πελάτης-client και γι' αυτόν τον πελάτη να δεσμευτεί μια υπηρεσία Zend_Gdata_Calendar.

Κατά τη δημιουργία γεγονότος στο Calendar, που γίνεται με το Google API insertEvent(), δημιουργείται το id του γεγονότος:

http://www.google.com/calendar/feeds/ditiel%40otenet.gr/private/full/93t46mp96nghh2v0i3or2sevro. Μετά τη δημιουργία γεγονότος στο Calendar, δημιουργείται και αντίστοιχη εγγραφή στον πίνακα task_googlecalendar, όπου μαζί με τις πληροφορίες του γεγονότος, εισάγεται και το id του γεγονότος που είναι μοναδικό. Στη συνέχεια δίνεται ένα μικρό κομμάτι κώδικα με την εντολή insertEvent() που δημιουργεί ένα γεγονός στο συγκεκριμένο subcalendar που ανήκει η εργασία και την ανάκτηση του id με την ιδιότητα text:

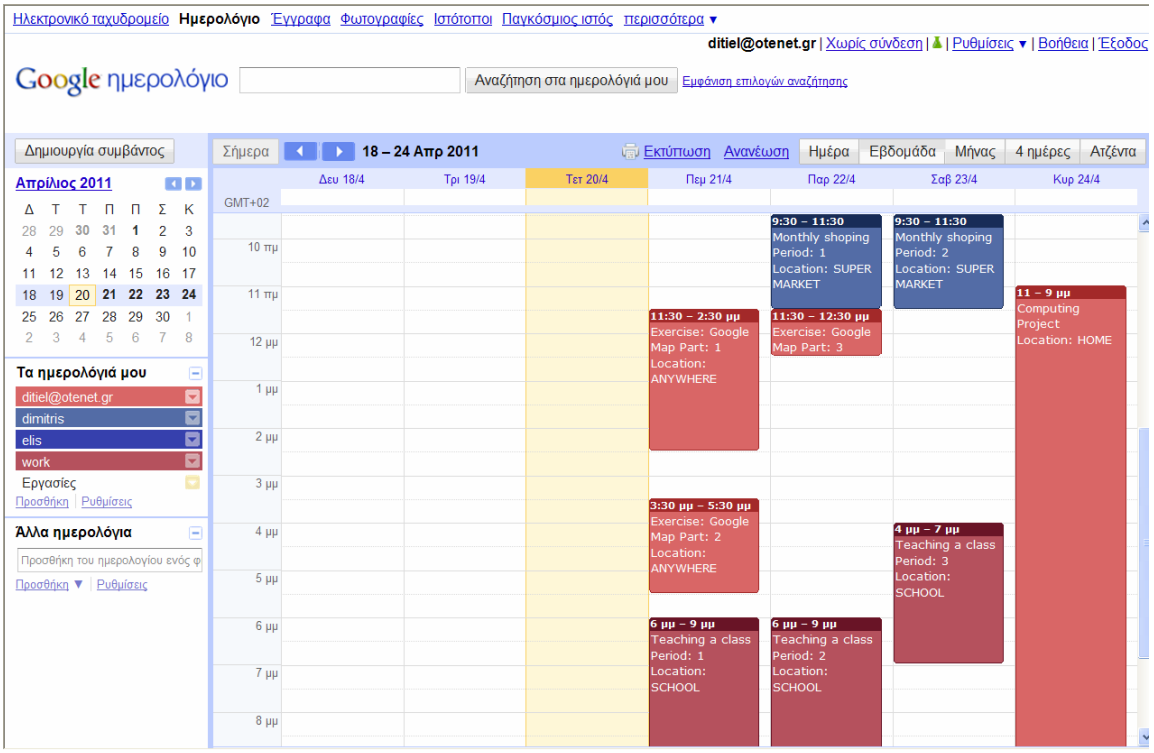
```
$createdEntry = $gc->insertEvent($newEntry, $appCallUri);
return $createdEntry->id->text;
```

Όπως έχει αναφερθεί παραπάνω, το πρόγραμμα διαγράφει τα γεγονότα στο Calendar από την τρέχουσα ημερομηνία και μετά, αλλά μόνο αυτών που έχουν δημιουργηθεί μέσα από την εφαρμογή SelfPlanner του συγκεκριμένου χρήστη. Όταν το πρόγραμμα διαβάζει τα γεγονότα από το Calendar, με το id του γεγονότος διαβάζει την αντίστοιχη εγγραφή από τον πίνακα task_googlecalendar για να διαπιστώσει αν υπάρχει ή όχι. Αν υπάρχει η εγγραφή στον πίνακα task_googlecalendar αυτό σημαίνει ότι το γεγονός έχει δημιουργηθεί μέσα από την εφαρμογή και μπορεί να τη διαγράψει. Για τη διαγραφή γεγονότος χρησιμοποιείται το Google API delete(). Όταν διαγράφεται το γεγονός από το Calendar, διαγράφεται και από τον πίνακα

task_googlecalendar. Στη συνέχεια δίνεται ένα μικρό κομμάτι κώδικα όπου διαβάζονται τα γεγονότα και γίνεται διαγραφή τους με την εντολή delete():

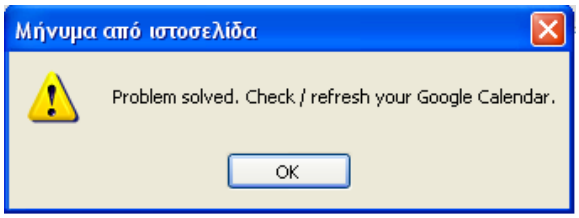
```
foreach ($eventFeed as $event)
{
        $event->delete();
}
```

Στην παρακάτω εικόνα φαίνονται τα γεγονότα που ενημερώθηκαν στο Google Calendar:

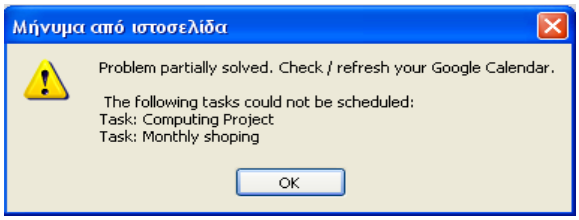


Εικόνα 31: «Google Calendar»

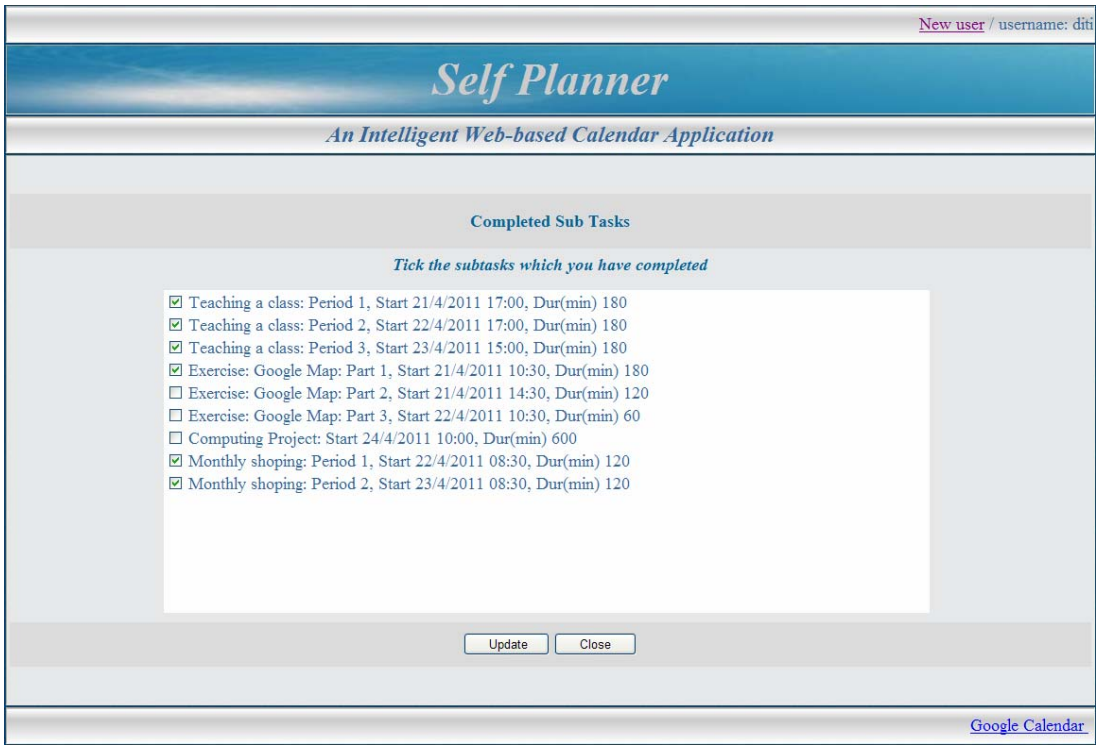
Μετά τη δημιουργία των γεγονότων στο Google Calendar, το πρόγραμμα «Solve» εμφανίζει μηνύματα ανάλογα τα αποτελέσματα. Αν όλες οι εργασίες επιλύθηκαν τότε εμφανίζει το παρακάτω πλαίσιο διαλόγου:



Αν κάποιες εργασίες δεν επιλύθηκαν τότε εμφανίζει το παρακάτω πλαίσιο διαλόγου που δείχνει ποιες εργασίες-tasks δεν επιλύθηκαν:



5.2.10. Οθόνη «Completed»



Εικόνα 32: Οθόνη «Completed»

Στην οθόνη «Completed» εμφανίζονται σε μία λίστα με checkbox, όλες οι εργασίες ή τμήματα των εργασιών που έχουν ενημερωθεί στο Google Calendar με ημερομηνία ενημέρωσης στο Calendar πριν από την τρέχουσα ημερομηνία. Εμφανίζονται μόνο οι εργασίες ή τα τμήματα των εργασιών που έχουν ενημερωθεί πριν την σημερινή ημερομηνία γιατί μόνο γι' αυτές μπορεί ο χρήστης να αποφασίσει αν έχουν ολοκληρωθεί ή όχι. Για τις υπόλοιπες που έχουν ενημερωθεί από τη σημερινή ημερομηνία και μετά δεν έχει έρθει ακόμη ο χρόνος εκτέλεσής τους για να μπορεί ο χρήστης να αποφασίσει αν μια εργασία την έχει πραγματοποιήσει ή όχι.

Από αυτές τις εργασίες, κάποιες έχουν πραγματοποιηθεί και ο χρήστης τις έχει επιλέξει σαν ολοκληρωμένες ενώ άλλες που δεν έγιναν στην πραγματικότητα δεν τις έχει επιλέξει. Ο χρήστης έχει τη δυνατότητα να επιλέξει τις εργασίες ή τμήματα εργασιών που θέλει να ολοκληρωθούν ή το αντίθετο. Δηλαδή, ενώ μια εργασία είναι ολοκληρωμένη (επιλεγμένη) να την επιλέξει ως μη ολοκληρωμένη. Στη συνέχεια

πατάει το κουμπί «Update» και ενημερώνονται όλες οι εγγραφές στον πίνακα task_googlecalendar και στον πίνακα tasks.

Στον πίνακα task_googlecalendar, όσες εργασίες είναι επιλεγμένες ενημερώνεται το πεδίο complete με τιμή 1, δηλαδή ολοκληρωμένο, ενώ όσες εργασίες δεν είναι επιλεγμένες ενημερώνεται το πεδίο complete με τιμή 0. Αν όλες οι εγγραφές της ίδιας εργασίας στον πίνακα task_googlecalendar είναι ολοκληρωμένες, τότε και στον πίνακα tasks ενημερώνεται το πεδίο complete με τιμή 1, δηλαδή ολοκληρωμένο και το πεδίο Remaining_duration παίρνει τη τιμή 0. Δηλαδή η εργασία χαρακτηρίζεται ολοκληρωμένη και δεν έχει υπολειπόμενο χρόνο. Αν στον πίνακα task_googlecalendar δεν είναι όλες οι εγγραφές της ίδιας εργασίας ολοκληρωμένες, τότε ο χρόνος από όλες τις ολοκληρωμένες εγγραφές μιας εργασίας αφαιρείται από τον υπολειπόμενο χρόνο Remaining_duration του πίνακα tasks.

Τα στοιχεία της εργασίας που εμφανίζονται στη λίστα είναι τα εξής:

- Το όνομα της εργασίας.
- Αν η εργασία είναι περιοδική τότε εμφανίζεται η λέξη Period και ακολουθεί ο αριθμός περιόδου.
- Αν η εργασία είναι διακοπτόμενη τότε εμφανίζεται η λέξη Part και ακολουθεί ο αριθμός του διακοπτόμενου μέρους.
- Η λέξη Start και ακολουθεί η ημερομηνία και ώρα έναρξης της εργασίας.
- Η λέξη Dur(min) και ακολουθεί ο χρόνος εκτέλεσης της εργασίας σε λεπτά.

5.3. Παραδείγματα χρήσης

Στην ενότητα αυτή, παρουσιάζονται παραδείγματα εργασιών που μπορεί να εισάγει ο χρήστης και η εφαρμογή να επιλύσει αυτές τις εργασίες. Η πιο απλή περίπτωση δραστηριότητας είναι εκείνη που έχει οριστεί συγκεκριμένο χρονοδιάγραμμα, όπως η συμμετοχή σε μία διάλεξη ή η παρακολούθηση μίας συναυλίας. Αντίθετα, δραστηριότητα όπως η αγορά βιβλίων είναι πιο ευέλικτη και ο χρήστης μπορεί να αποφασίσει μεταξύ εναλλακτικών χρονοδιαγραμμάτων. Κατά συνέπεια, ο αυτόματος προγραμματισμός αυτών των δραστηριοτήτων, στηρίζεται σε περισσότερες πληροφορίες όπως είναι η εκτίμηση της διάρκειας και το

χρονοδιάγραμμα των καταστημάτων που είναι ανοικτά. Στη συνέχεια παρουσιάζονται κάποιες αντιπροσωπευτικές περιπτώσεις αλλά υπάρχουν πολύ περισσότερες περιπτώσεις, όπως είναι φυσικό να συμβαίνει με τις πολλές δραστηριότητες που έχει ο άνθρωπος στη σημερινή εποχή.

1. «Παρακολούθηση Συναυλίας». Είναι ένα συγκεκριμένο γεγονός που θα εκτελεστεί σε συγκεκριμένη ημέρα και ώρα. Τα πεδία που ο χρήστης θα εισάγει είναι η ημερομηνία και η ώρα έναρξης, η διάρκεια και η τοποθεσία που θα εκτελεστεί.

2. «Συγγραφή σημειώσεις μαθήματος». Είναι μία εργασία που διαρκεί αρκετό χρόνο (π.χ. 25 ώρες) και θα πρέπει να είναι διακοπτόμενη. Ο χρήστης πρέπει να εισάγει την ημερομηνία έναρξης και την ημερομηνία λήξης, τη τοποθεσία, τη διάρκεια, τη μέγιστη διάρκεια του διακοπτόμενου τμήματος (π.χ. 4 ώρες), την ελάχιστη διάρκεια του διακοπτόμενου τμήματος (π.χ. 2 ώρες), την ελάχιστη χρονική απόσταση μεταξύ των τμημάτων (π.χ. 2 ώρες) και ένα ευρύ χρονικό πεδίο που μπορεί να γίνει χειροκίνητα (π.χ. κάθε μέρα από 8πμ έως 2μμ) ή με την εισαγωγή ενός προτύπου-template.

3. «Αγορά σχολικών βοηθημάτων». Είναι μία εργασία που θα διαρκέσει λίγο και μπορεί να εκτελεστεί σε οποιαδήποτε ημέρα μέχρι την ημερομηνία λήξης (π.χ. μία ημερομηνία πριν την έναρξη των σχολείων). Ο χρήστης πρέπει να εισάγει την ημερομηνία έναρξης και την ημερομηνία λήξης, τη διάρκεια (π.χ. 2 ώρες), την τοποθεσία που μπορεί να είναι οπουδήποτε-Anywhere ή σε διάφορες τοποθεσίες (π.χ. βιβλιοπωλεία) και ένα ευρύ χρονικό πεδίο που πρόκειται να εκτελεστεί η εργασία και μπορεί να γίνει χειροκίνητα (π.χ. κάθε μέρα από 9πμ έως 2μμ, 5μμ-8μμ) ή με την εισαγωγή ενός προτύπου-template (π.χ. ένα πρότυπο που έχει ώρες που λειτουργούν τα καταστήματα).

4. «Διδασκαλία μαθήματος». Είναι μία εργασία που επαναλαμβάνεται (π.χ. κάθε εβδομάδα), έχει μικρή διάρκεια, που είναι η διάρκεια του μαθήματος (π.χ. 3 ώρες) και ευρύ χρονικό πεδίο (π.χ. στο τέλος του εξαμήνου). Ο χρήστης πρέπει να εισάγει την ημερομηνία έναρξης και την ημερομηνία λήξης, τη διάρκεια, την τοποθεσία, ένα χρονικό πεδίο με συγκεκριμένες

ώρες και συγκεκριμένη ημέρα που εκτελείται η εργασία και οπωσδήποτε την περιοδικότητα της εργασίας (π.χ. εβδομαδιαία) έτσι ώστε να επαναλαμβάνεται το χρονικό της πεδίο.

5. «Εξετάσεις φροντιστηρίου». Είναι μία εργασία που επαναλαμβάνεται (π.χ. κάθε μήνα), έχει μικρή διάρκεια, που είναι η διάρκεια της εξέτασης (π.χ. 2 ώρες) και ευρύ χρονικό πεδίο (π.χ. στο τέλος του χρόνου). Η διαχείριση της εργασίας είναι παρόμοια με την εργασία «Διδασκαλία μαθήματος» με τη διαφορά ότι η περιοδικότητα είναι μηνιαία.

Τέλος, παρουσιάζονται παραδείγματα που αφορούν τους περιορισμούς διάταξης που υπάρχουν μεταξύ των εργασιών:

- Η αγορά τροφίμων, στη συνέχεια η προετοιμασία των φαγητών και τέλος το δείπνο για φίλους.
- Το διάβασμα των μαθημάτων και στη συνέχεια η επανάληψή τους.
- Η προετοιμασία μιας διάλεξης και στη συνέχεια η παρουσίασή της.

Κεφάλαιο 6ο

Συμπεράσματα & Μελλοντικές επεκτάσεις

Με τις τεχνολογίες AJAX, οι δικτυακές εφαρμογές μπορούν να ανακτούν δεδομένα από τον server ασύγχρονα, χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά της ιστοσελίδας. Η Ajax παίρνει την διαδραστικότητα από το internet και την κάνει να φαίνεται τοπική, εξαλείφοντας τις ανανεώσεις των απότομα εμφανιζόμενων σελίδων. Για το λόγο αυτό, οι τεχνολογίες AJAX έχουν αρχίσει να χρησιμοποιούνται από όλο και περισσότερες δικτυακές εφαρμογές, προσφέροντας πλούσια εμπειρία χρήστη και χαρακτηριστικά ανάλογα με αυτά των desktop εφαρμογών.

6.1. Συμπεράσματα

Μία από τις εφαρμογές που υλοποιήθηκαν με τεχνολογίες AJAX είναι η εφαρμογή SelfPlanner. Με τη χρήση των τεχνολογιών AJAX στην τρέχουσα εφαρμογή SelfPlanner, δεν υπήρχαν περιορισμοί στην υλοποίησή της, αυξήθηκε η διαδραστικότητα, η ταχύτητα της και υπήρχε συνεργασία χωρίς προβλήματα της εφαρμογής SelfPlanner με το περιβάλλον ημερολογίου της Google (Google Calendar) καθώς και με την εφαρμογή διαχείρισης τοποθεσιών (Google Maps).

Η εφαρμογή SelfPlanner χρησιμοποιώντας τις τεχνολογίες AJAX και τον αλγόριθμο SWO, κατάφερε να διαχειριστεί τον προγραμματισμό των προσωπικών εργασιών του χρήστη ευέλικτα, γρήγορα και αξιόπιστα στο διαδίκτυο και συγκεκριμένα στο Google Calendar.

6.2. Μελλοντικές επεκτάσεις

Για την εξέλιξη της συγκεκριμένης εφαρμογής SelfPlanner υπάρχουν διάφορες προτάσεις, οι οποίες θα μπορούσαν να δώσουν ακόμη περισσότερες δυνατότητες στο χρήστη. Οι μελλοντικές επεκτάσεις της εφαρμογής, με τη χρήση των τεχνολογιών AJAX και με τη MySql βάση δεδομένων, μπορούν να υλοποιηθούν με εύκολο, γρήγορο και αξιόπιστο τρόπο.

Καταρχήν, θα μπορούσαν να γίνουν επεκτάσεις στην εφαρμογή στα σημεία που δεν αποτελούν αντικείμενο της παρούσας εργασίας. Μεταξύ αυτών είναι η μαζική

διαγραφή εργασιών με διάφορα κριτήρια, η εκτύπωση των εργασιών για να έχει ο χρήστης μια διαφορετική εικόνα για τις εργασίες του, η αναζήτηση εργασιών, η εμφάνιση των εργασιών με κριτήρια για να επιλέγει ο χρήστης ποιες έχουν ολοκληρωθεί κ.α. Με τη χρήση των τεχνολογιών AJAX, όλες αυτές οι εμφανίσεις, μπορεί να γίνουν με εντυπωσιακό τρόπο και γρήγορα. Αυτό συμβαίνει γιατί κάθε φορά που ο χρήστης επιθυμεί να εμφανίζει διαφορετικά δεδομένα, η ανάκτηση των δεδομένων αυτών δεν παρεμβαίνει στην εμφάνιση και τη συμπεριφορά της εφαρμογής.

Θα μπορούσαν επίσης να οριστούν καινούριοι τύποι εργασιών, προτιμήσεων και περιορισμών, έτσι ώστε να δώσουν τη δυνατότητα στο χρήστη να έχει μεγαλύτερο έλεγχο στον τρόπο που θέλει να διαχειριστεί τις εργασίες του και στον τρόπο που θα εκτελεστούν στο Google Calendar. Κατά την επίλυση των εργασιών και την ενημέρωσή τους στο Google Calendar, μπορεί να καταχωρούνται και άλλες πληροφορίες εργασιών, όπως είναι η ζώνη ώρας, το χρώμα του συμβάντος ανάλογα με το είδος της εργασίας ή οι υπενθυμίσεις.

Στη διαχείριση των τοποθεσιών στο Google Map μπορεί να είναι χρήσιμη η αναζήτηση των τοποθεσιών που υπάρχουν στη βάση δεδομένων. Εκτός από τη δημιουργία τοποθεσιών, μπορεί να γίνεται επιλογή τοποθεσιών και καταχώρηση στη βάση δεδομένων, από μία λίστα με σημεία ενδιαφέροντος (μνημεία, καταστήματα, ξενοδοχεία κ.λ.π.) που υπάρχουν στο Google Map. Χρήσιμο μπορεί να είναι ακόμη, ο χρήστης να επιλέγει τον τρόπο που θα υπολογίζεται η χρονική απόσταση μεταξύ των τοποθεσιών, όπως είναι η χρονική απόσταση με τα πόδια ή με το αυτοκίνητο.

Μια άλλη επέκταση θα μπορούσαν να αποτελέσουν οι διαφορετικοί τύποι χρηστών. Μπορεί να οριστεί μέσα από τη διαχείριση των χρηστών μία καινούργια πληροφορία που θα είναι ο τύπος του χρήστη, όπως αν είναι απλός χρήστης ή αν είναι ένας χρήστης που θα είναι διαχειριστής-administrator και θα μπορεί να βλέπει τα αρχεία και τα ημερολόγια των άλλων χρηστών. Αυτό είναι χρήσιμο σε εταιρίες, όπου οι χρήστες διαχειρίζονται εργασίες που έχουν να κάνουν με την εταιρία τους και ο διαχειριστής μπορεί να επιβλέπει και να συντονίζει τις εργασίες των χρηστών καθώς και να τις διαχειρίζεται, όπως για παράδειγμα να αλλάζει μία εργασία και να την μεταφέρει σε άλλον χρήστη.

Εκτός από αυτές τις μελλοντικές επεκτάσεις υπάρχουν και άλλες που αναφέρονται στις επεκτάσεις [3] [4] της εφαρμογής SelfPlanner που αναπτύσσεται στο Πανεπιστήμιο Μακεδονίας.

Ανάμεσά τους είναι, η δυνατότητα η νέα εφαρμογή να μπορεί να κρατάει πληροφορίες σχετικά με πολιτιστικές δραστηριότητες που λαμβάνουν χώρα στην περιοχή της Βόρειας Ελλάδας, όπως η τοποθεσία, οι ώρες λειτουργίας, η διάρκεια επίσκεψης κ.λ.π. Με βάση αυτές τις πληροφορίες, μαζί με το ημερολόγιο του χρήστη και το προφίλ του, η εφαρμογή θα είναι σε θέση να προτείνει δραστηριότητες για το χρήστη και αν τις αποδέχεται να γίνεται ο προγραμματισμός τους στο ημερολόγιο, λαμβάνοντας υπόψη τους περιορισμούς που υπάρχουν.

Εκτός από τις δραστηριότητες που αφορούν μόνο ένα χρήστη, δηλαδή τις προσωπικές εργασίες του, η εφαρμογή μπορεί να προγραμματίζει και τις συναντήσεις που μπορεί να έχει ο χρήστης με άλλους χρήστες. Στη περίπτωση αυτή, πρέπει να υπάρχει συντονισμός μεταξύ των χρηστών, όπου η οργάνωση των συναντήσεων μπορεί να περιλαμβάνει τον επαναπρογραμματισμό των ήδη προγραμματισμένων δραστηριοτήτων για όλους τους χρήστες.

Άλλα μελλοντικά σχέδια περιλαμβάνουν επικάλυψη γεγονότων, πολλά ημερολόγια χρήστη καθώς και την ανάπτυξη της εφαρμογής σε κινητά.

Ωστόσο, μία σημαντική επέκταση αφορά στη μετατροπή του συστήματος σε ένα σύστημα σχεδιασμού, το οποίο θα κατέχει γνώσεις σχετικά με τη τρέχουσα κατάσταση του χρήστη, με πολλές δραστηριότητες που έχουν προϋποθέσεις και αποτελέσματα και ένα σύνολο στόχων που πρέπει να επιτευχθούν. Το σύστημα θα βοηθάει το χρήστη να εισάγει τις εργασίες του στο ημερολόγιο προκειμένου να επιτευχθούν οι στόχοι. Για παράδειγμα, ο χρήστης μπορεί να ορίσει ένα στόχο για ένα συνέδριο, το οποίο θα μπορούσε να δημιουργήσει μια σειρά από δραστηριότητες που θα καταχωρηθούν στο ημερολόγιο, όπως η προετοιμασία των διαφανειών, οι κρατήσεις αεροπορικών εισιτηρίων ή οι καιρικές συνθήκες (οι πληροφορίες αυτές μπορεί να εξαχθούν από το διαδίκτυο) και να εξοικονομήσει αρκετό χρόνο.

Τα παραπάνω θα καταστήσουν την εφαρμογή «ευφυέστερη» ως προς την υποστήριξη των εργασιών των χρηστών.

Με όλες τις μελλοντικές επεκτάσεις και σε συνδυασμό με την ανάπτυξη της εφαρμογής με τεχνολογίες AJAX, η εφαρμογή SelfPlanner μπορεί να κερδίσει και τους πιο δύσπιστους χρήστες και να εδραιωθεί σαν μία ισχυρή ευφυής διαδικτυακή εφαρμογή που διαχειρίζεται τον αυτόματο προγραμματισμό των εργασιών των χρηστών στο Google Calendar.

Αναφορές

1. Berry, P., Conley, K., Gervasio, M., Peintner, B., Uribe T.and Yorke-Smith, N.: '*Deploying a Personalized Time Management Agent*', 5th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-06) Industrial Track, Hakodate, Japan, pp. 1564-1571, May 2006.
2. Joslin, D.E. and Clements, D.P.: '*Squeaky Wheel Optimization*', Journal of Artificial Intelligence Research, vol. 10 (1999), 375-397.
3. Refanidis, I., Alexiadis, A. and Yorke-Smith, N: '*Beyond Calendar Mashups: SELFPLANNER 2.0*', University of Macedonia, Thessaloniki, Greece, American University of Beirut, Lebanon and SRI International, US, 2011.
4. Refanidis, I. and Alexiadis, A.: '*Deployment and evaluation of SelfPlanner, an automated individual task management system*', Department of Applied Informatics University of Macedonia, Thessaloniki, Greece, 2009
5. Refanidis, I.: '*Managing Personal Tasks with Time Constraints and Preferences*', 17th International Conference on Automated Planning and Scheduling Systems (ICAPS- 2007), Providence, Rhode Island, US, September 2007.
6. Singh, 'RCal: '*An Autonomous Agent for Intelligent Distributed Meeting Scheduling*', Robotics Institute Carnegie Mellon University, December 2003
7. Elmasri, R., Navathe S.B.: '*Θεμελιώδεις αρχές συστημάτων βάσεων δεδομένων*', Εκδόσεις Δίαυλος, 2007
8. Steven Holzner: '*Οδηγός της Ajax*', Εκδόσεις Μ. Γκιούρδας, 2009
9. Ajax Advantages and Disadvantages (http://www.ajaxwith.com/AJAX-Advantages-and-Disadvantages.html) (15/2/2011).
10. Ajax Applications (http://www.ajaxgoals.com/ajax-applications.html) (15/2/2011).
11. Apache Server (http://en.wikipedia.org/wiki/Apache_HTTP_Server) (25/2/2011)

12. Data API Developer's Guide: PHP
(http://code.google.com/apis/calendar/data/1.0/developers_guide_php.html)
(15/3/2011)
13. Google Calendar APIs and Tools
(http://code.google.com/apis/calendar/data/1.0/developers_guide_js.html)
(10/3/2011)
14. Google Maps API
(http://code.google.com/apis/maps/documentation/javascript/) (5/3/2011)
15. JQuery
(http://docs.jquery.com/How_jQuery_Works) (16/3/2011)
16. List of Ajax Frameworks
(http://en.wikipedia.org/wiki/List_of_Ajax_frameworks (15/2/2011).
17. MySQL
(http://en.wikipedia.org/wiki/MySQL) (1/3/2011)
18. PHP Manual
(http://www.php.net/manual/en/intro-whatis.php) (7/3/2011)
19. PHP Sessions
(http://www.php.net/manual/en/intro.session.php) (7/3/2011)
20. PhpMyAdmin
(http://www.phpmyadmin.net/home_page/index.php) (1/3/2011)
21. WampServer
(http://www.wampserver.com/en/presentation.php) (25/2/2011)
22. Zend Gdata
(http://framework.zend.com/manual/en/zend.gdata.calendar.html) (16/3/2011)

Παράρτημα Α: Εγκατάσταση Εφαρμογής

ΟΔΗΓΙΕΣ ΓΙΑ ΕΓΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ SELFPLANNER

1. Εγκατάσταση του WampServer
   www.wampserver.com
2. Αντιγράφουμε το φάκελο SelfPlanner που έχει τα προγράμματα κάτω από τον φάκελο wamp/www.
3. Από το WampServer επιλέγουμε PhpMyAdmin και εκτελούμε τα εξής:
   - Στη δημιουργία βάσης δεδομένων: δίνουμε όνομα = 'webcalendar' και collation = utf8_general_ci
   - Μετά τη δημιουργία της βάσης επιλέγουμε το κουμπί 'Import' και στο πεδίο 'Τοποθεσία του αρχείου κειμένου', αφού κάνουμε αναζήτηση, επιλέγουμε το αρχείο webcalendar.sql και στη συνέχεια επιλέγουμε το κουμπί 'Εκτέλεση'. Με αυτόν τον τρόπο εισάγονται οι πίνακες στη βάση.
4. Μέσα στο φάκελο SelfPlanner υπάρχει η βιβλιοθήκη Zend με την οποία διαχειριζόμαστε το Google Calendar. Το αρχείο .htaccess αλλάζει το include_path για να 'βλέπει' τη βιβλιοθήκη Zend. Αυτό που πρέπει να κάνει ο υπεύθυνος που αναλαμβάνει την εγκατάσταση της εφαρμογής SelfPlanner είναι το εξής: στο φάκελο www/bin/apache/bin να αλλάξει στο αρχείο php.ini και να βγάλει το ';' πριν από την εντολή ;extention=php.openssl.dll έτσι ώστε κάποια προγράμματα από τη βιβλιοθήκη Zend να δουλεύουν σωστά.