



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΘΕΣΣΑΛΟΝΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

Τεχνητής Νοημοσύνης

Πτυχιακή Εργασία

Δημιουργία περιεχομένου για Video Games με την χρήση μεθόδων Μηχανικής Μάθησης

Ευφροσύνη Καλτιριμίδου

Επιβλέπων:

Νίκος Νικολαΐδης

Καθηγητής

Θεσσαλονίκη 2020

Περιεχόμενα

1	Procedural Content Generation	5
1.1	Game Content	6
1.2	Παιχνίδια που χρησιμοποιούν PCG	7
1.3	Τεχνικά χαρακτηριστικά	7
1.3.1	Εκδόσεις	8
1.4	Λειτουργίες Frontent	9
1.5	Σενάρια χρήσης - Use cases	9
1.5.1	Είσοδος - Login	9
1.5.2	Έλεγχος εργασιών - View jobs	15
1.5.3	Υποβολή νέας εργασίας - Submit job	18
2	Service layer - Restful API	22
2.1	Περιγραφή	23
2.2	Σχεδίαση του API	23
2.3	Είδη APIs	23
2.4	REST - RESTful API	24

2.4.1	Χαρακτηριστικά του REST	24
2.4.2	Από URL σε HTTP	25

Κεφάλαιο 1

Procedural Content Generation

Το αντικείμενο του Procedural Content Generation (PCG) όπως ορίζεται στο [1] είναι η *Δημιουργία περιεχομένου για ηλεκτρονικά παιχνίδια με την χρήση αλγορίθμων και με την παροχή ελάχιστων ή καθόλου εισόδων από τον χρήστη*. Αποτελεί μια μεγάλη κατηγορία έρευνας και ανάπτυξης τόσο στην βιομηχανία των παιχνιδιών όσο και στην επιστήμη της πληροφορικής. Το PCG, όπως και πολλά άλλα αντικείμενα της πληροφορικής, αντιπροσωπεύεται από ιδιαίτερα προβλήματα και περιορισμούς τόσο στην πολυπλοκότητα των αλγορίθμων όσο και στην αυθεντικότητα και προτοτυπία των αποτελεσμάτων. Ως γνωστό αντικείμενο της επιστήμης της πληροφορικής, μπορεί να ταξινομηθεί κάτω από την κατηγορία της Τεχνητής Νοημοσύνης καθώς βασικός στόχος του PCG είναι η δημιουργία αλγορίθμων που μπορούν να προσομοιώσουν την ανθρώπινη δημιουργικότητα και ευφύια. Επιπλέον τα προβλήματα και οι προσεγγίσεις επίλυσης παρουσιάζουν πολλά κοινά με άλλα πεδία της Τεχνητής Νοημοσύνης.

Όπως και με πολλά άλλα αντικείμενα της Τεχνητής Νοημοσύνης, έτσι και το PCG είχε μικρή εξάπλωση και χρήση στην αρχή. Αυτό δεν οφείλεται στις δυνατότητες των αλγορίθμων αλλά στην αδυναμία του hardware των υπολογιστών εκείνων των περιόδων να χρησιμοποιήσουν αλγόριθμους τέτοιας πολυπλοκότητας. Τα τελευταία χρόνια με την ανάπτυξη των δυνατοτήτων των προσωπικών υπολογιστών και των κινητών συσκευών έχει διευρυνθεί η χρήση μεθόδων PCG για την παραγωγή διαφόρων ειδών game content όπως θα αναλυθεί στη συνέχεια. Μαζί με την αύξηση στην χρήση μεθόδων PCG ήρθε και η αύξηση των προβλημάτων που καλείται να επιλύσει.

1.1 Game Content

Για να κατανοήσουμε καλύτερη το πεδίο του PCG πρέπει πρώτα να καταλάβουμε τι θεωρείται περιεχόμενο σε ένα παιχνίδι. Ο τομέας του PCG αναφέρεται και έχει χρησιμοποιηθεί σε ηλεκτρονικά παιχνίδια (Video Games), επιτραπέζια παιχνίδια (Board Games), παιχνίδια με κάρτες (Card Games) κ.ά. Στην συγκεκριμένη εργασία, επικεντρωθήκαμε στα Video Games.

Game Content Όπως αναφέρεται και στο όνομα, περιεχόμενο, είναι κάτι που περιέχεται σε ένα παιχνίδι. Αυτός ο ορισμός όμως είναι πολύ γενικός και ευρύς και δεν περιορίζετε μόνο στο περιεχόμενο που αντιστοιχεί στο PCG. Στην βιβλιογραφία μπορούμε να ξεχωρίσουμε συγκεκριμένα είδη περιεχομένου που φαίνεται να μπορούν να δημιουργηθούν με μεθόδους PCG. Αυτά είναι:

- Γραφικά (Textures)
- Επίπεδα (Levels) και χάρτες (Maps)
- Αντικείμενα (Items)
- Αποστολές (Quests)
- Ιστορίες (Stories)
- Μουσική (Music)

Ο παραπάνω διαχωρισμός γίνεται με βάση το είδος του κάθε περιεχομένου, για παράδειγμα η μουσική ως περιεχόμενο ενός παιχνιδιού, διαφέρει από τα αντικείμενα. Αντίστοιχα οι μεθοδολογίες που έχουν αναπτυχθεί για την παραγωγή μουσικής διαφέρουν από τις μεθόδους που χρησιμοποιούνται για την παραγωγή αντικειμένων. Αυτό βέβαια δεν σημαίνει ότι δεν υπάρχουν κοινοί αλγόριθμοι που μπορούν να εφαρμοστούν σε παραπάνω από ένα είδος με επιτυχία, αντίθετα υπάρχουν πολλοί αλγόριθμοι που έχουν ως βάση έναν πιο γενικό αλγόριθμο και αποτελούν ειδικές εκδόσεις του για το κάθε είδος περιεχομένου. Ο διαχωρισμός του περιεχομένου βοηθάει στην καλύτερη κατανόηση των ιδιοτήτων και των περιορισμών που εμφανίζει το κάθε είδος το οποίο οδηγεί στην δημιουργία καλύτερων μεθόδων και αλγορίθμων.

1.2 Παιχνίδια που χρησιμοποιούν PCG

Από την πρώτη στιγμή που ξεκίνησε η διάδοση των Video Games φάνηκε η ανάγκη για την αυτόματη και αυτόνομη δημιουργία σωστού περιεχομένου. Κάποια από τα παιχνίδια που εφάρμοσαν με μεγάλη επιτυχία μεθόδους PCG είναι:

Rogue (1980) Ένα από τα πρώτα παιχνίδια που εφάρμοσε PCG για την αυτόματη δημιουργία επιπέδων (**levels**). Το Rogue ενέπνευσε την δημιουργία πολλών παιχνιδιών με αντίστοιχες δυνατότητες και PCG μεθόδους.

Spore (2008) Το Spore είναι ένα life-simulation και strategy παιχνίδι που χρησιμοποιεί PCG για την δημιουργία πλασμάτων και αντικειμένων. Οι αλγόριθμοι του Spore, συνδυάζουν απλά σχήματα και αντικείμενα για να δημιουργήσουν μεγαλύτερα και πιο πολύπλοκα πλάσματα με βάση διάφορους κανόνες και περιορισμούς.

No Man's Sky (2016) Ένα από τα πιο σημαντικά παραδείγματα για τις δυνατότητες του PCG είναι το παιχνίδι no Man's Sky. Το θέμα του παιχνιδιού είναι η εξερεύνηση του διαστήματος και η επιβίωση σε ξένους πλανήτες. Το παιχνίδι δημιουργεί σχεδόν ολόκληρο των κόσμο με PCG, αυτό περιλαμβάνει τους πλανήτες, τα αστέρια, τα φυτά, τα ζώα και τα encounters του παίκτη με τη χρήση ντετερμινιστικών αλγορίθμων PCG.

1.3 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.

1.3.1 Εκδόσεις

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

HTML 5

CSS 3

JavaScript 5

AngularJS 1.6.*

1.4 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:

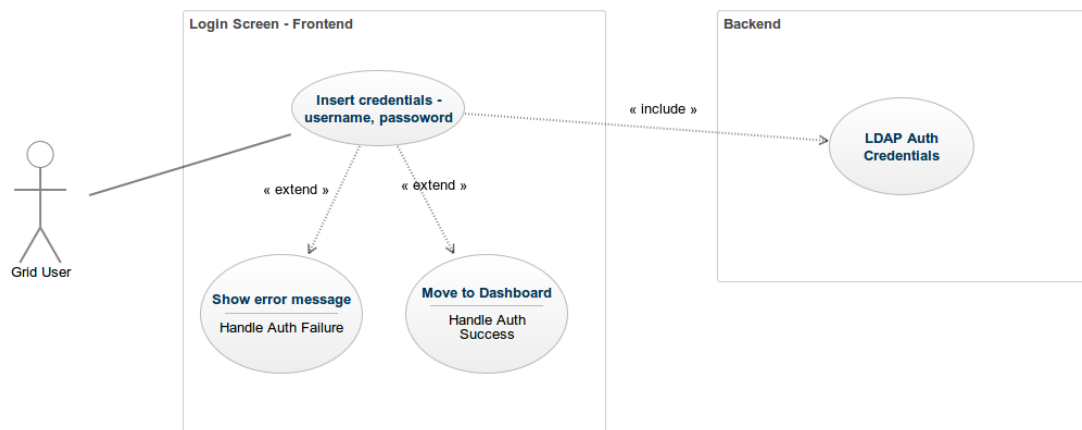
- **Σύνδεση** Ο χρήστης θα παρέχει το username και password που έχει πάρει κατά την εγγραφή του ως μέλος της *Ιδρυματικής Συστοιχίας ΑΠΘ.Προσοχή* , η εφαρμογή δεν παρέχει μέθοδο εγγραφής. Η εφαρμογή θα στείλει τα δεδομένα με ασφαλή πρωτόκολλα στο backend για επιβεβαίωση. Αν τα δεδομένα δεν είναι σωστά 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.

1.5 Σενάρια χρήσης - Use cases

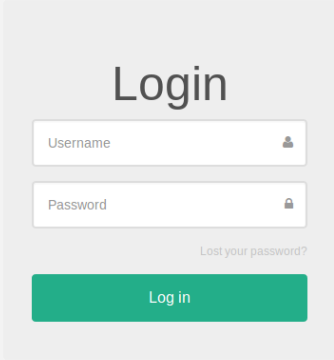
1.5.1 Είσοδος - Login

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

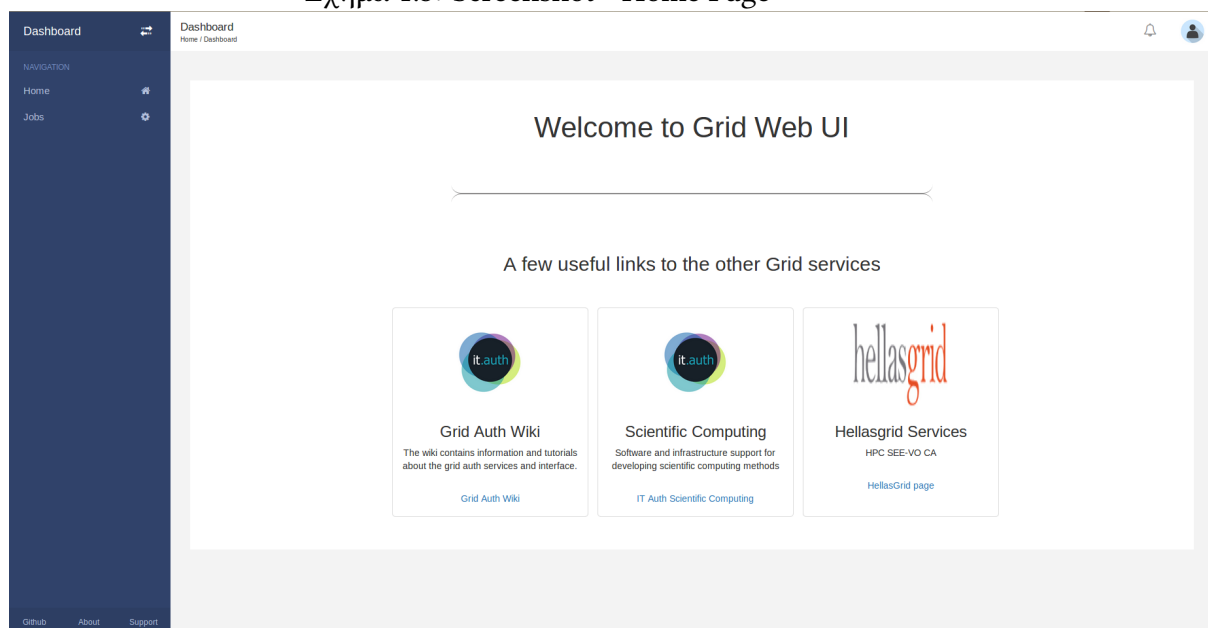


Σχήμα 1.2: Screenshot - Login



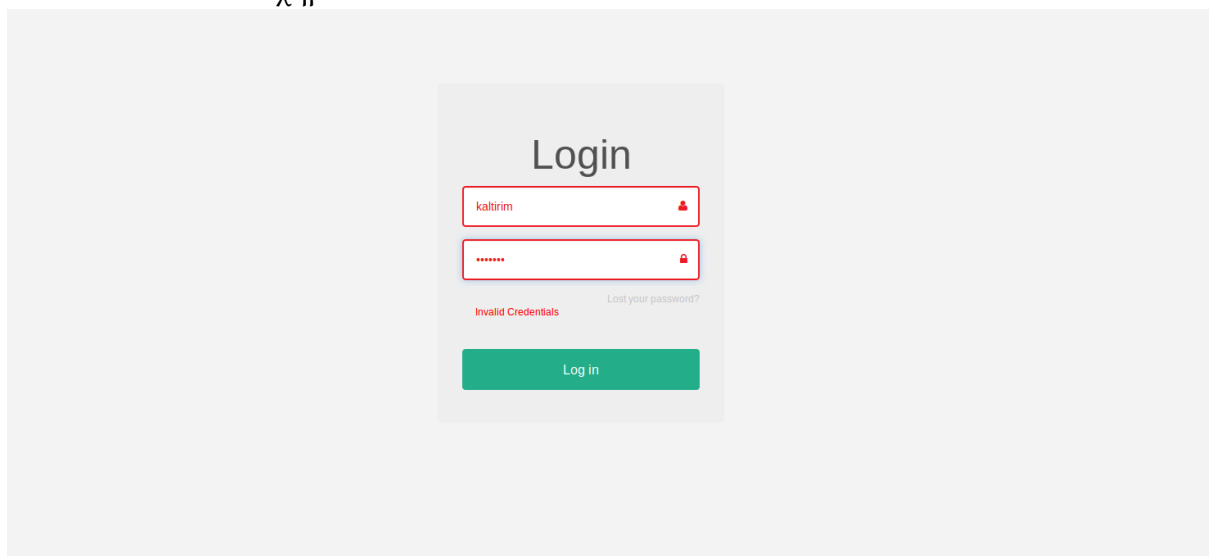
A screenshot of a login form centered on a light gray background. The form itself is a white rectangle with a thin gray border. At the top of the form, the word "Login" is displayed in a large, bold, dark gray font. Below the title, there are two input fields. The first field is labeled "Username" in a small, gray font and has a small gray user icon on its right side. The second field is labeled "Password" in a small, gray font and has a small gray lock icon on its right side. Below the password field, the text "Lost your password?" is written in a very small, gray font. At the bottom of the form, there is a green rectangular button with the text "Log in" in white.

Σχήμα 1.3: Screenshot - Home Page



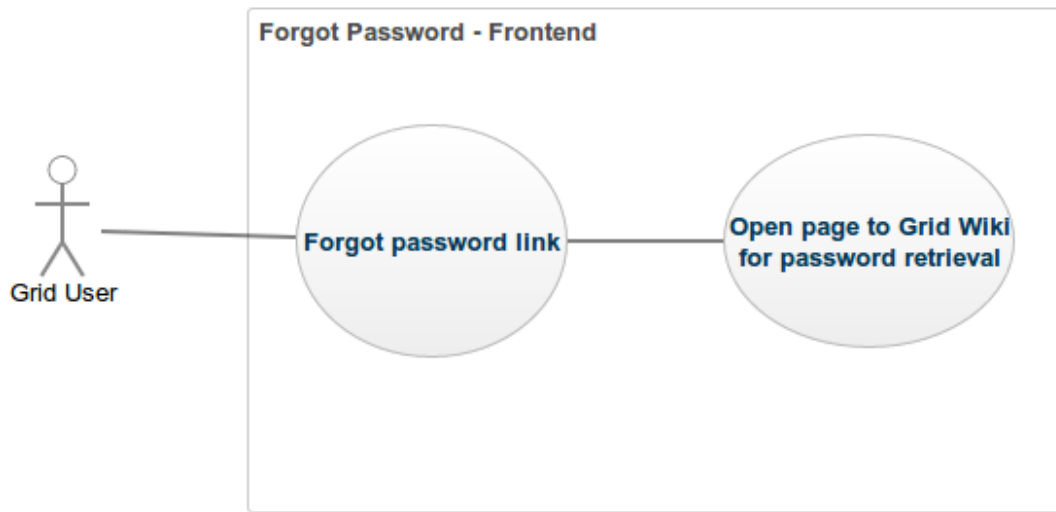
Μετά την επιτυχή επιβεβαίωση των στοιχείων, ο χρήστης μεταφέρετε στην κεντρική σελίδα της εφαρμογής.

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*.

1.5.2 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:

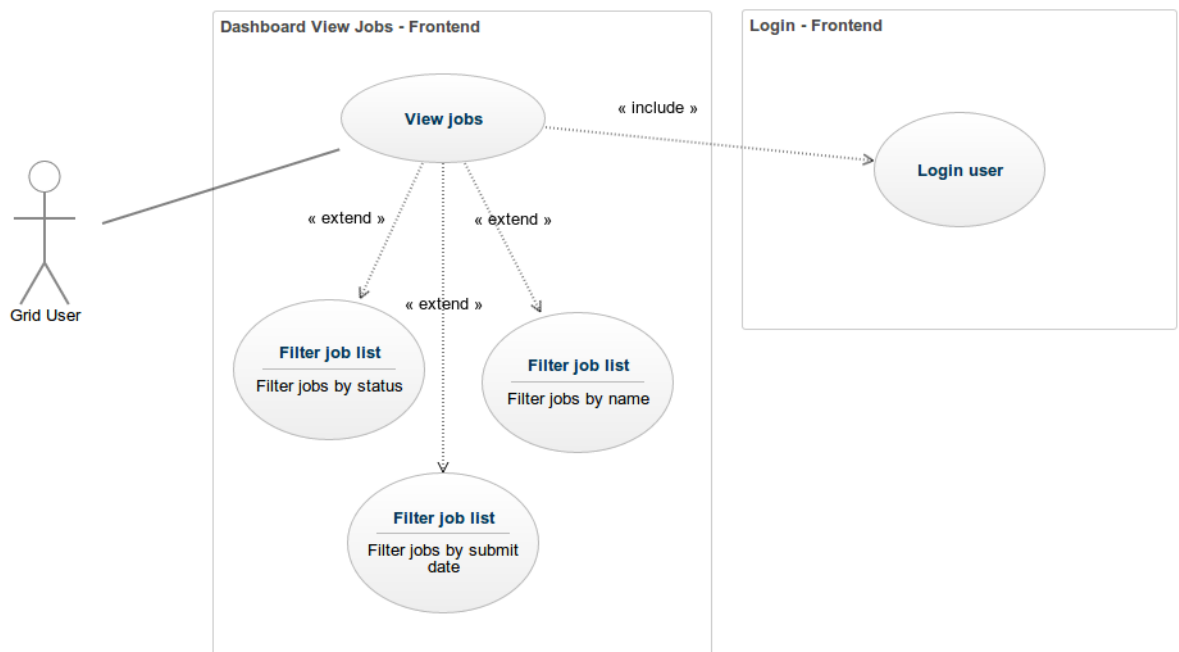
- Running - Η εργασία εκτελείται
- Finished - Η εργασία έχει ολοκληρωθεί επιτυχώς
- Stopped - Η εργασία έχει διακόψει την εκτέλεση της πριν ολοκληρωθεί

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

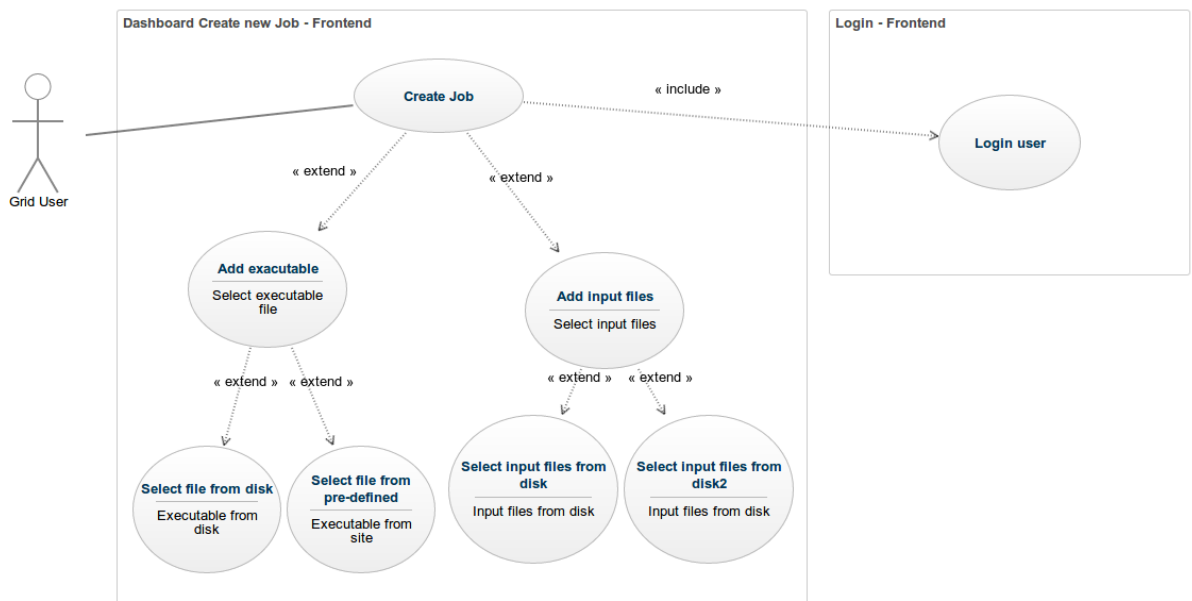
Github About Support

1.5.3 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

Input files

Drop input file(s) to upload

or

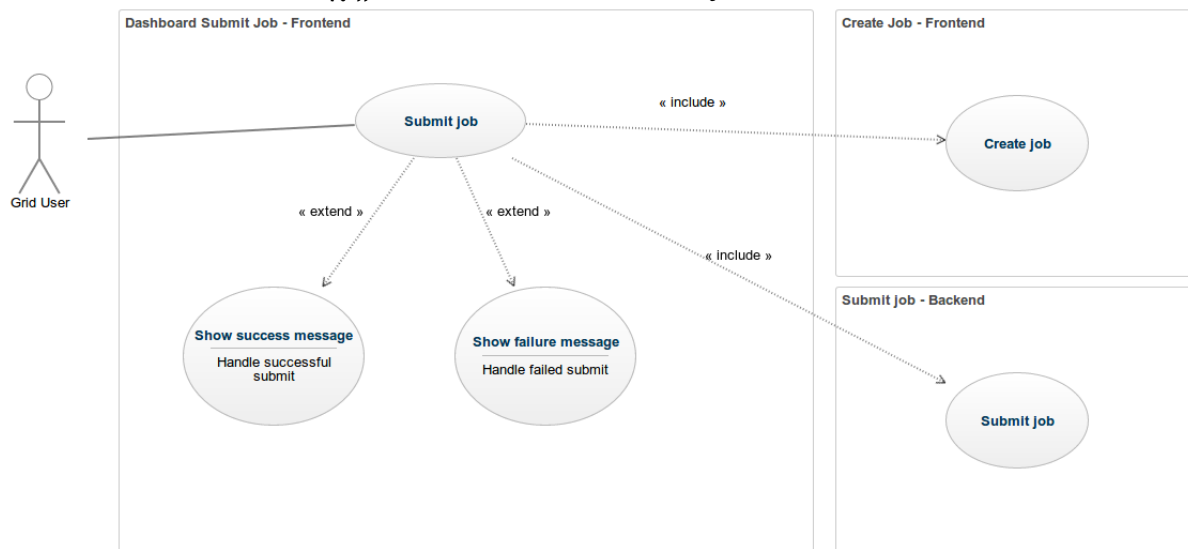
Browse

Selected input files

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

Submit job

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	HTTP GET	col3
http://backend/api/jobs/	cell5	cell6
2	cell8	cell9

