



2Designing Online Personal Environment

Use D.O.P.E. to become a finer designer

Σχεδίαση Συστήματος

Del.3.1

Version 1.0

Χρήστος Παπακωνσταντίνου papachri@ece.auth.gr

Αντώνης Μαυρομανώλης antomavr@ece.auth.gr

Γιώργος Μιχαηλίδης geormich@auth.gr

Θωμάς Πλιάκης tpliakis@ece.auth.gr

28/05/2019

Ιστορικό Αλλαγών

Όνομα	Ημ/νία	Περιγραφή Αλλαγής	Εκδ.
Α. Συμεωνίδης	29/05/2009	Δημιουργία Εγγράφου Προσαρμογή του ESA software engineering standards guidelines (1991) και του εγγράφου SDD document, από τους Bruegge και Dutoit (2004).	0.1

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Αντώνης Μαυρομανώλης	40	antomavr@ece.auth.gr
Θωμάς Πλιάκης	40	tpliakis@ece.auth.gr
Χρήστος Παπακωνσταντίνου	40	papachri@ece.auth.gr
Γιώργος Μιχαηλίδης	40	geormich@auth.gr

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
Λίστα Εικόνων	3
1 Προτεινόμενη Αρχιτεκτονική Λογισμικού	4
1.1 Αποδόμηση Συστήματος.....	5
1.1.1 Υποσύστημα ProjectFileCreateHandler.....	5
1.1.2 Υποσύστημα ProjectFileCreateGUI	5
1.1.3 Υποσύστημα ProjectFileBrowseHandler	6
1.1.4 Υποσύστημα ProjectFileBrowseGUI	7
1.1.5 Υποσύστημα SalesHundler	7
1.1.6 Υποσύστημα SalesGUI.....	8
1.1.7 Υποσύστημα DeviceHundler	8
1.1.8 Υποσύστημα DeviceGUI	9
1.1.9 Υποσύστημα ChatHundler.....	9
1.1.10 Υποσύστημα ChatGUI.....	10
1.1.11 Υποσύστημα Databases	10
1.2 Απεικόνιση Υλικού/Λογισμικού	11
1.2.1 Client Device.....	12
1.2.2 System Server.....	13
1.2.3 Συνολικό διάγραμμα ανάπτυξης	14
Έλεγχος Πρόσβασης και Ασφάλεια	15
2 Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού	16
3 RESTful API.....	17
3.1 Endpoints	17
3.1.1 Project microservice.....	17
3.1.2 Files microservice	20
3.1.3 Devices microservice.....	22
3.1.4 Purchasing Program microservice.....	25
3.1.5 Purchasing Space microservice	26
4 Ανάλυση SWOT.....	27
5 Χρονοδιάγραμμα και Κοστολόγηση Έργου	28

Λίστα Εικόνων

Εικόνα 1 : ProjectFileCreateHandler.....	5
Εικόνα 2 : ProjectFileCreateGUI	6
Εικόνα 3: ProjectFileBrowserHandler.....	6
Εικόνα 4: ProjectFileBrowseGUI.....	7
Εικόνα 5: SalesHandler	8
Εικόνα 6: SalesGUI.....	8

Εικόνα 7: DeviceHandler	9
Εικόνα 8: DeviceGUI	9
Εικόνα 9: ChatHandler	10
Εικόνα 10: ChatGUI	10
Εικόνα 11:Databases	11
Εικόνα 12 : Υλικο - Λογισμικό	12
Εικόνα 13:Client Device	13
Εικόνα 14: System Server	13
Εικόνα 15: Διάγραμμα Σχεδίασης.....	15
Εικόνα 16: Ανάλυση SWOT	27
Εικόνα 17: Χρονοδιάγραμμα.....	28
Εικόνα 18: Διάγραμμα Gantt.....	29

1 Προτεινόμενη Αρχιτεκτονική Λογισμικού

Το παρόν κεφάλαιο περιγράφει το μοντέλο σχεδίασης του συστήματος της εφαρμογής. Τα μοντέλα

σχεδίασης είναι αφαιρετικά και έχουν οργανωτική δομή, με στόχο τη μείωση της πολυπλοκότητας

του συστήματος μετατρέποντας σύνθετα προβλήματα σε απλά.

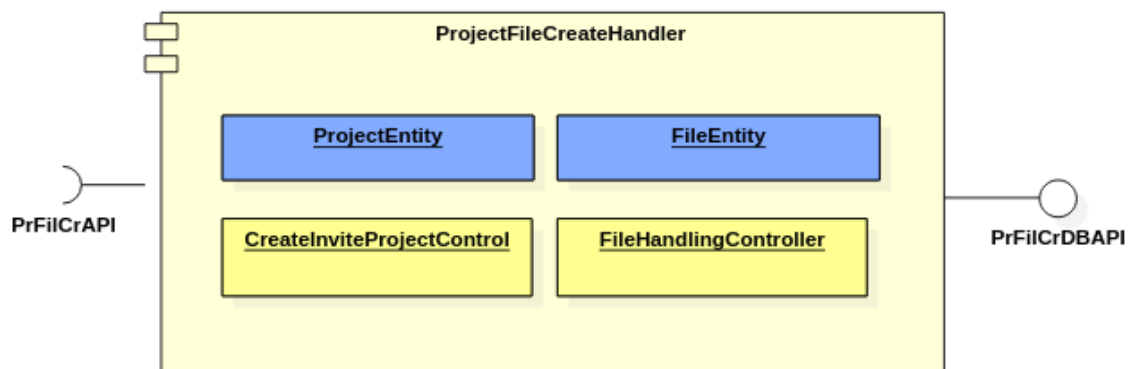
1.1 Αποδόμηση Συστήματος

Πρόκειται για το διαχωρισμό του συστήματος σε επιμέρους υποσυστήματα και τμήματα, τα οποία

φέρουν ξεχωριστές ευθύνες και λειτουργίες. Στην παράγραφο που ακολουθεί ορίζονται τα υποσυστήματα με τα τμήματα που εμπεριέχουν και αναλύεται η λειτουργικότητα καθενός από αυτά.

1.1.1 Υποσύστημα ProjectFileCreateHandler

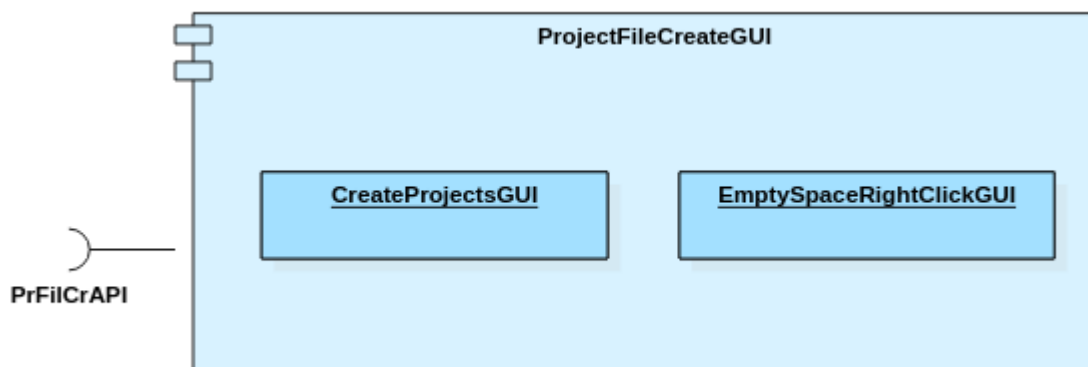
Το συγκεκριμένο υποσύστημα εμπεριέχει ως τμήματά του τους ελεγκτές που ελέγχουν την δημιουργία και τις υπόλοιπες επιλογές ενός project και ενός αρχείου καθώς και τις οντότητες που τα αφορούν. Συγκεκριμένα, οι κλάσεις που περιλαμβάνει το υποσύστημα είναι οι ProjectEntity, FileEntity, CreateInviteProjectControl και η FileHandlingController.



Εικόνα 1 : ProjectFileCreateHandler

1.1.2 Υποσύστημα ProjectFileCreateGUI

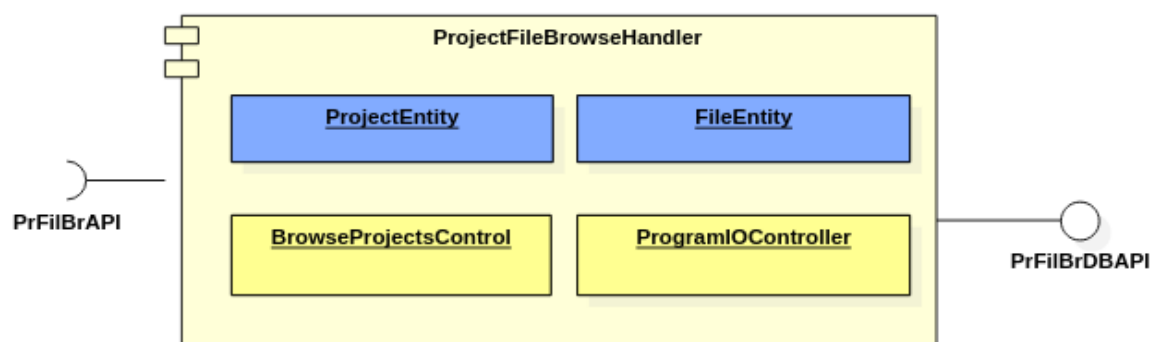
Το υποσύστημα αυτό αποτελείται από τις διεπαφές που συναντά ένας χρήστης όταν θέλει να δημιουργήσει ένα νέο Project ή ένα νέο αρχείο. Αναλυτικότερα, οι κλάσεις που το απαρτίζουν είναι η CreateProjectsGUI, δηλαδή η σελίδα που εξυπηρετεί την δημιουργία νέου project και η EmptySpaceRightClickGUI, που αφορά το μενού που εμφανίζεται όταν ο χρήστης κάνει δεξί κλικ στον κενό χώρο μεταξύ των αρχείων.



Εικόνα 2 : ProjectFileCreateGUI

1.1.3 Υποσύστημα ProjectFileBrowseHandler

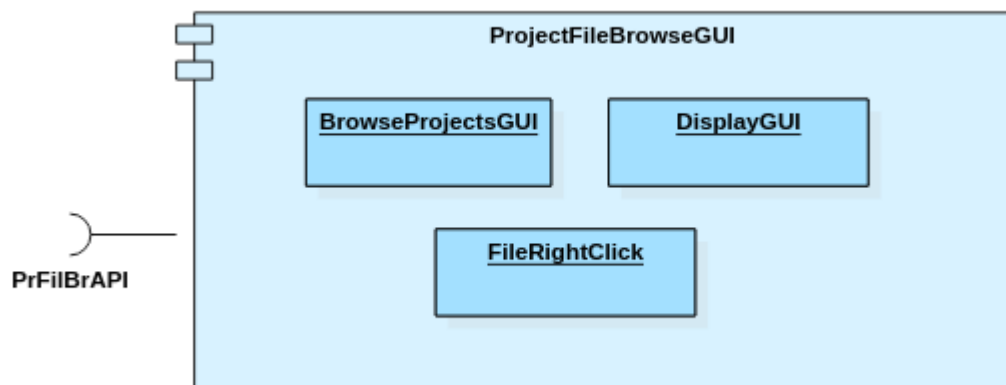
Το συγκεκριμένο υποσύστημα περιέχει ως τμήματά του τον ελεγκτή που ελέγχει την περιήγηση στα αρχεία ενός project και τον ελεγκτή που ελέγχει τα αρχεία που εισέρχονται ή εξέρχονται από τα προγράμματα, καθώς και τις οντότητες που τα αφορούν. Συγκεκριμένα, οι κλάσεις που περιλαμβάνει το υποσύστημα είναι οι **ProjectEntity**, **FileEntity**, **BrowseProjectsControl** και η **ProgramIOController**.



Εικόνα 3: ProjectFileBrowserHandler

1.1.4 Υποσύστημα ProjectFileBrowseGUI

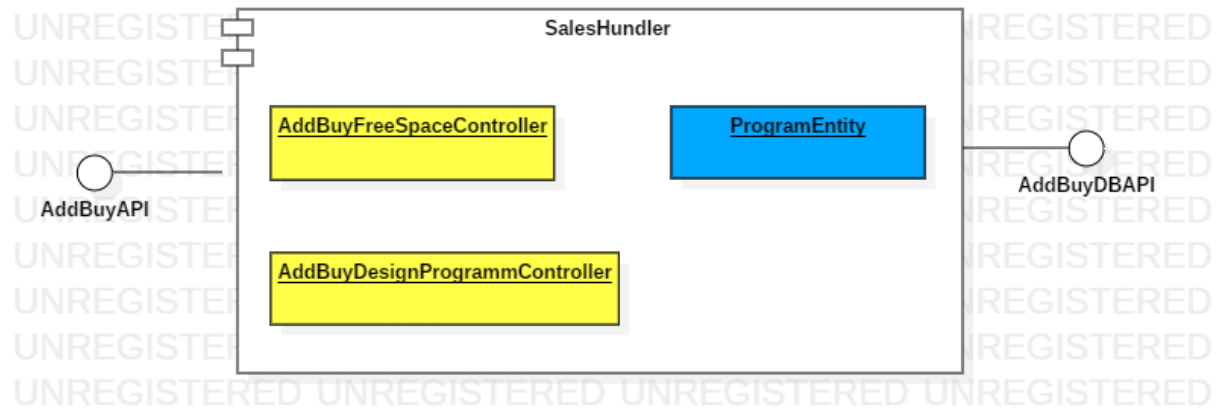
Το υποσύστημα αυτό αποτελείται από τις διεπαφές που συναντά ένας χρήστης όταν θέλει να περιηγηθεί στα υπάρχοντα Project ή αρχείο. Αναλυτικότερα, οι κλάσεις που το απαρτίζουν είναι η `BrowseProjectsGUI`, δηλαδή η σελίδα που εμφανίζει όλα τα διαθέσιμα project, η `DisplayGUI`, που αφορά τη γραφική διεπαφή του χρήστη με τα αρχεία και το περιβάλλον του συστήματος και η `FileRightClick` που αφορά το μενού που εμφανίζεται όταν ο χρήστης κάνει δεξί κλικ πάνω σε ένα αρχείο.



Εικόνα 4: `ProjectFileBrowseGUI`

1.1.5 Υποσύστημα SalesHundler

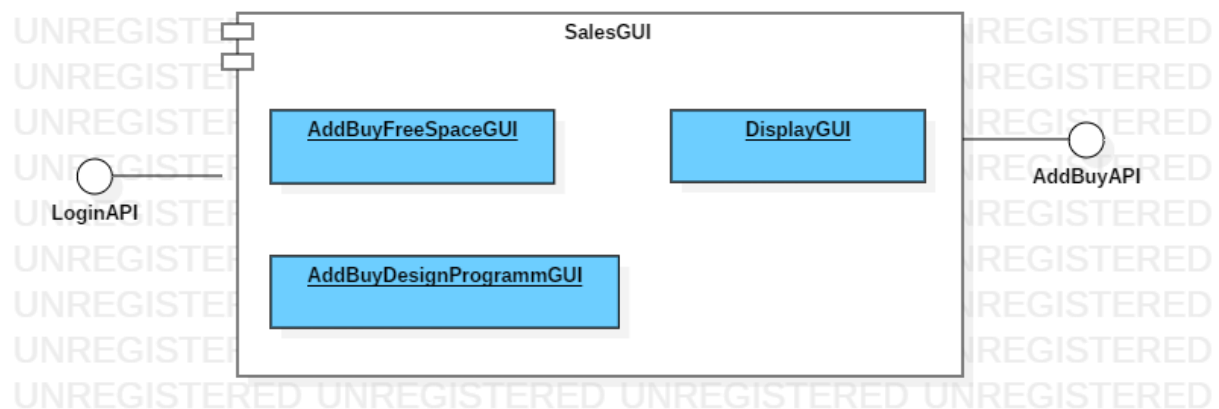
Πρόκειται για το υποσύστημα που αναλαμβάνει να επιτελέσει κάθε λειτουργία που αφορά την προσθήκη-αγορά νέων υποστηριζόμενων σχεδιαστικών προγραμμάτων από τον χρήστη και την αναβάθμιση του αποθηκευτικού του χώρου που έχει στην πλατφόρμα, δεδομένου ότι μπορεί κάποια στιγμή να τελειώσει ο ελεύθερος χώρος. Οι ελεγκτές που το απαρτίζουν είναι υπεύθυνοι για την πρόσθεση ή την αγορά ενός σχεδιαστικού προγράμματος, την επιλογή του και την φόρτωσή του (`AddBuyDesignProgrammController`) και για την αγορά επιπλέον αποθηκευτικού χώρου (`AddBuyFreeSpaceController`). Οι ελεγκτές αυτοί φέρουν εις πέρας τις αρμοδιότητές τους μέσω των στοιχείων που συλλέγουν από την κλάση οντότητας `ProgramEntity`.



Εικόνα 5: SalesHandler

1.1.6 Υποσύστημα SalesGUI

Πρόκειται για το υποσύστημα κλάσεων διεπαφών που σχετίζονται με την προσθήκη-αγορά νέων υποστηριζόμενων σχεδιαστικών προγραμμάτων από τον χρήστη και την αναβάθμιση του αποθηκευτικού χώρου που έχει στην πλατφόρμα, δεδομένου ότι μπορεί κάποια στιγμή να τελειώσει ο ελεύθερος χώρος. Η διεπαφή **AddBuyDesignProgrammGUI** αναπαριστά τη σελίδα που προβάλλεται στο χρήστη κατά τη διαδικασία αγοράς ή προσθήκης ενός σχεδιαστικού προγράμματος, η διεπαφή **AddBuyFreeSpaceGUI** αναπαριστά τη σελίδα που προβάλλεται στο χρήστη κατά τη διαδικασία προβολής ή προσθήκης αποθηκευτικού χώρου, ενώ η **DisplayGUI** εμφανίζει το chat και οτιδήποτε λειτουργία ή διεπαφή χρειάζεται το GUI της αγοράς-προσθήκης σχεδιαστικών προγραμμάτων ή αποθηκευτικού χώρου κάτι που χρησιμοποιούν και τα 2 παραπάνω GUI.

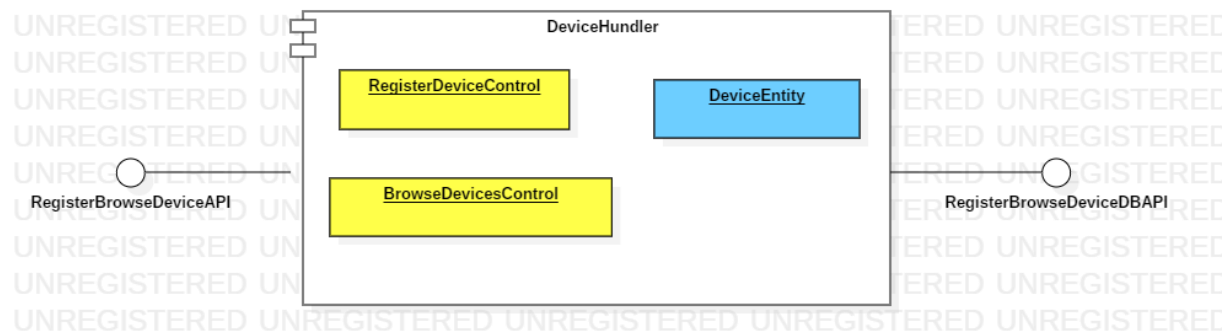


Εικόνα 6: SalesGUI

1.1.7 Υποσύστημα DeviceHundler

Πρόκειται για το υποσύστημα που αναλαμβάνει να επιτελέσει κάθε λειτουργία που αφορά την προσθήκη καινούργιας συσκευής στην πλατφόρμα και την περιήγηση του χρήστη στις συνδεδεμένες συσκευές. Οι ελεγκτές που το απαρτίζουν είναι υπεύθυνοι για την καταχώρηση μιας νέας συσκευής στην πλατφόρμα (**RegisterDeviceControl**) και για την περιήγηση στη λίστα των συνδεδεμένων συσκευών στην πλατφόρμα.

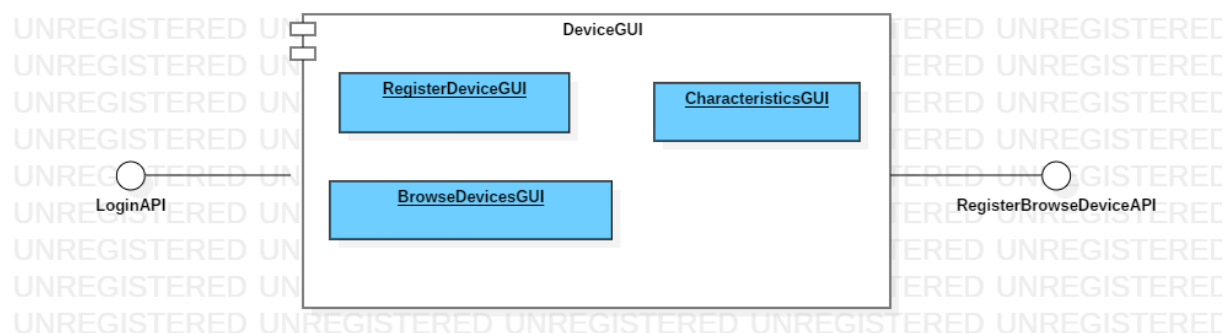
(**BrowseDevicesControl**). Οι ελεγκτές αυτοί φέρουν εις πέρας τις αρμοδιότητές τους μέσω των στοιχείων που συλλέγουν από την κλάση οντότητας **DeviceEntity**.



Εικόνα 7: DeviceHandler

1.1.8 Υποσύστημα DeviceGUI

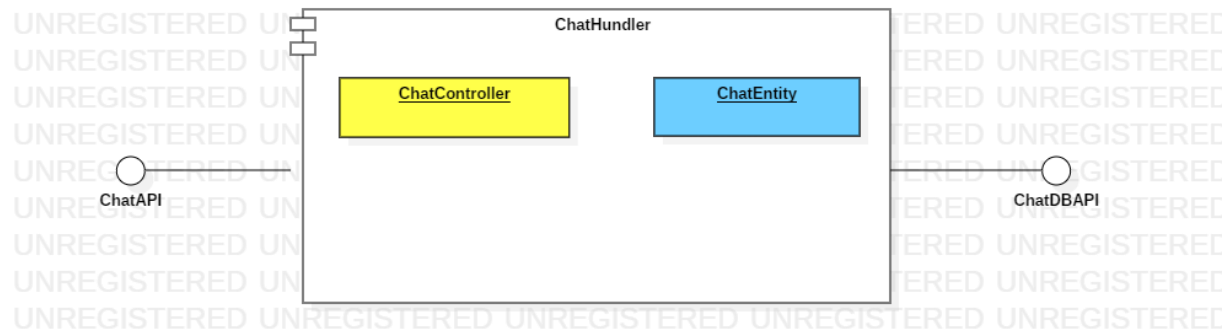
Πρόκειται για το υποσύστημα κλάσεων διεπαφών που σχετίζονται με την προσθήκη καινούργιας συσκευής στην πλατφόρμα και την περιήγηση του χρήστη στις συνδεδεμένες συσκευές. Η διεπαφή **RegisterDeviceGUI** αναπαριστά τη σελίδα που εμφανίζεται στο χρήστη με τη φόρμα καταχώρησης της εξωτερικής συσκευής, η διεπαφή **BrowseDevicesGUI** αναπαριστά τη σελίδα που προβάλλεται στο χρήστη κατά την περιήγηση μεταξύ των συνδεδεμένων συσκευών στην πλατφόρμα, ενώ η **CharacteristicsGUI** εμφανίζει την διεπαφή για τα χαρακτηριστικά κάθε συνδεδεμένης συσκευής στην πλατφόρμα.



Εικόνα 8: DeviceGUI

1.1.9 Υποσύστημα ChatHandler

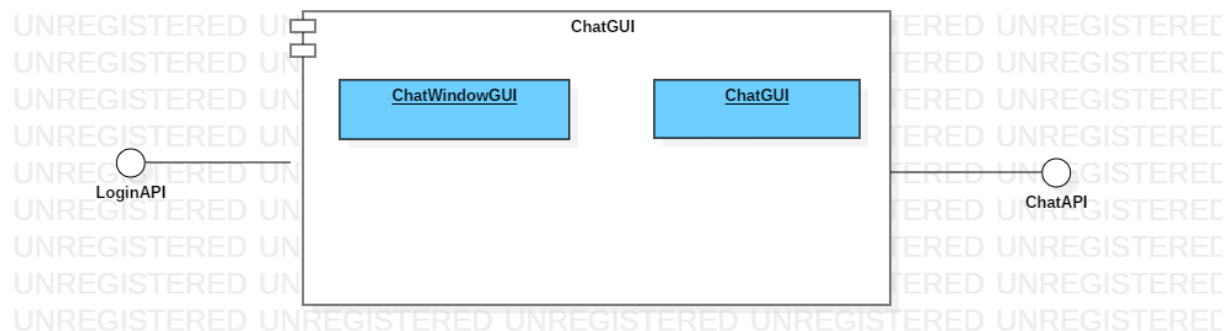
Πρόκειται για το υποσύστημα που αναλαμβάνει να επιτελέσει κάθε λειτουργία που αφορά την επικοινωνία των χρηστών μεταξύ τους. Ο ελεγκτής που το απαρτίζει είναι υπεύθυνος για την κλήση όλων των κατάλληλων συναρτήσεων για τη λειτουργία του Chat (**ChatControl**). Ο ελεγκτής αυτός φέρει εις πέρας τις αρμοδιότητές του μέσω των στοιχείων που συλλέγει από την κλάση οντότητας **ChatEntity**.



Εικόνα 9: ChatHandler

1.1.10 Υποσύστημα ChatGUI

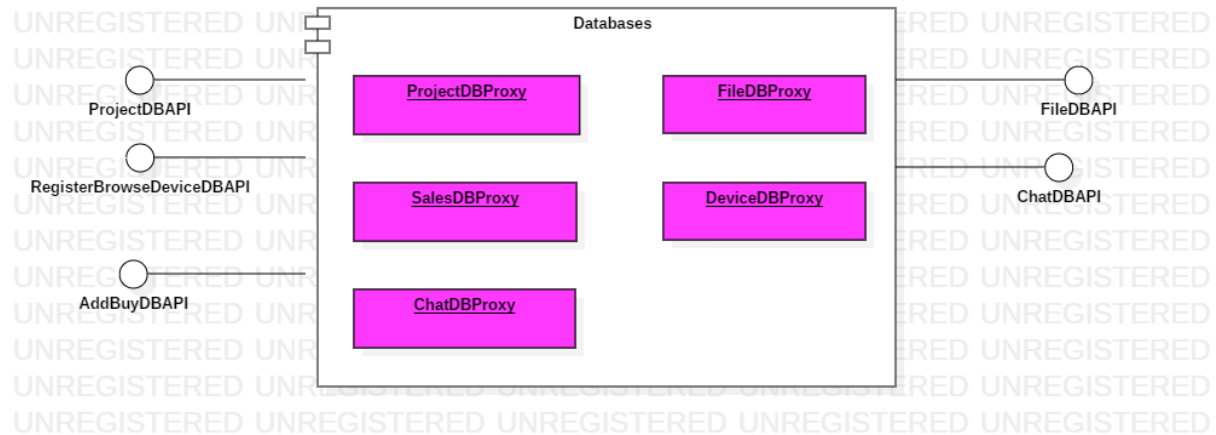
Πρόκειται για το υποσύστημα κλάσεων διεπαφών που σχετίζονται με την επικοινωνία των χρηστών μεταξύ τους. Η διεπαφή **ChatWindowGUI** αναπαριστά τη διεπαφή για την εμφάνιση του παραθύρου επικοινωνίας 2 χρηστών, η διεπαφή **ChatGUI** αναπαριστά τη διεπαφή για την εμφάνιση του chat με όλους τους συνδεδεμένους χρήστες.



Εικόνα 10: ChatGUI

1.1.11 Υποσύστημα Databases

Όλα τα παραπάνω υποσυστήματα, που διαχειρίζονται λειτουργίες της εφαρμογής, επικοινωνούν με αντίστοιχες βάσεις δεδομένων. Οι τελευταίες με τη σειρά τους διαθέτουν αποθηκευμένες όλες τις πληροφορίες, που είναι απαραίτητες για την περάτωση των προαναφερθέντων λειτουργιών και συνεπώς, την ορθή εκτέλεση της εφαρμογής. Πιο αναλυτικά, οι βάσεις αυτές εξυπηρετούν την πρόσβαση στους φακέλους των projects (**ProjectDBProxy**), την φόρτωση της λίστας με τα αρχεία του κάθε χρήστη (**FileDBProxy**), την επικοινωνία των ελεγκτών για την αγορά ή την προσθήκη σχεδιαστικού προγράμματος ή αποθηκευτικού χώρου για να κάνουν τις διάφορες λειτουργίες τους (είτε τραβάνε δεδομένα από αυτήν, είτε αποθηκεύουν) (**SalesDBProxy**), την αποθήκευση και τη φόρτωση συνδεδεμένων συσκευών στην πλατφόρμα (**DeviceDBProxy**) καθώς και την αποθήκευση των υπόλοιπων χρηστών που είναι συνδεδεμένοι σε ένα chat ώστε να μπορεί να γίνει η επικοινωνία μαζί τους (**ChatDBProxy**).

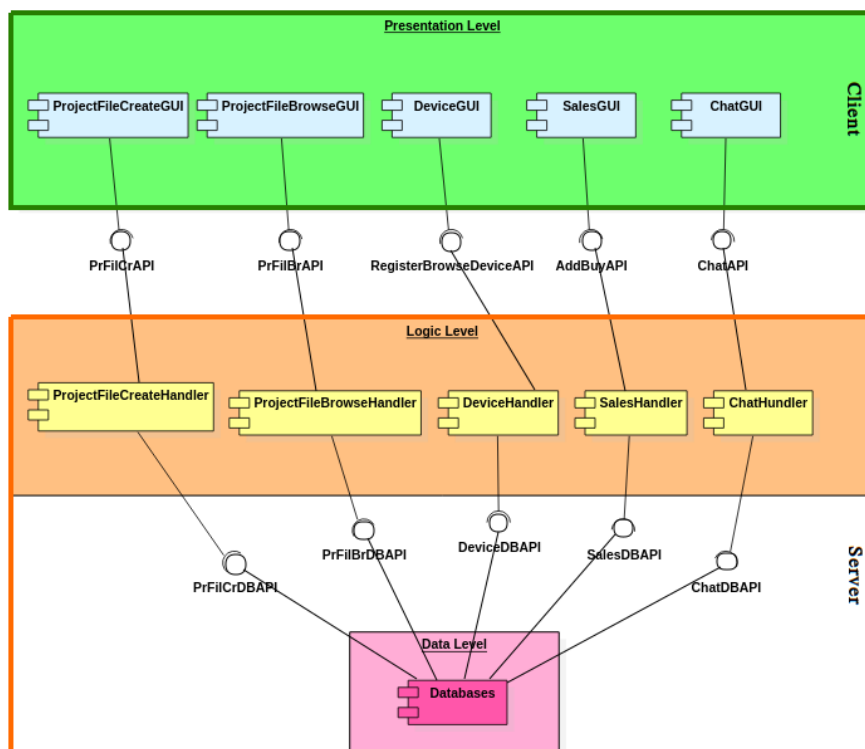


Εικόνα 11: Databases

Εν συνεχεία, παρατίθεται το διάγραμμα τμημάτων που παρουσιάζει τις εξαρτήσεις μεταξύ των επιμέρους υποσυστημάτων. Τα διαγράμματα τμημάτων αξιοποιούνται για επιμερισμό του συστήματος σε ομάδες ανάπτυξης, αλλά και προς έλεγχο για το αν παρέχεται η απαιτούμενη λειτουργικότητα από το σύστημα.

1.2 Απεικόνιση Υλικού/Λογισμικού

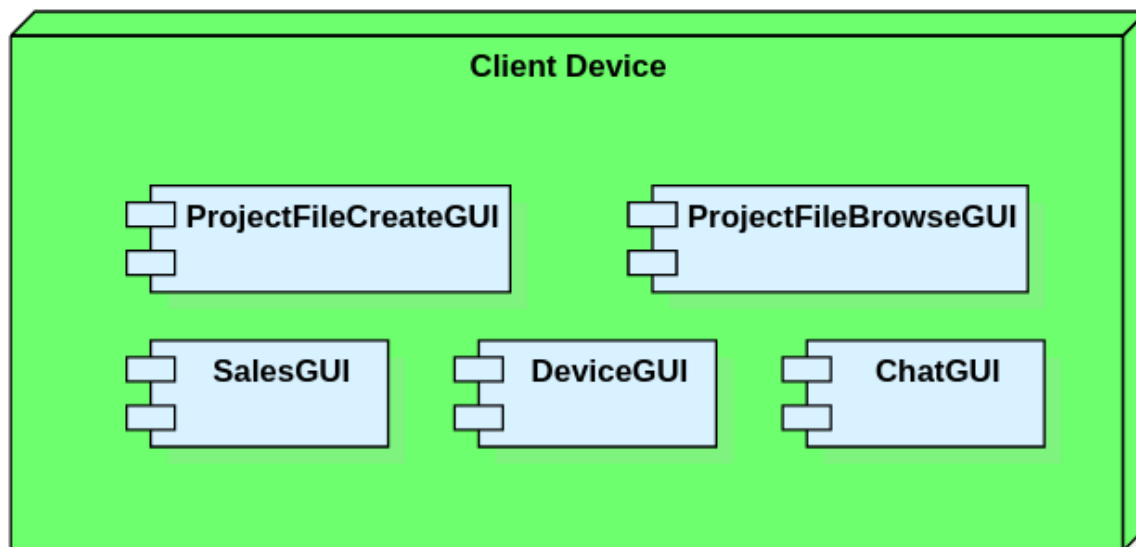
Η εφαρμογή cloud D.O.P.E. , η οποία αναπτύσσεται στο παρόν έγγραφο, έχει ως βασική προϋπόθεση για τη λειτουργία της την συνεχή επικοινωνία του εκάστοτε χρήστη με αυτή. Συνεπώς εξετάζοντας τις υπάρχουσες αρχιτεκτονικές βρέθηκε ότι η κατάλληλη αρχιτεκτονική για το σύστημα προς ανάπτυξη είναι η αρχιτεκτονική πελάτη-διακομιστή (client-server).



Εικόνα 12 : Υλκό - Λογισμικό

1.2.1 Client Device

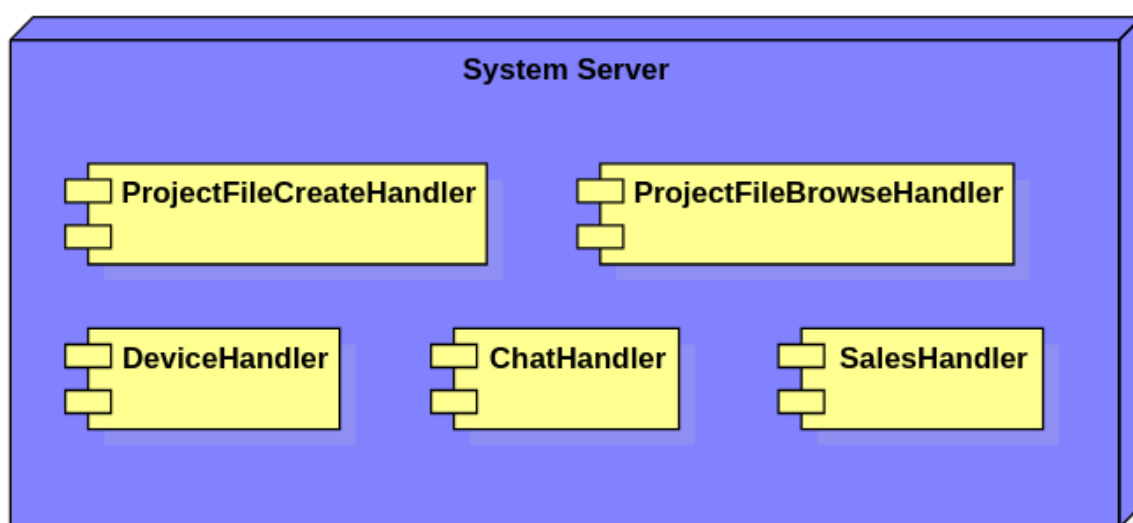
Ο πρώτος κόμβος του συστήματος περιέχει αποκλειστικά τις διεπαφές που απαιτούνται από το χρήστη προκειμένου να χρησιμοποιηθεί η εφαρμογή. Η χρήση του γίνεται μέσω ενός προγράμματος περιήγησης, το οποίο θα μπορούσε να βρίσκεται σε οποιαδήποτε ηλεκτρονικό υπολογιστή που διαθέτει σύνδεση στο διαδίκτυο, η οποία κρίνεται αναγκαία για την ομαλή επικοινωνία και αναταλλαγή δεδομένων με τον άλλο κόμβο του συστήματος.



Εικόνα 13: Client Device

1.2.2 System Server

Σε αυτόν τον κόμβο περιέχονται όλα εκείνα τα υποσυστήματα που αναλαμβάνουν να επικοινωνήσουν με τον server του συστήματος έτσι ώστε να ικανοποιηθούν τα αιτήματα των χρηστών. Γενικά ο κόμβος αυτός οργανώνει την ορθή λειτουργία όλου του συστήματος (μέσω των handler υποσυστημάτων του) και την επικοινωνία με την εξωτερική βάση δεδομένων του συστήματος (μέσω των proxies).



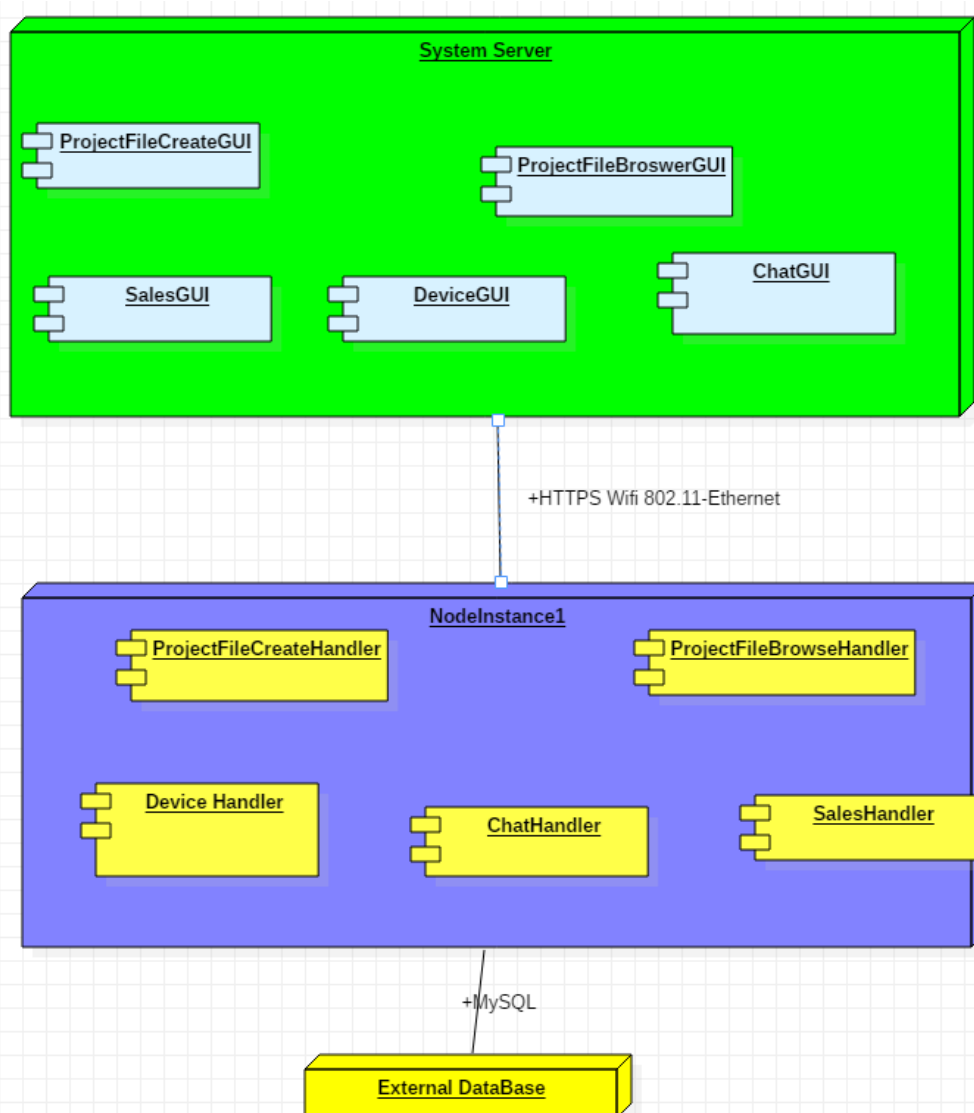
Εικόνα 14: System Server

1.2.3 Συνολικό διάγραμμα ανάπτυξης

Εν συνεχεία θα παρουσιαστεί ο τρόπος με τον οποίο επικοινωνούν οι διάφοροι κόμβοι του συστήματος. Αρχικά, ο κόμβος Client device και ο κόμβος System Server επικοινωνούν μέσω διαδικτύου. Το πρωτόκολλο με το οποίο επικοινωνούν οι δύο κόμβοι είναι το HTTPS. Το πρωτόκολλο HTTPS χρησιμοποιείται στην πληροφορική για να δηλώσει μία ασφαλή δικτυακή σύνδεση http. Ένας σύνδεσμος (URL) που αρχίζει με το πρόθεμα https υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα (443 αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα. Ακόμη πρέπει να τονιστεί ότι αναφέρεται στον συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL). Η κρυπτογράφηση που χρησιμοποιείται διασφαλίζει ότι τα κρυπτογραφημένα δεδομένα δεν θα μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες ή από επιθέσεις man-in-the-middle.

Τέλος, η σύνδεση μεταξύ client και server γίνεται είτε με WiFi είτε με Ethernet. Συνεχίζοντας την ανάλυση στο παρακάτω διάγραμμα έχει προσθεθεί η εξωτερική βάση δεδομένων.

Όπως φαίνεται σε αυτό η επικοινωνία μεταξύ των συστημάτων γίνεται μέσω της MySQL, ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ιδιαίτερα δημοφιλές για διαδικτυακά προγράμματα και ιστοσελίδες.



Εικόνα 15: Διάγραμμα Σχεδίασης

1.3 Έλεγχος Πρόσβασης και Ασφάλεια

Ο έλεγχος πρόσβασης και η ασφάλεια περιγράφουν τη μοντελοποίηση των χρηστών του συστήματος με τη μορφή ενός πίνακα πρόσβασης (access matrix). Η παράγραφος αυτή επίσης περιγράφει και διάφορα θέματα ασφαλείας, όπως επιλογή μηχανισμού αυθεντικοποίησης, χρήση κρυπτογράφησης και διαχείριση κλειδιών.

2 Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού

Δεν πραγματοποιήθηκε κάποια αλλαγή από παραδοτέο σε παραδοτέο για αυτό και δεν έγινε κάποιος πίνακας ιχνηλασιμότητας.

3 RESTful API

3.1 Endpoints

3.1.1 Project microservice

Το συγκεκριμένο API είναι υπεύθυνο για την διαχείριση των Projects που δημιουργεί ή συμμετέχει ένας χρήστης.

Method	URL		
GET	project/{project id}		
Description			
Επιστρέφει όλες τις πληροφορίες για τον προσδιορισμό του project .			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{project id}	Αναφέρεται στο project με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
Sample Request Body (if exists)			
Not needed			
Sample Responses			
Status	Response		
200	{ "project": [{ "ProjectName": "myProject", "CreatorID": 13 "ParticipantsNames": ["myName","minionName1"] }] }		
401	{ "error": " Unauthorized access."}		
404	{ "error": " Project ID not found"}		
500	{ "error": " Something went wrong. Please try again later."}		

Method	URL
PUT	project/{project id}

Description			
Κάνει update τα στοιχεία ενός project.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{project id}	Αναφέρεται στο project με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
ParticipatnsNames	Required	The list with name of the participants.	String []
Sample Request Body (if exists)			
<pre>{ "ParticipantsNames": ["myName","minionName1"] }</pre>			
Sample Responses			
Status	Response		
200	<pre>{ "project": [{ "ProjectName": "myProject", "CreatorID": 13 "ParticipantsNames": ["myName","minionName1"] }] }</pre>		
401	{ "error": " Unauthorized access."}		
404	{ "error": " Project ID not found"}		
500	{ "error": " Something went wrong. Please try again later."}		

Method	URL
DELETE	project/{project id}
Description	
Διαγράφει το project με το συγκεκριμένο id .	
Parameters	
Path Parameters	
Path Parameter	Description
{project id}	Αναφέρεται στο project με το συγκεκριμένο id.

Query string parameters	
Sample Request Body (if exists)	
Not needed	
Sample Responses	
Status	Response
200	OK
401	{"error": "Unauthorized access."}
404	{"error": "Project ID not found"}
500	{"error": "Something went wrong. Please try again later."}

Method	URL		
POST	project/{project id}		
Description			
Δημιουργεί ένα καινούργιο Project με το συγκεκριμένο id .			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{project id}	Αναφέρεται στο project με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
ProjectName	Required	The name of the project.	String
ParticipatnsNames	Optional	The list with name of the participants.	String []
CreatorID	Required	The user id of the creator.	Int
Sample Request Body (if exists)			
<pre>{ "project": [{ "ProjectName": "myProject", "CreatorID": 13 "ParticipantsNames": ["myName","minionName1"] }] }</pre>			
Sample Responses			

Status	Response
200	{ "projectId": 14 }
400	{"error": "Please specify database version."}
500	{"error": "Something went wrong. Please try again later."}

3.1.2 Files microservice

Η υπηρεσία υπεύθνη για τη διαχείριση αρχείων.

Method	URL		
POST	file/{fileId}		
Description			
Δημιουργεί ένα νέο αρχείο .			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{fileId}	Αναφέρεται στο αρχείο με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
FileName	Required	The name of the file to be created.	String
FileType	Required	The type of the file to be created	String
Sample Request Body (if exists)			
<pre>{ "file": [{ "FileName": "myFile", "FileType": ".txt", }] }</pre>			
Sample Responses			
Status	Response		
200	<pre>{ "fileId": 13 }</pre>		

500	{"error": "Something went wrong. Please try again later."}
-----	--

Method	URL		
PUT	file/{fileId}		
Description			
Επεξεργασία ενός αρχείου .			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{fileId}	Αναφέρεται στο αρχείο με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
Sample Request Body (if exists)			
Not needed			
Sample Responses			
Status	Response		
200	OK		
401	{ "error" :" Unauthorized access."}		
404	{ "error" :" File ID not found"}		
500	{ "error" :" Something went wrong. Please try again later."}		

Method	URL		
DELETE	file/{fileId}		
Description			
Διαγραφή ενός αρχείου .			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{fileId}	Αναφέρεται στο αρχείο με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
Sample Request Body (if exists)			
Not needed			

Sample Responses	
Status	Response
200	OK
401	{"error": "Unauthorized access."}
404	{"error": "File ID not found"}
500	{"error": "Something went wrong. Please try again later."}

3.1.3 Devices microservice

Το microservice υπεύθυνο για τη διαχείριση συσκευών.

Method	URL		
GET	device/{DeviceID}		
Description			
Επιστρέφει όλες τις πληροφορίες που αφορούν την συσκευή με το συγκεκριμένο ID.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{deviceId}	Αναφέρεται στοστη συσκευή με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
Sample Request Body (if exists)			
Not needed			
Sample Responses			
Status	Response		
200	{ "device" : [{ "DeviceType" : "Printer3D" "DeviceName" : "myPrinter" "DeviceOwner" : "Krits" "Status" : True "LiveStreamAvailability" : "False" }] }		

	<pre> }] } </pre>
401	{"error": "Unauthorized access."}
404	{"error": "Device ID not found"}
500	{"error": "Something went wrong. Please try again later."}

Method	URL		
POST	device/{DeviceID}		
Description			
Προσθέτει στη λίστα συσκευών την συσκευή με το συγκεκριμένο ID.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{deviceId}	Αναφέρεται στοστη συσκευή με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
DeviceOwner	Required	The owner of the device	String
DeviceName	Required	The name of the device	String
DeviceType	Required	The type of the device	String
Sample Request Body (if exists)			
<pre>{ "DeviceType": "Printer3D" "DeviceName": "myPrinter" "DeviceOwner": "Krits" }</pre>			
Sample Responses			
Status	Response		
200	<pre>{ "DeviceID": "13" }</pre>		
401	<pre>{"error": " Unauthorized access."}</pre>		
500	<pre>{"error": " Something went wrong. Please try again later."}</pre>		

Method	URL
--------	-----

PUT	device/{DeviceID}		
Description			
Κάνει update κάποια χαρακτηριστικά της συσκευής με το συγκεκριμένο ID.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{deviceId}	Αναφέρεται στοστη συσκευή με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
DeviceOwner	Required	The owner of the device	String
DeviceName	Required	The name of the device	String
Sample Request Body (if exists)			
{ "DeviceName" : "myPrinter" "DeviceOwner" : "Krits" }			
Sample Responses			
Status	Response		
200	OK		
401	{ "error": " Unauthorized access." }		
500	{ "error": " Something went wrong. Please try again later." }		

Method	URL		
DELETE	device/{DeviceID}		
Description			
Προσθέτει στη λίστα συσκευών την συσκευή με το συγκεκριμένο ID.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{deviceID}	Διαγράφει τη συσκευή με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
Sample Request Body (if exists)			
Not Needed			

Sample Responses	
Status	Response
200	OK
401	{"error": "Unauthorized access."}
500	{"error": "Something went wrong. Please try again later."}

3.1.4 Purchasing Program microservice

Το microservice υπεύθυνο για τη διαχείριση των αγορών προγραμμάτων.

Method	URL		
POST	progPurchase/{purchId}		
Description			
Δημιουργεί αίτηση αγοράς προγράμματος με το συγκεκριμένο ID.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{purchID}	Αναφέρεται στην αγορά με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
UserID	Required	The ID of the user that makes a purchase.	Int
ProgrammID	Required	The ID of the program to be purchased.	Int
ProgramName	Required	The name of the program to be purchased.	String
Sample Request Body (if exists)			
<pre>{ "UderID" : "13 "ProgrammID" : 4 "ProgrammName" : "Solidworks" }</pre>			
Sample Responses			
Status	Response		
200	OK		

401	{"error": "Unauthorized access."}
500	{"error": "Something went wrong. Please try again later."}

3.1.5 Purchasing Space microservice

Το microservice υπεύθυνο για τη διαχείριση των αγορών χώρου μνήμης.

Method	URL		
POST	spcePurchase/{purchId}		
Description			
Δημιουργεί αίτηση αγοράς χώρου αποθήκευσης.			
Parameters			
Path Parameters			
Path Parameter	Descriptoin		
{deviceID}	Αναφέρεται στο πρόγραμμα με το συγκεκριμένο id.		
Query string parameters			
Query Parameters	Required/Optional	Description	Type
UserID	Required	The ID of the user that makes a purchase.	Int
SpaceSize	Required	The size in GB of the space to be purchased.	Int
Sample Request Body (if exists)			
{ "UderID" : "13 "SpaceSize" : 7 }			
Sample Responses			
Status	Response		
200	OK		
401	{"error": " Unauthorized access."}		
500	{"error": " Something went wrong. Please try again later."}		

4 Ανάλυση SWOT

Strengths:

- Φιλικό για το χρήστη
- Απουσία ανταγωνιστικού προϊόντος
- Χρήση ποικιλίας σχεδιαστικών προγραμμάτων μέσα σε μια πλατφόρμα
- Διευκόλυνση στην επικοινωνία μεταξύ μιας ομάδας σχεδιαστών
- Χρήση απαιτητικών σχεδιαστικών προγραμμάτων μέσω browser

Weaknesses:

Μεγάλο κόστος για χρήση σχεδιαστικών προγραμμάτων

Έλλειψη στρατηγικών συνεργασιών με εταιρείες-κατόχους σχεδιαστικών προγραμμάτων

Opportunities:

Σχεδιαστές χωρίς πολλούς υπολογιστικούς πόρους μπορούν να χρησιμοποιούν απαιτητικά προγράμματα

Γρηγορότερη ολοκλήρωση ενός project

Threats:

Άρνηση από τις εταιρείες-κατόχους, χρήση των σχεδιαστικών προγραμμάτων τους στην πλατφόρμα

Ανάλυση SWOT

	Helpful	Harmful
Internal	<u>Strengths:</u> <ul style="list-style-type: none"> • Φιλικό για το χρήστη • Απουσία ανταγωνιστικού προϊόντος • Χρήση ποικιλίας σχεδιαστικών προγραμμάτων μέσα σε μια πλατφόρμα • Διευκόλυνση στην επικοινωνία μεταξύ μιας ομάδας σχεδιαστών • Χρήση απαιτητικών σχεδιαστικών προγραμμάτων μέσω browser 	<u>Weaknesses:</u> <ul style="list-style-type: none"> • Μεγάλο κόστος για χρήση σχεδιαστικών προγραμμάτων • Έλλειψη στρατηγικών συνεργασιών με εταιρείες-κατόχους σχεδιαστικών προγραμμάτων
External	<u>Opportunities:</u> <ul style="list-style-type: none"> • Σχεδιαστές χωρίς πολλούς υπολογιστικούς πόρους μπορούν να χρησιμοποιούν απαιτητικά προγράμματα • Γρηγορότερη ολοκλήρωση ενός project 	<u>Threats:</u> <ul style="list-style-type: none"> • Άρνηση από τις εταιρείες-κατόχους, χρήση των σχεδιαστικών προγραμμάτων τους στην πλατφόρμα

Εικόνα 16: Ανάλυση SWOT

5 Χρονοδιάγραμμα και Κοστολόγηση Έργου

Χρονοδιάγραμμα:

Κατά μέσο όρο, ο κάθε μηχανικός της εταιρίας γράφει 1000 πλήρως δοκιμασμένες και αποσφαλματωμένες γραμμές κώδικα το μήνα. Για τις κλάσεις του συστήματος ισχύει:

- 13 Interfaces * 300 γραμμές κώδικα/Interface = 3900 γραμμές κώδικα
- 5 Entities * 600 γραμμές κώδικα/Entity = 3000 γραμμές κώδικα
- 9 Controllers * 600 γραμμές κώδικα/Controller = 5400 γραμμές κώδικα
- 5 Proxies * 600 γραμμές κώδικα/Proxy = 3000 γραμμές κώδικα

Συνολικά 32 κλάσεις και περίπου 15500 γραμμές κώδικα. Η ομάδα αποτελείται από 4 προγραμματιστές, άρα μπορούν να γράφονται 4000 αποσφαλματωμένες γραμμές κώδικα το μήνα. Άρα $15500/4000 \approx 4$ ανθρωπομήνες για την συγγραφή κώδικα. Για τον έλεγχο του συστήματος θα χρειαστούν άλλοι 2 ανθρωπομήνες. Οπότε συνολικά θα χρειαστούν 6 ανθρωπομήνες για την διεκπεραίωση του έργου. Λαμβάνοντας υπόψιν σαββατοκύριακα και αργίες ο τελικός χρόνος είναι 8,5 μήνες.

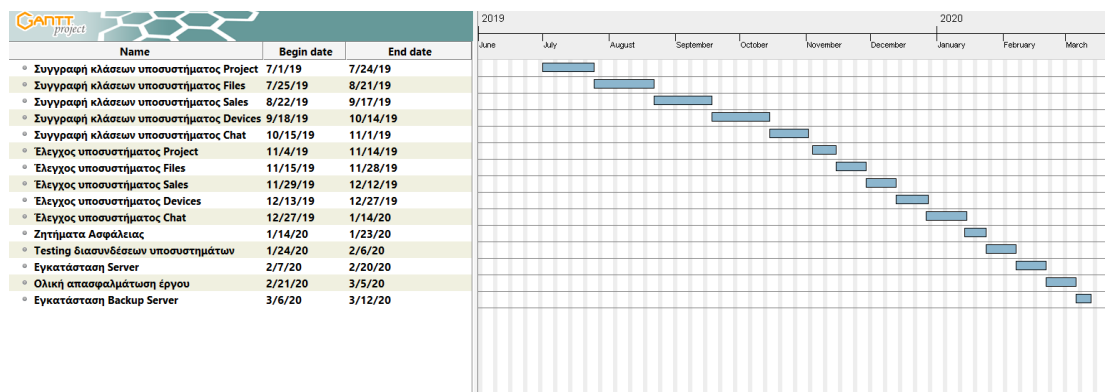
**Στο χρονοδιάγραμμα συμπεριλαμβάνονται και οι αργίες της 15^{ης} Αυγούστου και των Χριστουγέννων.

1	D.O.P.E.	Διάρκεια	Ημερομηνίες
2	Διάρκεια ολοκλήρωσης έργου		
3	Συγγραφή κλάσεων	89 days	1/7/2019 - 1/11/2019
4	Συγγραφή κλάσεων υποσυστήματος Project 3000 lines	18 days	24/7/2019
5	Συγγραφή κλάσεων υποσυστήματος Files 3300 lines	19 days	21/8/2019
6	Συγγραφή κλάσεων υποσυστήματος Sales 3300 lines	19 days	17/9/2019
7	Συγγραφή κλάσεων υποσυστήματος Devices 3300 lines	19 days	14/10/2019
8	Συγγραφή κλάσεων υποσυστήματος Chat 2400 lines	14 days	1/11/2019
9	Έλεγχος υποσυστημάτων	44 days	2/11/2019 - 13/1/2020
10	Έλεγχος υποσυστήματος Project	9 days	14/11/2019
11	Έλεγχος υποσυστήματος Files	10 days	28/11/2019
12	Έλεγχος υποσυστήματος Sales	10 days	12/12/2019
13	Έλεγχος υποσυστήματος Devices	10 days	27/12/2020
14	Έλεγχος υποσυστήματος Chat	5 days	13/1/2020
15	Υπόλοιπα ζητήματα	43 days	14/1/2020 - 12/3/2020
16	Ζητήματα Ασφάλειας	8 days	23/1/2020
17	Testing διασυνδέσεων υποσυστημάτων	10 days	6/2/2020
18	Εγκατάσταση Server	10 days	20/2/2020
19	Ολική απασφαλμάτωση έργου	10 days	5/3/2020
20	Εγκατάσταση Backup Server	5 days	12/3/2020
21			
22	Συνολικά	176 days	

Εικόνα 17: Χρονοδιάγραμμα

Διάγραμμα Gantt:

Λόγω του γεγονότος ότι θα δουλεύουμε μόνο εμείς οι 4, η συγγραφή κάθε κλάσης θα γίνεται ταυτόχρονα από όλους μας. Για αυτό και στο διάγραμμα Gantt φαίνεται πως τελειώνει ένα task για να ξεκινήσει ένα άλλο.



Εικόνα 18: Διάγραμμα Gantt

Κοστολόγηση έργου:

Η κοστολόγηση έγινε με βάση την παραδοχή ότι, ο κάθε μηχανικός της εταιρίας γράφει 1000 πλήρως δοκιμασμένες και απασφαλματωμένες γραμμές κώδικα το μήνα.

Για την ολοκλήρωση (integration) ενός συστήματος απαιτούνται $0,3 * N$ ανθρωπομήνες, όπου 0,3 μια εμπειρική σταθερά και N ο αριθμός των κλάσεων που υπάρχουν στο σύστημα.

Οι κλάσεις UI (Interfaces) αποτελούνται κατά μέσο όρο από 300 γραμμές κώδικα η κάθε μία, ενώ οι λειτουργικές κλάσεις (Entities, Controllers, Proxies) από 600 γραμμές κώδικα η κάθε μία.

Άρα το συνολικό κόστος προκύπτει από τους ανθρωπομήνες που χρειάζονται για την συγγραφή κώδικα, τον έλεγχο των υποσυστημάτων, την διευθέτηση των ζητημάτων ασφάλειας, το test των διασυνδεδεμένων υποσυστημάτων, την εγκατάσταση του server και του backup server και την ολική απασφαλμάτωση του συστήματος. Για την ολοκλήρωση του project χρειάζεται να καλυφθούν κάποια επιπλέον κόστη όπως αυτά της κτηριακής εγκατάστασης και του στησίματος και συντήρησης του server.

Οπότε:

- **Ανθρώπινο δυναμικό:** 6 ανθρωπομήνες * 4 άτομα * 2000 €/μήνα = 48000 €
- **Κτηριακές εγκαταστάσεις:** 5000€
- **Στήσιμο και συντήρηση Servers:**
 - ❖ Server: 2000€/έτος
 - ❖ Backup server: 1000€/έτος

Συνολικά 55000 €