



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΘΕΣΣΑΛΟΝΙΚΗΣ

Τμήμα Πληροφορικής

Πτυχιακή Εργασία

Σχεδίαση και Υλοποίηση Web Πλατφόρμας για την Υποβολή Επιστημονικών Εφαρμογών σε Κατανεμημένα Συστήματα

Ευφροσύνη Καλτιριμίδου

Επιβλέπων:

Αθηνά Βακάλη

Καθηγήτρια

Θεσσαλονίκη 2017

Περιεχόμενα

1	Front - End Architecture	4
1.1	Στόχοι της σχεδίασης	5
1.2	Τεχνικά χαρακτηριστικά	5
1.2.1	Εκδόσεις	6
1.3	Λειτουργίες Frontent	7
1.4	Σενάρια χρήσης - Use cases	7
1.4.1	Είσοδος - Login	7
1.4.2	Έλεγχος εργασιών - View jobs	13
1.4.3	Υποβολή νέας εργασίας - Submit job	16
2	Service layer - Restful API	20
2.1	Περιγραφή	21
2.2	Σχεδίαση του API	21
2.3	Είδη APIs	21
2.4	REST - RESTful API	22
2.4.1	Χαρακτηριστικά του REST	22

2.4.2	Από URL σε HTTP	23
-------	---------------------------	----

Κεφάλαιο 1

Front - End Architecture

Το frontend είναι το πιο σημαντικό κομμάτι της εφαρμογής για τον χρήστη. Αν η εφαρμογή ήταν ένα σπίτι, το frontend θα ήταν η εξωτερική πρόσοψη του σπιτιού και η οικοδεσπότης, ο οποίος καλωσορίζει τους ανθρώπους μέσα. Το frontend είναι το κομμάτι της εφαρμογής που χρησιμοποιεί ο χρήστης για να αποκτήσει, να μεταβάλλει και να αποθηκεύσει δεδομένα. Στην συγκεκριμένη εργασία, το frontend είναι αποκλειστικά μια web εφαρμογή. Όλη η επίδραση με τον χρήστη γίνεται μέσω του frontend, γιαυτό η σχεδίαση του αποτελεί το δεύτερο βήμα στην υλοποίηση της εφαρμογής. Η τεχνική υλοποίηση του θα γίνει μετά την τεχνική υλοποίηση του backend καθώς η λειτουργικότητα του στηρίζετε στα API calls που προσφέρει το backend.

1.1 Στόχοι της σχεδίασης

Οι αρχές που ακολουθεί η σχεδίαση του frontend χωρίζεται σε δύο ειδών, τις γενικές αρχές και τις αρχές ασφάλειας.

Γενικές αρχές Αναφέρονται σε όλο το frontend και στοχεύουν να κάνουν την εφαρμογή απλή και εύκολη για τον χρήστη.

Μερικές γενικές αρχές:

- Σε κάθε σελίδα του frontend, θα πρέπει να είναι ξεκάθαρες οι διαθέσιμες ενέργειες του χρήστη.
- Σε οποιαδήποτε σελίδα βρίσκετε ο χρήστης θα πρέπει να μπορεί να πάει σε οποιαδήποτε άλλη σελίδα της εφαρμογής με μία μόνο κίνηση.
- Για κάθε ενέργεια του χρήστη θα πρέπει να υπάρχει μια άμεση απάντηση της εφαρμογής, είτε με αλλαγή περιεχομένου, είτε με ενημερωτικό μήνυμα κ.τ.λ.
- Σε περίπτωση αποτυχίας κάποιας ενέργειας, θα πρέπει να ενημερώνετε με λεπτομέρεια ο χρήστης ώστε να γνωρίζει πως να συνεχίσει. Αυτό περιλαμβάνει την χρήση μηνυμάτων που θα εμφανίζονται στην οθόνη, το focus της σελίδας σε συγκεκριμένα πεδία ή δεδομένα κ.τ.λ.

Αρχές ασφάλειας Οι αρχές ασφάλειας αναφέρονται σε συγκεκριμένα κομμάτια του frontend και είναι κυρίως κομμάτι της ασφάλειας των δεδομένων.

- Η εφαρμογή θα χρησιμοποιεί πρωτόκολλα ασφάλειας για την μεταφορά των δεδομένων του χρήστη.
- Ο χρήστης θα παραμένει συνδεδεμένος για πεπερασμένο χρονικό διάστημα και στη συνέχεια θα πρέπει να ανανεώσει την σύνδεση του.

1.2 Τεχνικά χαρακτηριστικά

Οι τεχνολογίες που χρησιμοποιούνται σε κάθε εφαρμογή ποικίλουν ανάλογα με τα άτομα που την υλοποίησαν, τα χαρακτηριστικά και τις δυνατότητες της εφαρμογής.

Για τις web εφαρμογές, δύο τεχνολογίες αποτελούν αναπόσπαστα κομμάτια τους:

Hypertext Markup Language (HTML) Αποτελεί την γλώσσα σήμανσης που δημιουργεί τις σελίδες της εφαρμογής.

Cascading Style Sheets (CSS) Η οποία περιγράφει την μορφοποίηση του κάθε αντικειμένου στη σελίδα.

Σε συνδυασμό με μία client-based γλώσσα, όπως η Javascript, αποτελούν τα συστατικά στοιχεία κάθε web εφαρμογή σήμερα.

JavaScript (JS) Είναι μια client-based γλώσσα προγραμματισμού, η οποία δίνει δυνατότητα για δυναμικές ιστοσελίδες και εφαρμογές στο διαδίκτυο.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το web framework **AngularJS**, ένα δομικό framework για δυναμικές web εφαρμογές. Χρησιμοποιεί την HTML ως γλώσσα προτύπου και επεκτείνει τη σύνταξη της για να εκφράσει σαφώς και συνοπτικά τα στοιχεία της εφαρμογής. Διάφορες λειτουργίες της AngularJS μειώνουν ένα μεγάλο μέρος του κώδικα που θα έπρεπε να γραφτεί. Βασίζετε στη Javascript, και την επεκτείνει για να προσθέσει επιπλέον λειτουργίες προς διευκόλυνση του προγραμματιστή.

1.2.1 Εκδόσεις

Η εφαρμογή έχει υλοποιηθεί και δοκιμαστεί, στο κομμάτι του frontend με τις παρακάτω εκδόσεις:

HTML 5

CSS 3

JavaScript 5

AngularJS 1.6.*

1.3 Λειτουργίες Frontent

Στο frontend της εφαρμογής έχουν πρόσβαση όλοι οι χρήστες της *Ιδρυματικής Συστοιχίας ΑΠΘ* με τα στοιχεία πιστοποίησης του λογαριασμού τους στο *GRID*. Έχουν πρόσβαση στις παρακάτω λειτουργίες:

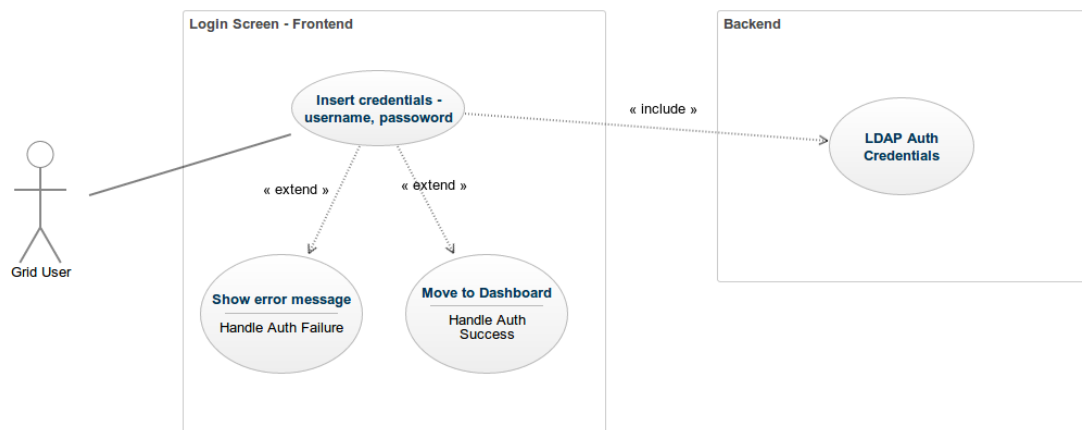
- **Σύνδεση** Ο χρήστης θα παρέχει το username και password που έχει πάρει κατά την εγγραφή του ως μέλος της *Ιδρυματικής Συστοιχίας ΑΠΘ*. Προσοχή, η εφαρμογή δεν παρέχει μέθοδο εγγραφής. Η εφαρμογή θα στείλει τα δεδομένα με ασφαλή πρωτόκολλα στο backend για επιβεβαίωση. Αν τα δεδομένα δεν είναι σωστά θα εμφανίσει το ανάλογο μήνυμα λάθους αλλιώς η εφαρμογή θα μεταβεί στο dashboard του χρήστη.
- **Έλεγχος εργασιών** Ο χρήστης μπορεί να δει τις εργασίες που έχει κάνει submit στον cluster, σε τι κατάσταση βρίσκονται και να τις φιλτράρει βάση κατάστασης, ημερομηνίας υποβολής και ονόματος.
- **Υποβολή εργασίας** Σε αυτή την λειτουργία ο χρήστης μπορεί να επιλέξει να υποβάλει το εκτελέσιμο της εργασίας από τον τοπικό του δίσκο ή να δημιουργήσει η εφαρμογή ένα εκτελέσιμο για αυτόν με βάση παραμέτρους που θα επιλέξει. Επίσης μπορεί να υποβάλει αρχεία εισόδου από τον τοπικό του δίσκο ή να δώσει το path που βρίσκονται μέσα στη συστοιχία.
- **Παραλαβή αποτελεσμάτων** Αν κάποια εργασία έχει εκτελεστεί επιτυχώς, μπορεί να επιλέξει να κατεβάσει τα αποτελέσματα της στον τοπικό του δίσκο μέσω της εφαρμογής.

1.4 Σενάρια χρήσης - Use cases

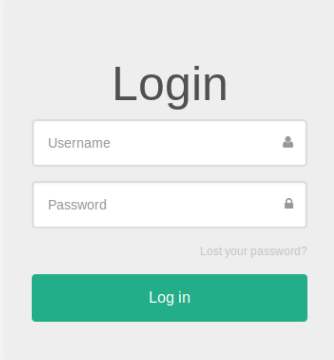
1.4.1 Είσοδος - Login

Οι χρήστες της *Ιδρυματικής Συστοιχίας ΑΠΘ* μπορούν να συνδεθούν στην εφαρμογή με τα στοιχεία που χρησιμοποιούν και για τις υπόλοιπες επιστημονικές υπηρεσίες που προσφέρει η συστοιχία. Η πρώτη σελίδα της εφαρμογής είναι η σελίδα του **Login**. Παρακάτω φαίνεται το σενάριο χρήσης για το **Login** καθώς και μια εικόνα της σελίδας από την εφαρμογή.

Σχήμα 1.1: Use case - Login

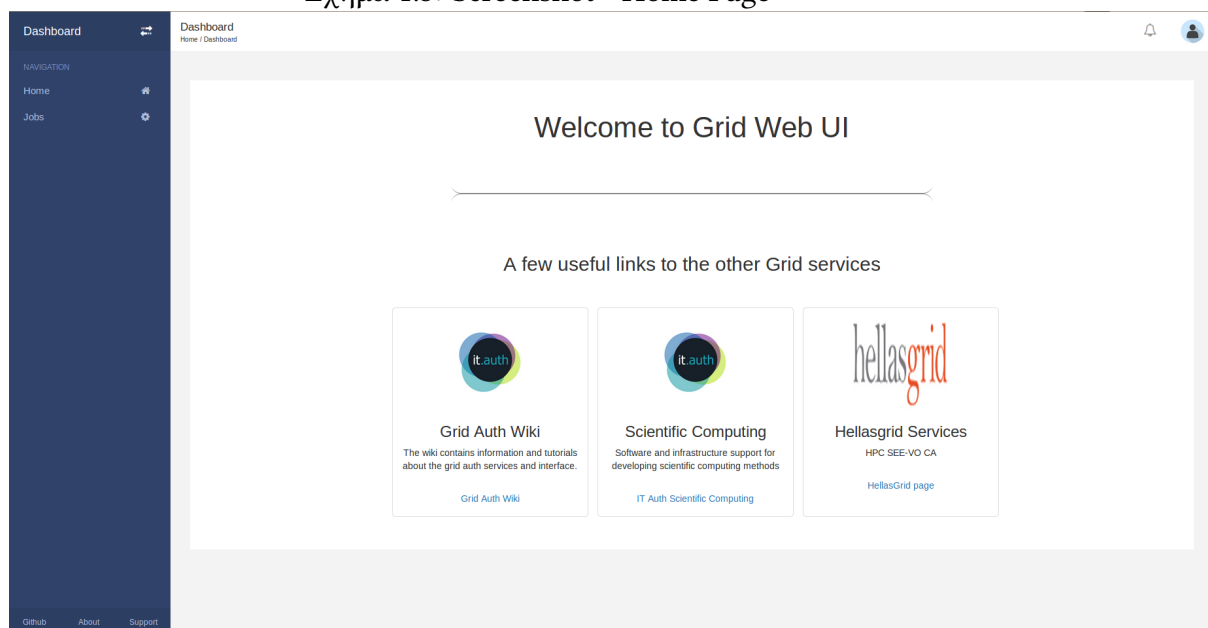


Σχήμα 1.2: Screenshot - Login



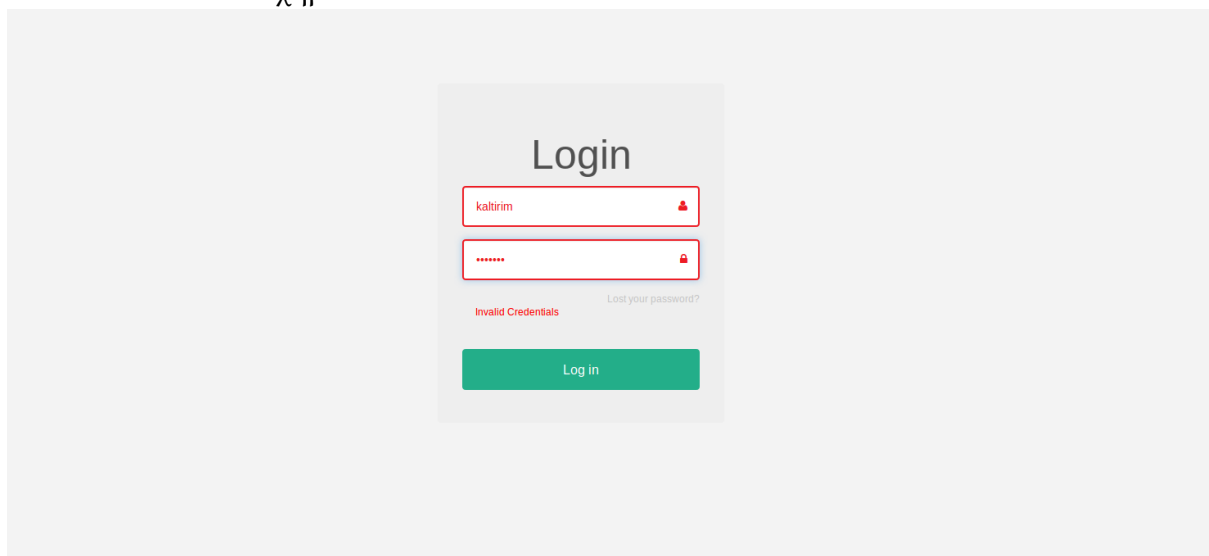
A screenshot of a login form centered on a light gray background. The form is contained within a slightly darker gray rectangular area. At the top of the form is the word "Login" in a large, bold, dark gray font. Below it are two input fields: the first is labeled "Username" with a small user icon to its right, and the second is labeled "Password" with a small lock icon to its right. Below the password field is a link that says "Lost your password?". At the bottom of the form is a green rectangular button with the text "Log in" in white.

Σχήμα 1.3: Screenshot - Home Page



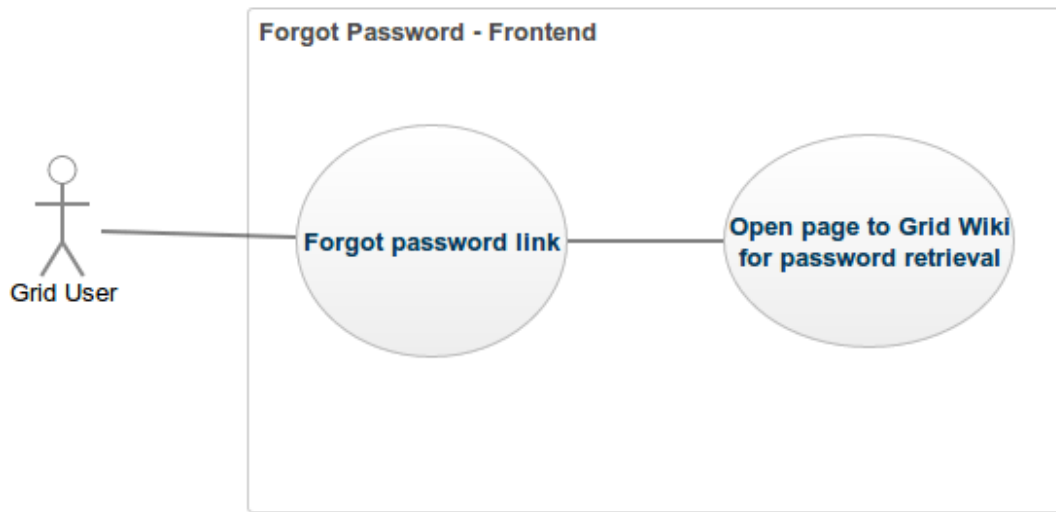
Μετά την επιτυχή επιβεβαίωση των στοιχείων, ο χρήστης μεταφέρετε στην κεντρική σελίδα της εφαρμογής.

Σχήμα 1.4: Screenshot - Invalid credentials



Αν τα στοιχεία του χρήστη είναι λανθασμένα θα του εμφανίσει ανάλογο μήνυμα και θα κοκκινίσει τα πεδία εισαγωγής.

Σχήμα 1.5: Use case - Forgot password



Σε περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό του, πατώντας τον σύνδεσμο *Forgot your password* θα ανοίξει ένα καινούργιο παράθυρο στον *browser* το οποίο αναλύει την διαδικασία ανάκτησης κωδικού, στο *Grid Wiki*.

1.4.2 Έλεγχος εργασιών - View jobs

Από το navigation menu στα δεξιά, της κάθε σελίδας, οι χρήστες μπορούν να μεταφερθούν στην σελίδα προβολής των εργασιών που έχουν υποβάλλει στη συστοιχία.

Αρχικά βλέπουν όλες τις εργασίες που έχουν υποβάλλει, και στη συνέχεια μπορούν να φιλτράρουν τις εργασίες για να δουν όποιες τους ενδιαφέρουν. Το φιλτράρισμα μπορεί να γίνει με βάση τρεις διαφορετικούς παράγοντες. Την κατάσταση της εργασίας, η οποία μπορεί να είναι:

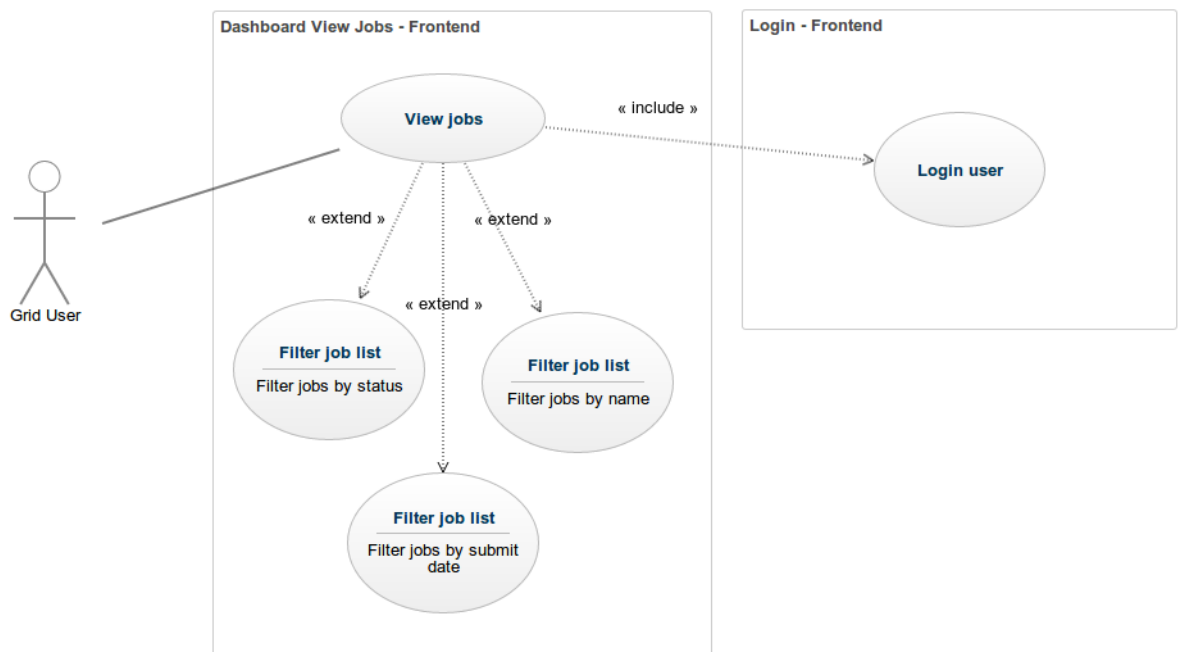
- Running - Η εργασία εκτελείται
- Finished - Η εργασία έχει ολοκληρωθεί επιτυχώς
- Stopped - Η εργασία έχει διακόψει την εκτέλεση της πριν ολοκληρωθεί

Την ημερομηνία υποβολής της εργασίας, από την πιο πρόσφατη έως την πιο παλιά και αντίστροφα.

Και τέλος αλφαβητικά με βάση το όνομα της εργασίας.

Τα φίλτρα μπορούν να χρησιμοποιηθούν και συνδυαστικά για να βγάλουν πιο συγκεκριμένα αποτελέσματα.

Σχήμα 1.6: Use case - View jobs



Σχήμα 1.7: Screenshot - View Jobs

Dashboard

NAVIGATION

- Home
- Jobs
- View jobs
- Submit job

View Jobs

Home / View Jobs

Filters

Status

- ☒ Running
- ☒ Finished
- ☒ Stopped

Status

- ☒ Running
- ☒ Finished
- ☒ Stopped

Status

- ☒ Running
- ☒ Finished
- ☒ Stopped

Jobs

#	Name	Status	Description	Submitted at
1	job1	running	a test job	07/02/2017 11:56
2	job2	running	a second test job	07/02/2017 12:04
4	4 job	finished	a fourth test job	13/02/2017 13:23
3	third-job	stopped	a third test job	13/02/2017 13:05

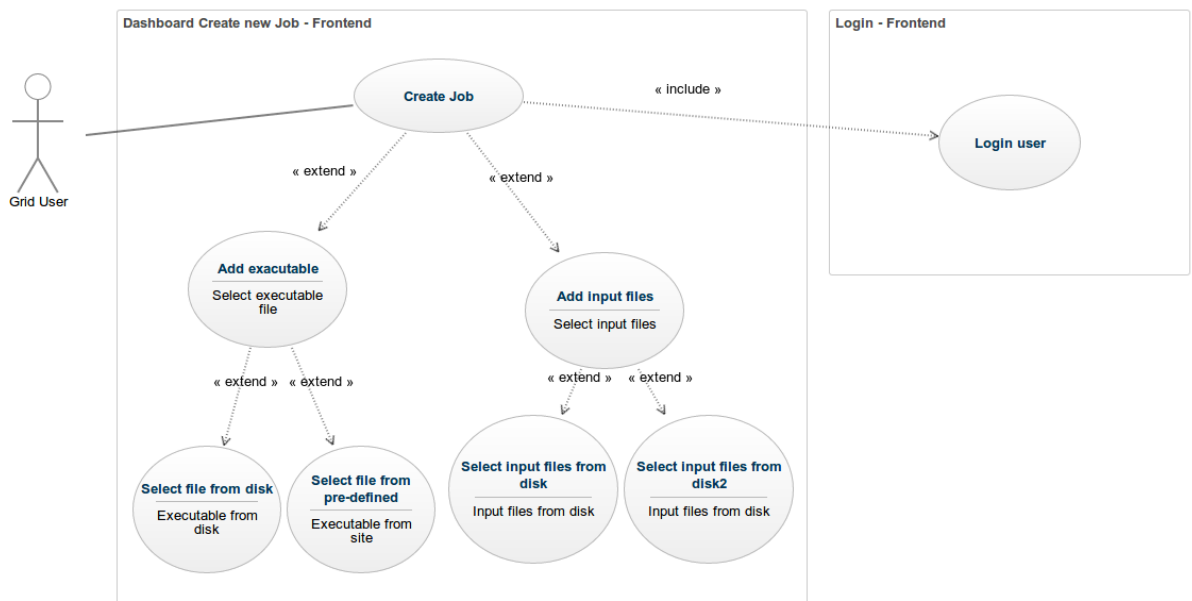
Github About Support

1.4.3 Υποβολή νέας εργασίας - Submit job

Από το αντίστοιχο μενού *Navigation > Jobs* στα δεξιά της σελίδας, ο χρήστης μπορεί να επιλέξει το *Submit job* για να υποβάλει μια νέα εργασία. Για να γίνει η υποβολή ο χρήστης θα πρέπει απαραίτητα να υποβάλει ένα εκτελέσιμο αρχείο. Επίσης μπορεί να προσθέσει αρχεία εισόδου για το πρόγραμμα ή να δώσει *file path* στο οποίο είναι αποθηκευμένα τα αρχεία εισόδου μέσα στην ιδρυματική συστοιχία. Τα αρχεία εισόδου που θα υποβάλει μέσα από το *web application* μπορούν να είναι περισσότερα από ένα, καθώς και ολόκληροι φάκελοι.

Στο κάτω μέρος της σελίδας μπορεί να δει ποια αρχεία έχει δηλώσει για υποβολή και να αφαιρέσει οποιοδήποτε δεν θέλει να συμπεριληφθούν στην εργασία.

Σχήμα 1.8: Use case - Create new job



Σχήμα 1.9: Screenshot - Create new job

Dashboard

NAVIGATION

Home

Jobs

View jobs

Submit job

Github

About

Support

Submit Jobs

Home / Submit Jobs

Executable

Drop executable file to upload

or

Browse

Selected executable file

Filename	Type	Size	Action
container-executor	Executable	106.263 kB	Remove

Submit job

Input files

Drop input file(s) to upload

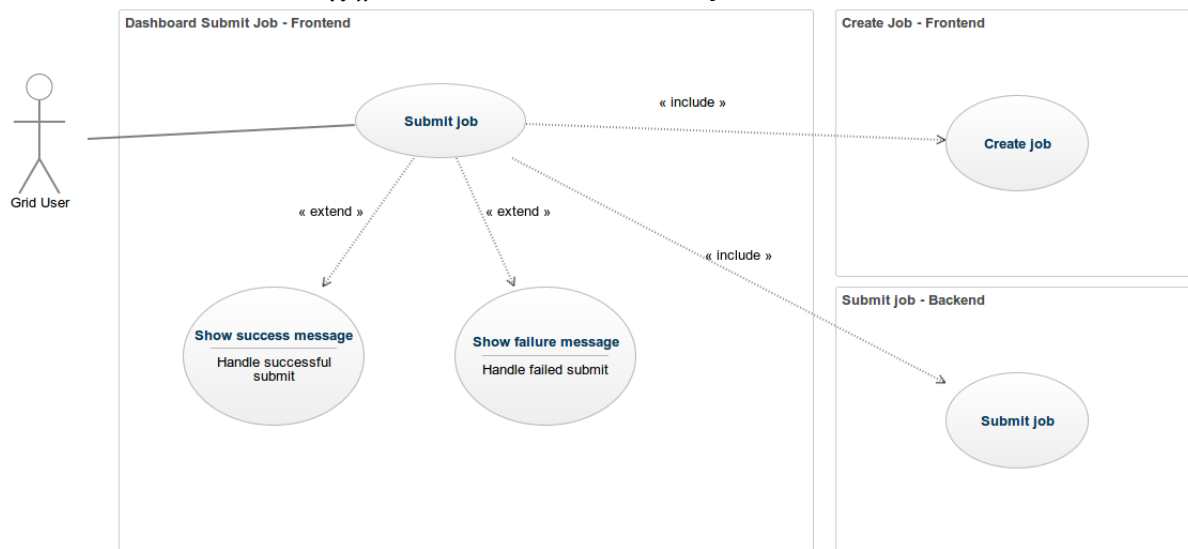
or

Browse

Selected input files

Filename	Type	Size	Action
capacity-scheduler.xml	Input	4.332 kB	Remove
configuration.xml	Input	1.304 kB	Remove
kms-actis.xml	Input	3.436 kB	Remove
			Remove all

Σχήμα 1.10: Use case - Submit job



Τέλος, αφού έχει υποβληθεί ένα εκτελέσιμο αρχείο, ο χρήστης μπορεί να υποβάλει την εργασία για εκτέλεση πατώντας το κουμπί *submit*.

Κεφάλαιο 2

Service layer - Restful API

Τα κομμάτια του frontend και του backend, παρόλο που είναι ανεξάρτητα το ένα από το άλλο, για την σωστή λειτουργία της εφαρμογής είναι απαραίτητο να επικοινωνήσουν μεταξύ τους. Την επικοινωνία τους αναλαμβάνει ένα τρίτο κομμάτι της εφαρμογής που ονομάζεται **Service Layer**.

Σε πολλές εφαρμογές, το **Service Layer** υλοποιήτε από το κομμάτι του backend καθώς οι ίδιες μέθοδοι που κάνουν την επικοινωνία backend με frontend κάνουν και την επικοινωνία backend με τη βάση δεδομένων. Και σε αυτήν την περίπτωση το **Service Layer** έχει υλοποιηθεί μαζί με το backend.

Σε γενικές γραμμές, πρόκειται για ένα σύνολο σαφώς καθορισμένων μεθόδων επικοινωνίας το οποίο για συντομία το λέμε **API (application programming interface)**. Το API ορίζει τον σωστό τρόπο για έναν προγραμματιστή να γράψει ένα πρόγραμμα που ζητά υπηρεσίες από λειτουργικό σύστημα (OS) ή άλλη εφαρμογή. Τα API υλοποιούνται με λειτουργικές κλήσεις που αποτελούνται από ρήματα και ουσιαστικά. Η απαιτούμενη σύνταξη περιγράφεται συνήθως στο documentation της κάθε εφαρμογής.

2.1 Περιγραφή

Τα API αποτελούνται από δύο στοιχεία. Το πρώτο είναι ένα σύνολο κανόνων που περιγράφει τον τρόπο ανταλλαγής πληροφοριών μεταξύ προγραμμάτων, που γίνεται με τη μορφή αίτησης για επεξεργασία και επιστροφής των απαραίτητων δεδομένων. Το δεύτερο είναι το λογισμικού γραμμένο με τις προδιαγραφές που ορίστηκαν από τους κανόνες και δημοσιεύεται για χρήση.

Το λογισμικό που επιθυμεί να αποκτήσει πρόσβαση στις δυνατότητες και τις δυνατότητες του API λέγεται ότι καλεί το API και το λογισμικό που υλοποιεί το API λέγεται ότι το δημοσιεύει.

2.2 Σχεδίαση του API

Ο σωστός σχεδιασμός του API βασίζεται στις αρχές της διαφάνειας και της ασφάλειας των δεδομένων. Το API θα πρέπει να αποκρύπτει τις μεθόδους και την πολυπλοκότητα της επικοινωνίας από και προς τις δύο υπηρεσίες. Έτσι, ο σχεδιασμός του API επιχειρεί να παρέχει μόνο τα εργαλεία που θα περίμενε να παραλάβει το frontend.

2.3 Είδη APIs

Υπάρχουν τριών ειδών APIs.

Local APIs Είναι η αρχική μορφή, από την οποία ήρθε το όνομα. Προσφέρουν υπηρεσίες OS ή middleware σε προγράμματα εφαρμογών. Το API της Microsoft .NET, το TAPI (API τηλεφωνίας) για φωνητικές εφαρμογές και τα API πρόσβασης σε βάσεις δεδομένων είναι Local API.

Web APIs Αυτά τα API έχουν σχεδιαστεί για να αντιπροσωπεύουν ευρέως χρησιμοποιούμενους πόρους όπως οι σελίδες HTML και έχουν πρόσβαση μέσω ενός απλού πρωτοκόλλου HTTP. Οποιαδήποτε διαδικτυακή διεύθυνση URL καλεί ένα web API. Τα API του Web ονομάζονται συχνά REST ή RESTful επειδή ο εκδότης των διασυνδέσεων REST δεν αποθηκεύει εσωτερικά δεδομένα μεταξύ αιτήσεων.

Ως εκ τούτου, τα αιτήματα πολλών χρηστών μπορούν να αναμειχθούν όπως και στο διαδίκτυο.

Program APIs Τα API προγραμμάτων βασίζονται σε τεχνολογία κλήσης εξ αποστάσεως (Remote Procedure Call) (RPC), που καθιστά ένα εξ αποστάσεως συστατικό του προγράμματος να φαίνεται τοπικό για το υπόλοιπο λογισμικό. Για παράδειγμα η σειρά API της Microsoft WS.

Καθώς η συγκεκριμένη εφαρμογή έχει επικοινωνία με web application έχει υλοποιηθεί ένα web API.

2.4 REST - RESTful API

Οι REST (Representational state transfer) ή RESTful υπηρεσίες, όπως η συγκεκριμένη, αποτελούν έναν τρόπο παροχής λειτουργικότητας μεταξύ συστημάτων στο Διαδίκτυο. Οι υπηρεσίες Web που είναι συμβατές με τους κανόνες που ορίζει το REST επιτρέπουν στα συστήματα να έχουν πρόσβαση και να χειρίζονται τα δεδομένα του διαδικτύου χρησιμοποιώντας ένα ομοιόμορφο και προκαθορισμένο σύνολο λειτουργιών. Υπάρχουν και άλλες μορφές υπηρεσιών με τους δικούς τους κανόνες και λειτουργίες, όπως το WSDL και το SOAP.

Σε μια RESTful υπηρεσία, τα αιτήματα (*requests*) που υποβάλλονται στο URL ενός πόρου θα προκαλέσουν μια απάντηση (*response*) που μπορεί να είναι XML, HTML, JSON ή κάποια άλλη καθορισμένη μορφή. Η απόκριση μπορεί να επιβεβαιώσει ότι έχουν γίνει ορισμένες αλλαγές στον αποθηκευμένο πόρο και μπορεί να παρέχει συνδέσεις υπερκειμένου με άλλους σχετικούς πόρους ή συλλογές πόρων. Χρησιμοποιώντας HTTP, όπως είναι πιο συνηθισμένο, το είδος των διαθέσιμων λειτουργιών περιλαμβάνει εκείνες που είναι προκαθορισμένες από τα ρήματα HTTP GET, POST, PUT, DELETE κ.τ.λ

2.4.1 Χαρακτηριστικά του REST

Οι κανόνες που ορίζονται από το REST εξυπηρετούν στην διασφάλιση ορισμένων πλεονεκτημάτων που ορίζουν τις RESTful υπηρεσίες όταν υλοποιούνται σωστά. Τα παρακάτω είναι μερικά από τα χαρακτηριστικά που προσπαθεί να διασφαλίσει το REST

- Performance

- Scalability
- Simplicity
- Modifiability
- Reliability

2.4.2 Από URL σε HTTP

Ένας από τους πιο βασικούς κανόνες που ορίζει το REST είναι η αντιστοίχιση των URL με συγκεκριμένα HTTP ρήματα και λειτουργίες. Ο παρακάτω πίνακας περιγράφει ένα από τα αντικείμενα της εφαρμογής, το *job* και πως αντιστοιχίζονται τα URL της web εφαρμογής με τα HTTP calls του API.

URL	col2	col3
http://backend/api/jobs/ 2	cell5 cell8	cell6 cell9

