

GenOn Web App

Στα πλαίσια του μαθήματος:
“Αρχιτεκτονική Εφαρμογών Διαδικτύου και Βιοπληροφορικής”

Βενετσιάνου Νεφέλη Κλεοπάτρα, AM: 7113132100019

Γεωργακόπουλος Ηλίας, AM: 7113132100020

Θεοχάρης Ιορδάνης, AM: 7113132100006

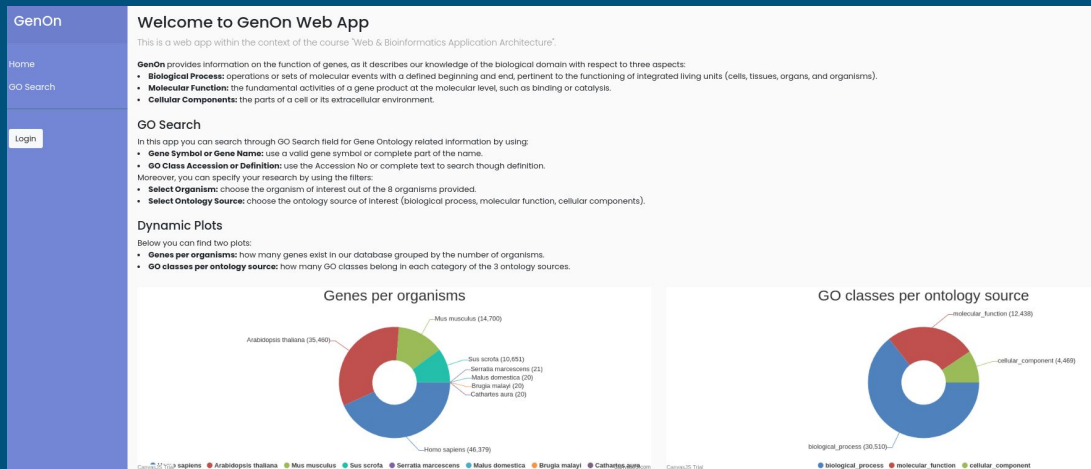
Διδάσκων: Δρ. Ι. Χαμόδρακας

Οντολογία Γονιδίων

- Πληροφορίες σχετικά με γονίδια, τις λειτουργίες και τις σχέσεις μεταξύ τους με ένα δομημένο τρόπο.
- Διακρίνεται σε 3 κύριους τύπους:
 - **Μοριακή Λειτουργία** (Molecular Function): δραστηριότητες μοριακού επιπέδου
 - **Βιολογικές Διαδικασίες** (Biological Process): μεγαλύτερες διεργασίες - πολλαπλές μοριακές δραστηριότητες.
 - **Κυτταρικά Συστατικά** (Cellular Components): κυτταρικές θέσεις εκτέλεσης μίας λειτουργίας/διεργασίας.

GenOn

- Αναζήτηση βάσει Γονιδίων ή Οντολογίας
- Επιλογή συγκεκριμένου οργανισμού ή οντολογίας
- Πλοήγηση στα δεδομένα
- 2 ρόλοι χρηστών:
 - Administrator -> εισαγωγή, επεξεργασία και διαγραφή εγγραφών
 - Anonymous -> περιήγηση
- Δυναμικά Διαγράμματα

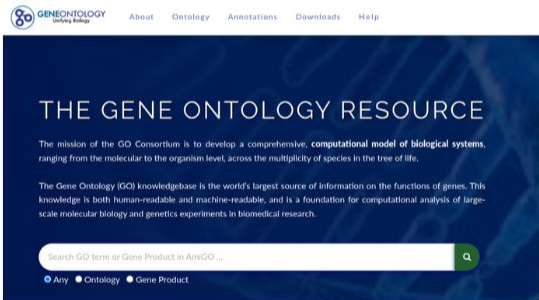


Σχεδιασμός

1. Λήψη Δεδομένων

2. Σχεδιασμός Βάσης Δεδομένων

3. Δημιουργία Εφαρμογής



1. Λήψη Δεδομένων: Gene Ontology Resources

- **Ontology (GO term):**

- Biological Process
- Molecular Function
- Cellular Component

} 47,417 data

- **Genes & Gene products:**

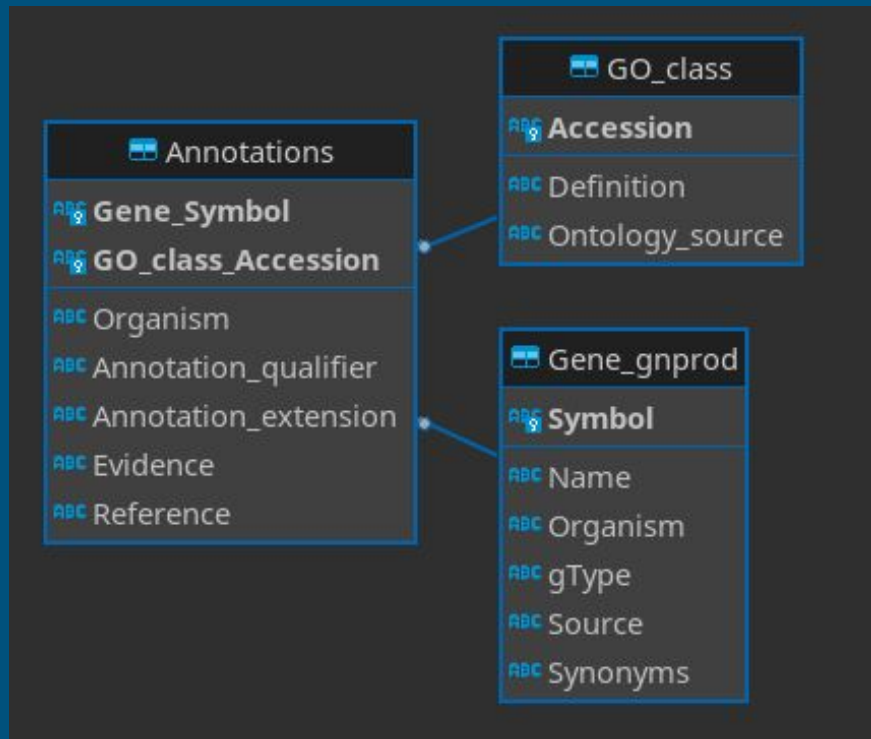
- *Homo sapiens*
- *Mus musculus*
- *Arabidopsis thaliana*
- *Sus scrofa*
- *Malus domestica*
- *Cathartes aura*
- *Serratia marcescens*
- *Brugia malayi*

} 8 organisms -> 107,271 data

- **Annotations** (statement about the function of a particular gene - associating a gene or gene product with a GO term):

- same 8 organisms -> 253,843 data

2. Σχεδιασμός Βάσης Δεδομένων MySQL: Μοντέλο Οντοτήτων Συσχετίσεων

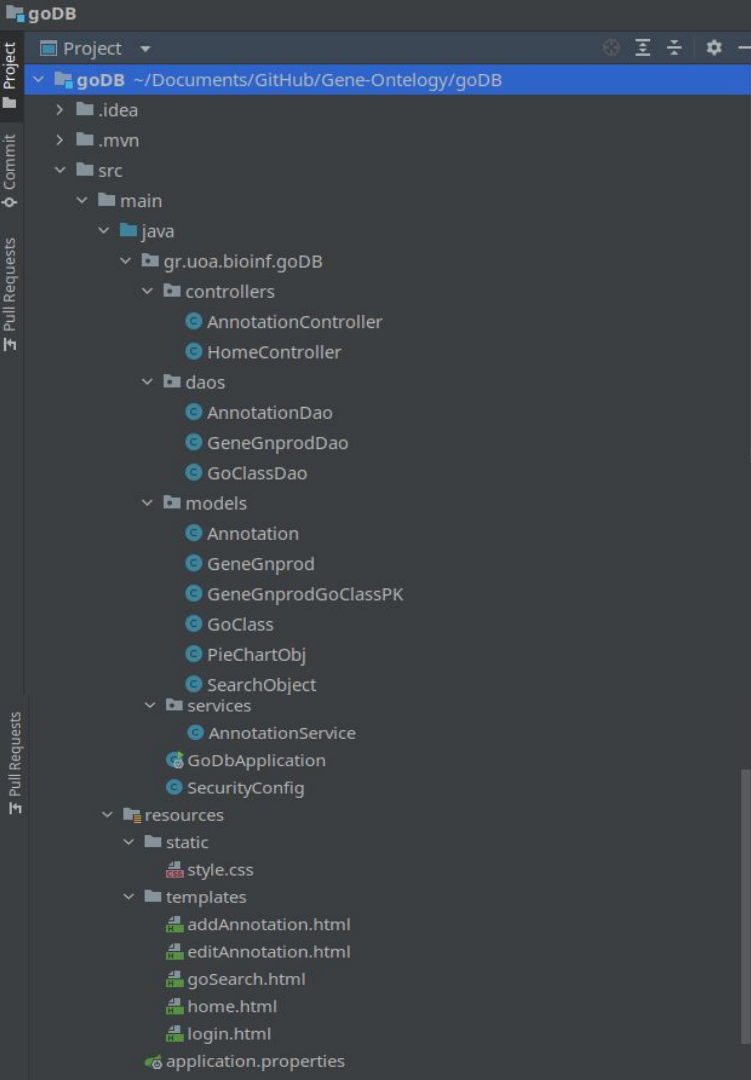


3. Δημιουργία Εφαρμογής & Υλοποίηση

- MVC Architecture (model-viewer-controller)
- Technologies used:
 - **Spring Boot Framework:** εργαλείο κατασκευής Web Apps με ελάχιστη παραμετροποίηση
 - **jdbc:** dependency για σύνδεση με ΒΔ
 - **Spring Security:** χρήση για δημιουργία ρόλων & παραμετροποίηση προσβασιμότητας
 - **Thymeleaf:** παροχή templates - χρήση για HTML, CSS, JavaScript
 - **MySQL:** δημιουργία ΒΔ
 - **HTML:** δημιουργία UI
 - **CSS:** style UI
 - **Jquery:** δημιουργία δυναμικών διαγραμμάτων

Languages





MVC

- **Models:**
 - 1-1 με τους πίνακες της ΒΔ
 - Ονόματα πινάκων & πεδίων-στηλών
- **Views:**
 - HTML
 - CSS
- **Controllers:**
 - endpoints
- **DAO (CRUD):**
 - Διασύνδεση με ΒΔ
 - Create
 - Read (search/get)
 - Update
 - Delete
- **Services**
 - helper methods & business logic

CriteriaBuilder: Δυναμική Αναζήτηση & Δημιουργία query

περιέχει προκαθορισμένες μεθόδους που χρησιμοποιούνται για τον καθορισμό queries για την ανάκτηση των εγγραφών/οντοτήτων

```
public List search(SearchObject searchObject) {
    CriteriaBuilder cb = entityManager.getCriteriaBuilder();
    CriteriaQuery<Annotation> cq = cb.createQuery(Annotation.class);
    Root<Annotation> annotationRoot = cq.from(Annotation.class);
    Join<Annotation, GoClass> annotationGoClass = null;
    List<Predicate> orPredicates = new ArrayList<>();
    List<Predicate> andPredicates = new ArrayList<>();
    Predicate finalQuery;
    cq.select(annotationRoot);

    if(!StringUtils.isEmpty(searchObject.getGeneTerm())) {
        String geneTerm = "%" + searchObject.getGeneTerm() + "%";
        Join<Annotation, GeneEnprod> annotationGene = annotationRoot.join( attributeNames: "geneEnprod");
        orPredicates.add(cb.like(annotationRoot.get("geneSymbol"), geneTerm));
        orPredicates.add(cb.like(annotationGene.get("name"), geneTerm));
    }

    if(!StringUtils.isEmpty(searchObject.getGoClassTerm())) {
        String goClassTerm = "%" + searchObject.getGoClassTerm() + "%";
        annotationGoClass = annotationRoot.join( attributeNames: "goClass");
        orPredicates.add(cb.like(annotationRoot.get("goClassAccession"), goClassTerm));
        orPredicates.add(cb.like(annotationGoClass.get("definition"), goClassTerm ));
    }

    if(!StringUtils.isEmpty(searchObject.getOrganism()) && !"0".equals(searchObject.getOrganism())) {
        andPredicates.add(cb.equal(annotationRoot.get("organism"), searchObject.getOrganism()));
    }
}
```

```
if(!StringUtils.isEmpty(searchObject.getOntologySource()) && !"0".equals(searchObject.getOntologySource())) {
    if (annotationGoClass == null ) {
        annotationGoClass = annotationRoot.join( attributeNames: "goClass");
    }
    andPredicates.add(cb.equal(annotationGoClass.get("ontologySource"), searchObject.getOntologySource()));
}

Predicate orQueries = cb.or(orPredicates.toArray(new Predicate[0]));
Predicate andQueries = cb.and(andPredicates.toArray(new Predicate[0]));

if(orPredicates.isEmpty() && !andPredicates.isEmpty()) {
    finalQuery = andQueries;
    cq.where(finalQuery);
}

if(!orPredicates.isEmpty() && andPredicates.isEmpty()) {
    finalQuery = orQueries;
    cq.where(finalQuery);
}

if(!orPredicates.isEmpty() && !andPredicates.isEmpty()) {
    finalQuery = cb.and(orQueries, andQueries);
    cq.where(finalQuery);
}
```

Spring Security

SecurityFilterChain:

- επιλογή πεδίων που εμφανίζονται σε anonymous και admin
- εμφάνιση πεδίων “Add Annotation”, “Edit”, “Delete” μόνο στον admin

```
@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .authorizeHttpRequests((requests) -> requests
                .requestMatchers(...patterns: "/", "/annotation/search", "/login", "/style.css").permitAll()
                .requestMatchers(...patterns: "/annotation/add", "/annotation/edit/**", "/annotation/persist",
                    "/annotation/update", "/annotation/delete/**").hasRole("ADMIN"))
            .formLogin((form) -> form
                .loginPage("/login")
                .permitAll())
            .logout((logout) -> logout.permitAll());
        return http.build();
    }
}
```

jquery

Δημιουργία δυναμικών διαγραμμάτων (CanvasJSChart - charting library)

```
<script th:inline="javascript">
    window.onload = function() {
        let genesGrouped = /*[[${genesGrouped}]]*/ {};
        let goClassesGrouped = /*[[${goClassesGrouped}]]*/ {};

        let dataPointsGenes = [];
        for (d in genesGrouped) {
            dataPointsGenes.push({
                'label': genesGrouped[d][0],
                'y': genesGrouped[d][1]
            });
        }

        let dataPointsGoClasses = [];
        for (d in goClassesGrouped) {
            dataPointsGoClasses.push({
                'label': goClassesGrouped[d][0],
                'y': goClassesGrouped[d][1]
            });
        }
    }
}
```

```
var optionsGenes = {
    title: {
        text: "Genes per organisms"
    },
    data: [{
        type: "doughnut",
        innerRadius: "40%",
        showInLegend: "true",
        legendText: "{label}",
        indexLabel: "{label} ({y})",
        yValueFormatString: "#,##0.##%",
        dataPoints: dataPointsGenes
    }]
};

$("#chartGenes").CanvasJSChart(optionsGenes);

var optionsGoClasses = {
    title: {
        text: "GO classes per ontology source"
    },
    data: [{
        type: "doughnut",
        innerRadius: "40%",
        showInLegend: "true",
        legendText: "{label}",
        indexLabel: "{label} ({y})",
        yValueFormatString: "#,##0.##%",
        dataPoints: dataPointsGoClasses
    }]
};

$("#chartGoClass").CanvasJSChart(optionsGoClasses);
</script>
```

Demo

—

Σας ευχαριστούμε πολύ για την προσοχή σας !

Q&A