

HERMES



A SIMPLE OPEN-SOURCE APPOINTMENT SCHEDULER

ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Εσωτερικό Σύστημα:

- Οι Διαχειριστές/στριες Οργανισμών μπορούν:
 - Να διαχειρίζονται τον κατάλογο οργανισμών του συστήματος.
 - Για κάθε οργανισμό να διαχειρίζονται τον κατάλογο των προϊσταμένων του.
 - Να διαχειρίζονται τον κατάλογο χρηστών των πολιτών.
- Οι Προϊστάμενοι/ες Οργανισμών μπορούν:
 - Να διαχειρίζονται τον κατάλογο χρηστών των υπαλλήλων στον οργανισμό τον οποίο εργάζονται.
- Οι Υπάλληλοι Οργανισμών μπορούν:
 - Να διαχειρίζονται τις αιτήσεις ραντεβού του οργανισμού στον οποίο εργάζεται.

ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Εξωτερικό Σύστημα:

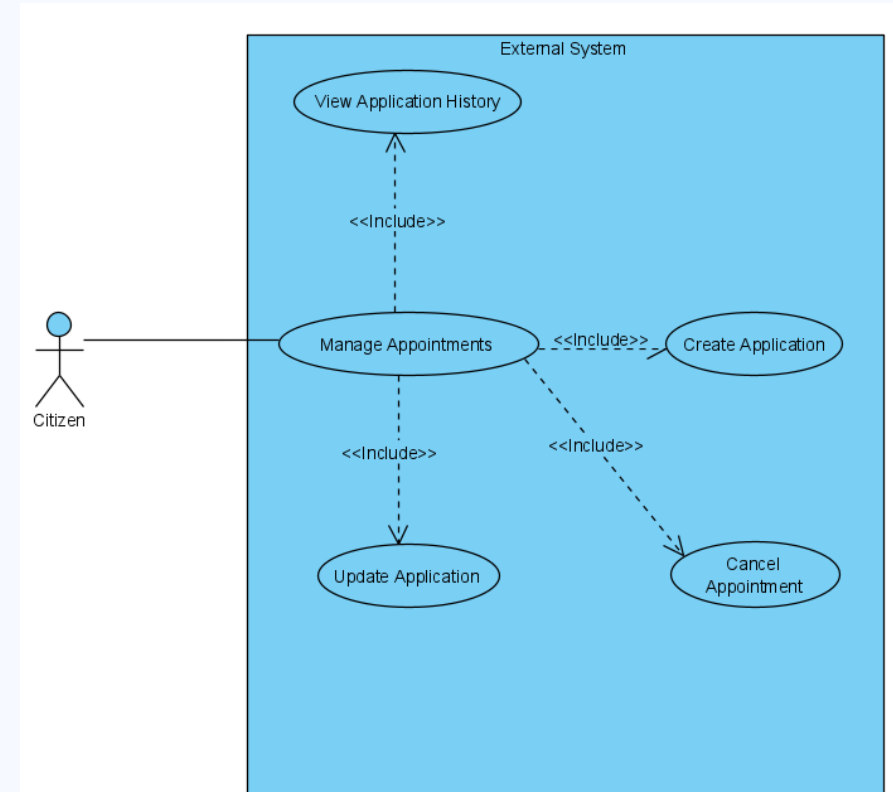
- Οι Πολίτες μπορούν:
 - Να δημιουργούν και γενικώς να διαχειρίζονται τις αιτήσεις που πραγματοποιούν σε οποιοδήποτε από τους οργανισμούς του συστήματος

ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Εσωτερικό Σύστημα

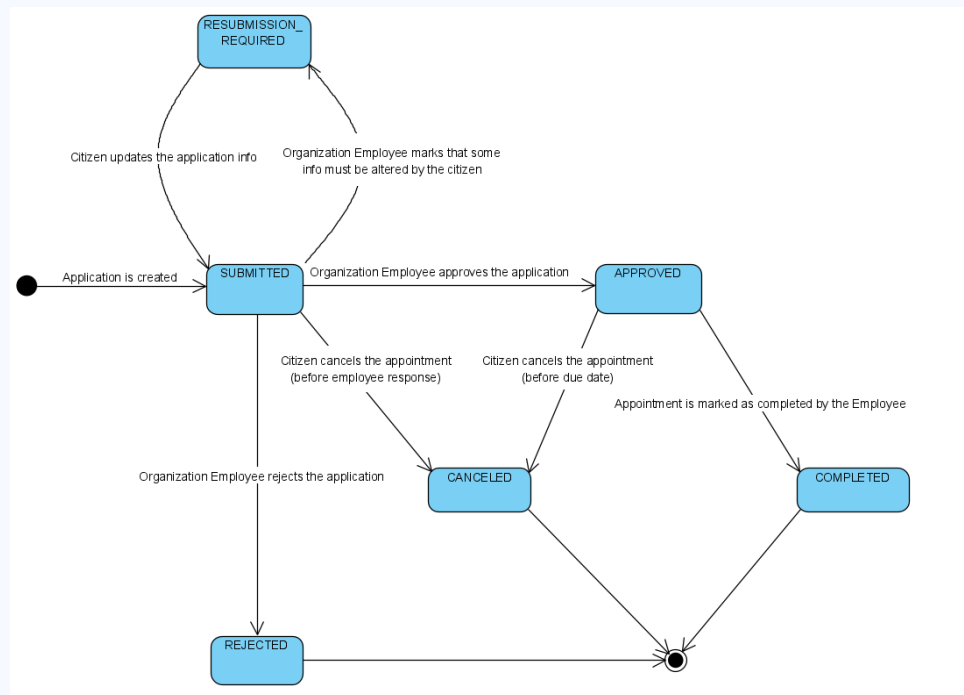


Εξωτερικό Σύστημα



ΛΕΙΤΟΥΡΓΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Πιθανές Καταστάσεις Μιας Αίτησης



Απόκομμα από User Guide εξωτερικού συστήματος

Στο πεδίο Details μπορείτε να αφήσετε οποιοδήποτε σχόλιο επιθυμείτε σχετικά με την αίτηση σας. Επιλέξτε την ημερομηνία και την ώρα που επιθυμείτε να κλείσετε το ραντεβού σας και πατήστε Save(4). Ένας υπάλληλος του οργανισμού στον οποίο καταθέσατε την αίτηση, θα ελέγξει το αίτημά σας και θα ενημερώσει καταλλήλως την κατάσταση της αίτησης σας. Βεβαίως, αν μετανιώσετε για κάτι μπορείτε πάντα να ακυρώσετε την καταχώριση της αίτησης, πατώντας το Cancel(5).

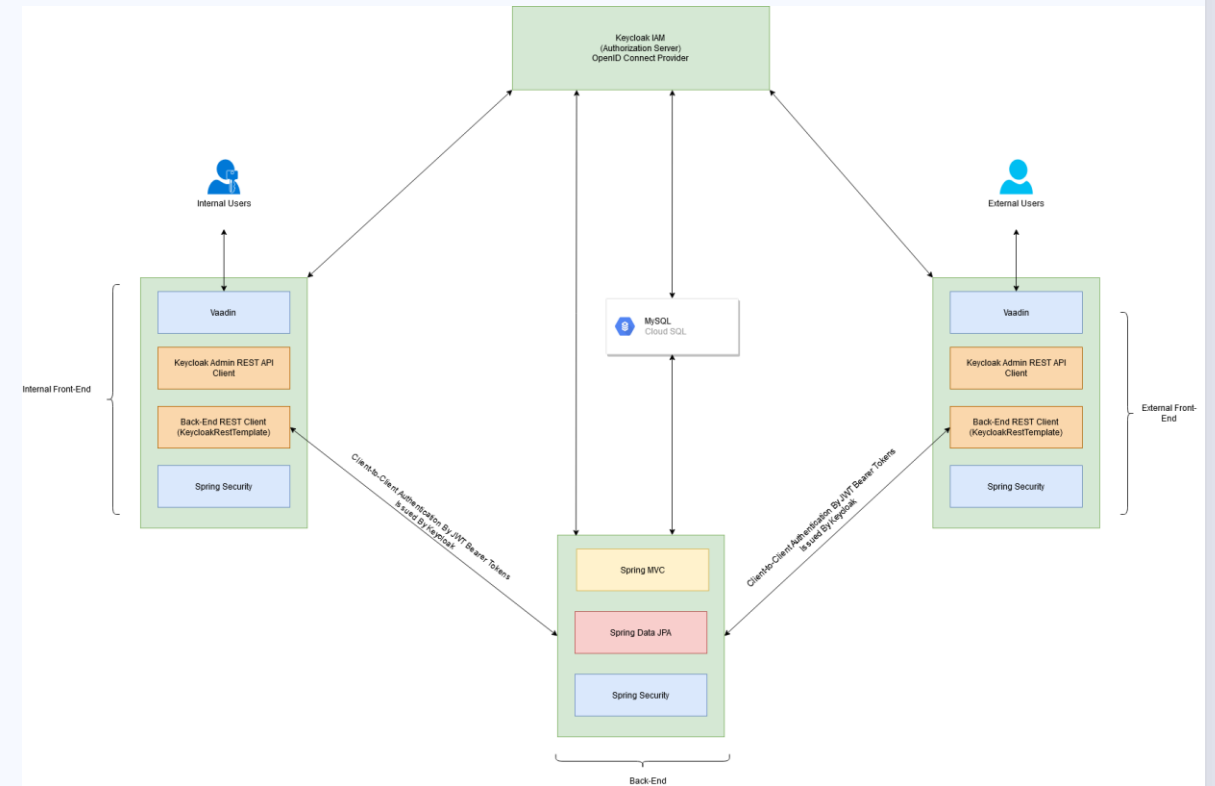
Μπορείτε να **τροποποιήσετε** μία αίτηση πατώντας το Edit(2) **αλλά μόνο** αν ο υπάλληλος του οργανισμού έχει ζητήσει να γίνουν αλλαγές στην αίτηση (Resubmission Required κατάσταση). Η εμπειρία χρήσης είναι ακριβώς η ίδια με την δημιουργία μίας νέας αίτησης.

Μπορείτε οποιοδήποτε στιγμή να **ακυρώσετε** τις αιτήσεις σας πατώντας το Cancel(3), **εκτός** αν το ραντεβού **έχει ήδη πραγματοποιηθεί** (Completed κατάσταση), **έχει απορριφθεί** (Rejected κατάσταση) ή **έχει ήδη ακυρωθεί** (Canceled κατάσταση).

HIGH LEVEL DESIGN

5 Services:

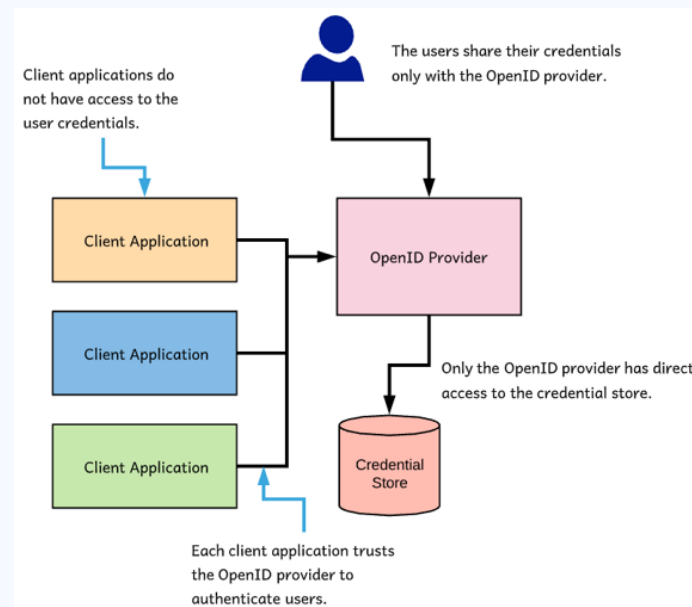
- Authorization Server (Keycloak)
- Applications Service (Back-End)
- Internal Front-End
- External Front-End
- MySQL RDBMS



KEYCLOAK



- An Open-Source Identity And Access Management (IAM) Software Product
- Maintained By Red Hat
- Enables Single Sign-On (SSO)
- Can Be Easily Integrated In Software Projects
- Extremely Customizable
- High Performance



Single-Sign On

Login once to multiple applications



Standard Protocols

OpenID Connect, OAuth 2.0 and SAML 2.0



Centralized Management

For admins and users



Adapters

Secure applications and services easily



LDAP and Active Directory

Connect to existing user directories



Social Login

Easily enable social login



Identity Brokering

OpenID Connect or SAML 2.0 IdPs



Extensible

Customize through code



Password Policies

Customize password policies



Clustering

For scalability and availability



Themes

Customize look and feel



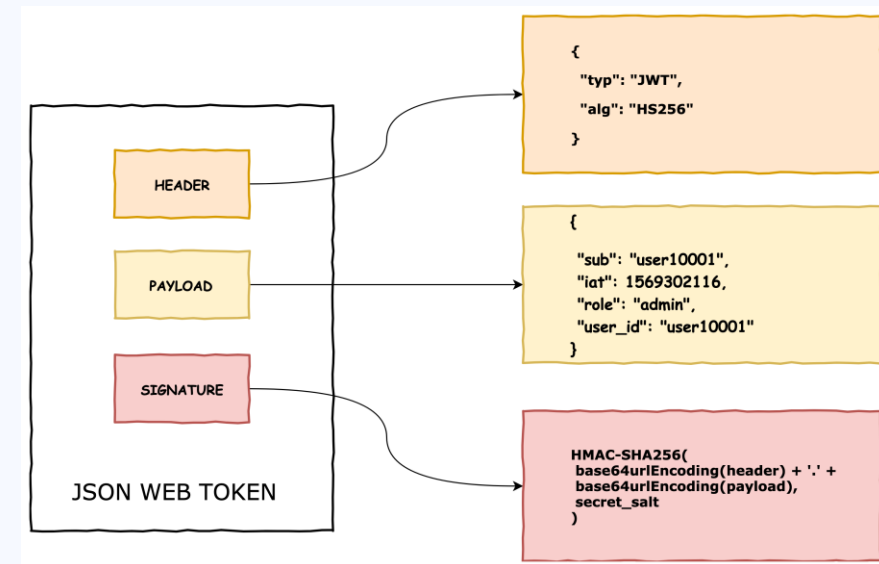
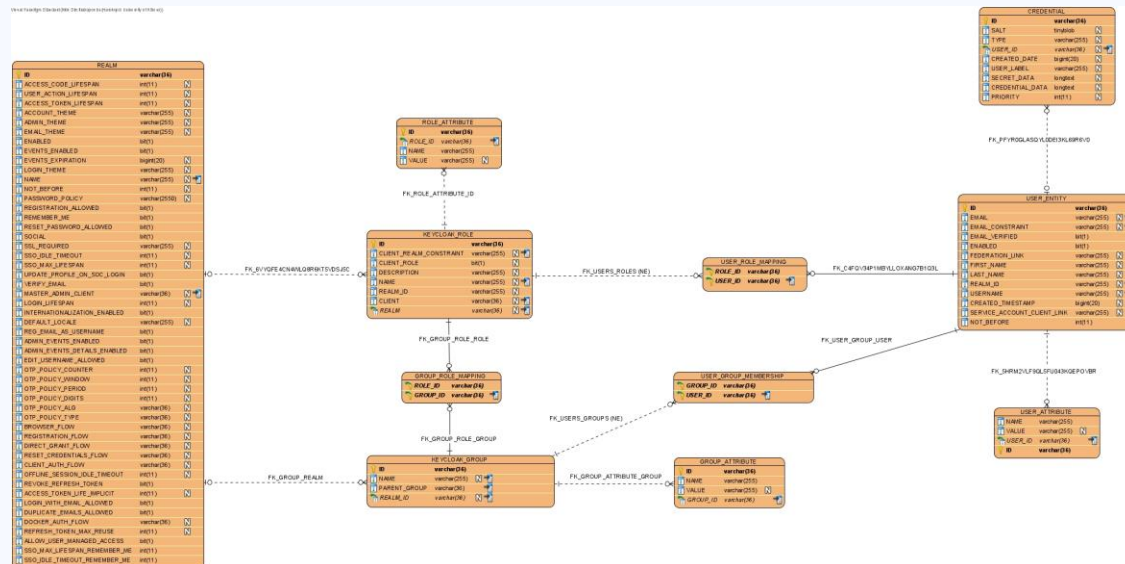
High Performance

Lightweight, fast and scalable



DB & JWT TOKENS WITHOUT ANY IMPLEMENTATION

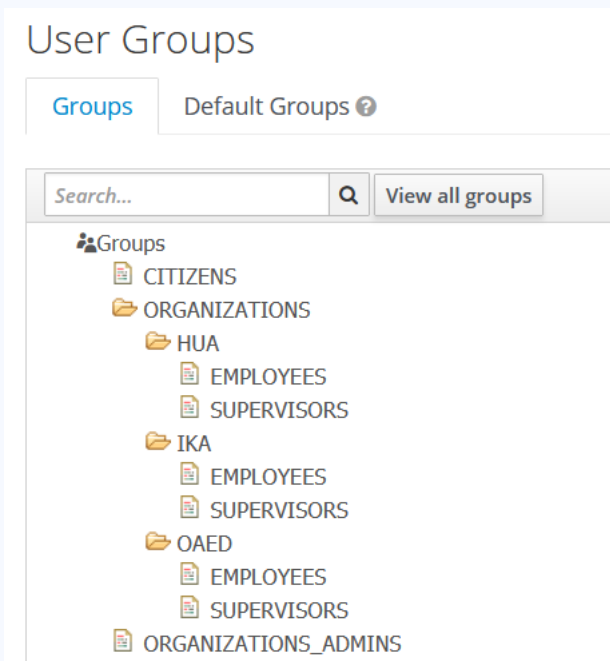
- We don't need to create a DB Schema For Users & Organizations – It's already there
- We don't need to write custom code to generate the JWT Token – Keycloak does it for us





USER CATALOGS (+ ORGANIZATIONS) HASSLE FREE

- Hierarchical User Catalog Management Out-Of-The-Box



KEYCLOAK

CUSTOMIZATION



Custom REST Endpoints (JAX-RS – RESTEasy)

```
public OrganizationsResource(KeycloakSession session) {
    this.session = session;
    this.realm = session.getContext().getRealm();
    this.em = session.getProvider(JpaConnectionProvider.class).getEntityManager();
    this.clientConnection = session.getContext().getConnection();

    AdminAuth adminAuth = authenticateRealmAdminRequest(session.getContext().getRequestHeaders());
    this.auth = AdminPermissions.evaluator(session, realm, adminAuth);
    this.adminEvent = new AdminEventBuilder(realm, adminAuth, session, clientConnection);
    this.organizationsGroup = findOrganizationsGroup();
}

@GET
@Produces({MediaType.APPLICATION_JSON})
public List<GroupRepresentation> list(@QueryParam("offset") Integer offset,
                                     @QueryParam("limit") Integer limit)
{
    auth.groups().requireView();

    //As far as I know, there isn't any ready implementation to get subgroups with paginated queries
    //So I made one myself
    return em.createQuery("SELECT g FROM GroupEntity g WHERE g.parentId = :parentId", GroupEntity.class)
        .setParameter("parentId", organizationsGroup.getId())
        .setFirstResult(offset)
        .setMaxResults(limit)
        .getResultStream()
        .map(entity -> (GroupModel) new GroupAdapter(realm, em, entity))
        .map(model -> ModelToRepresentation.toRepresentation(model, true))
        .collect(Collectors.toList());
}
```

Custom Mappers(Put Info In Token)

```
protected void setClaim(IDToken token, ProtocolMapperModel mappingModel, UserSessionModel userSession) {

    UserModel user = userSession.getUser();
    Set<RoleModel> roles = new HashSet<>();
    roles.add(KeycloakModelUtils.getRoleFromString(userSession.getRealm(), "ROLE_ORG_SUPERVISOR"));
    roles.add(KeycloakModelUtils.getRoleFromString(userSession.getRealm(), "ROLE_ORG_EMPLOYEE"));

    boolean worksInOrganization = user.getGroupsStream()
        .flatMap(RoleMapperModel::getRoleMappingsStream)
        .anyMatch(roles::contains);

    if(!worksInOrganization)
        return;

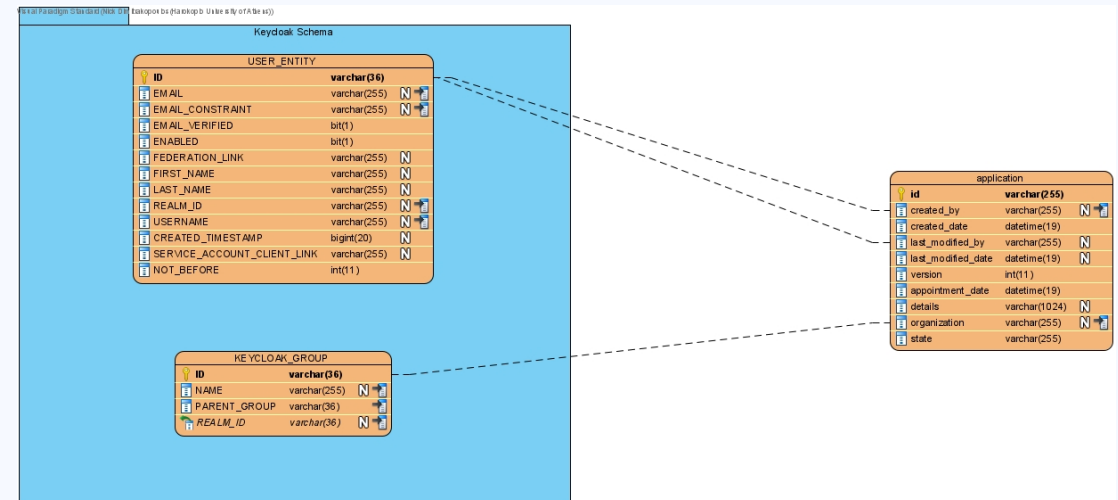
    //We assume that a correct setup is made here.
    //If the role is not assigned to a group or
    //that group does not have a parent, this will fail miserably.
    GroupModel group = userSession.getUser()
        .getGroupsStream()
        .findFirst()
        .get()
        .getParent();

    String protocolClaim = mappingModel.getConfig().get(OIDCAttributeMapperHelper.TOKEN_CLAIM_NAME);
    token.getOtherClaims().put(protocolClaim, new SummarizedGroup(group.getId(), group.getName()));
}
```

APPLICATIONS SERVICE (SPRING BOOT)



- Our simple Back-End
- Handles Applications Business Logic For Employees And Citizens
- REST API With Pagination Support (Integrates Smoothly With Vaadin)
- Simple DB Schema (Just 1 Table – Thank You Keycloak !)
- Not Exposed To The Outside World But That Doesn't Mean That Security Is Unnecessary
- Bearer-Only Client In Keycloak



APPLICATIONS SERVICE (SPRING BOOT)



- User And Organization Data Are Transmitted By JWT Tokens And Are Smoothly Injected Into SQL Queries (Using SpEL)

```
/**
 * @author <a href="mailto:nikosdelta@protonmail.com">Nick Dimitrakopoulos
 */
public interface OrganizationApplicationRepository extends JpaRepository<Application,String>
{
    @Query("SELECT a FROM Application a WHERE a.organization = ?#{@keycloakTokenHelper.organization.id}")
    Page<Application> findAll(Pageable pageable);

    @PostAuthorize("returnObject.orElseThrow(@resourceNotFoundExceptionSupplier).organization == @keycloakTokenHelper.organization.id")
    Optional<Application> findById(String id);

    @Query("SELECT COUNT(ap.id) FROM Application ap WHERE ap.organization = ?#{@keycloakTokenHelper.organization.id}")
    long count();
}
```

An Example



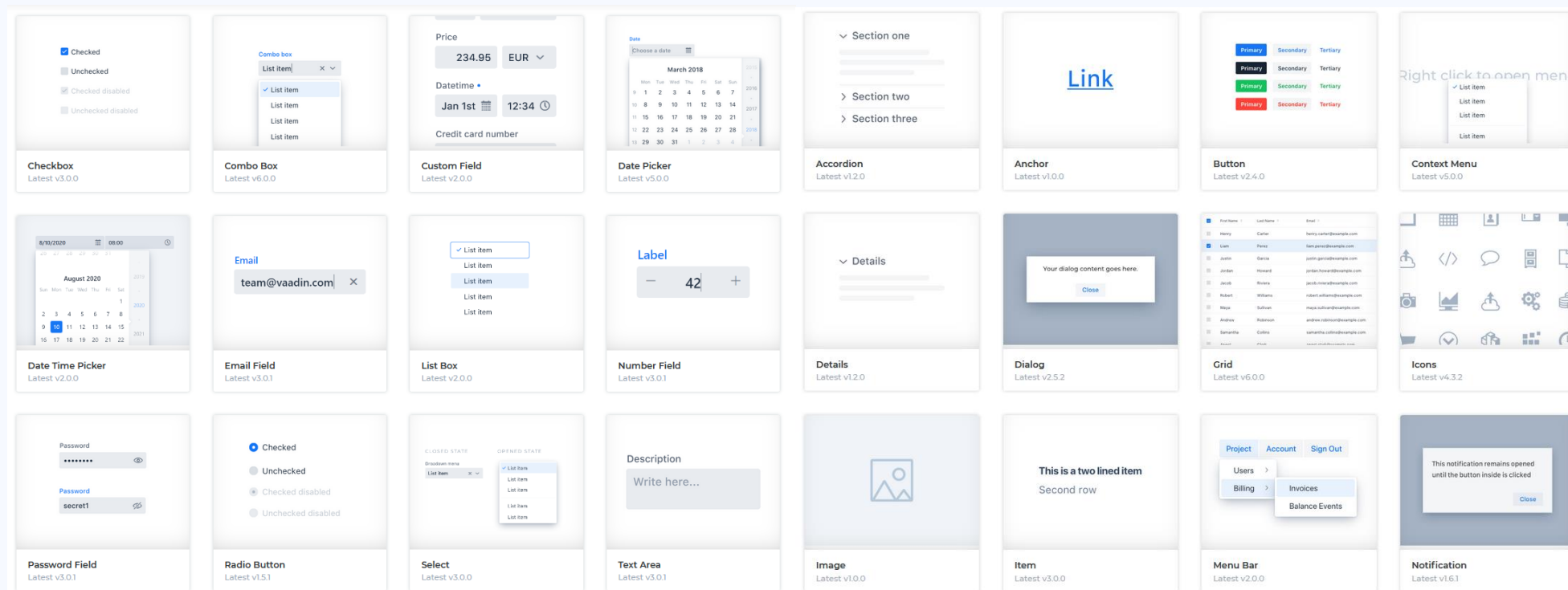
- An Open-Source Platform For Building Web Apps In Java
- Scalable UIs Without Extensive Knowledge of HTML, CSS, Javascript Or Any Other Modern Front-End Library / Framework (React,Vue.js,Angular)
- With This Framework We Handicaped Our Lack In Front-End Development Experience
- Very Powerful If Combined With Spring-Boot
- Great Community (They Even Have A Discord Server Where They Help Devs In Their Vaadin Projects)

A screenshot of a web application named "Hermes". The left sidebar has a menu with "Home", "Organizations", and "Citizens" (selected). The main area displays a table titled "Citizens" with columns: Email, Date of Birth, Address, Postal Code, Locality, Region, Country, Created On, and Account Status. The table contains 20 rows of citizen data. The "Account Status" column has green "Enabled" buttons and red "Disabled" buttons. A "New Citizen" button is at the bottom right.

	Email	Date of Birth	Address	Postal Code	Locality	Region	Country	Created On	Account Status
185-9476	bennett.lemke@email.com	6/6/02	966 Murphy Shore	03333	Lake Dariusfurt	North Carolina	United States	20/2/21, 7:24 μ.μ.	Enabled
00-6244 x56613	bud.jacobs@email.com	1/3/09	475 Ernser Port	60211	Josiahstown	Nevada	United States	20/2/21, 7:24 μ.μ.	Enabled
5-3383	candelaria.muller@email.com	12/8/11	8839 Doreatha Flats	93566-6812	Eugenafort	Kansas	United States	20/2/21, 7:24 μ.μ.	Enabled
3.2465 x65112	cordell.stanton@email.com	14/10/86	98609 Nicky Spring	48984-5740	West Garrett	Alaska	United States	20/2/21, 7:24 μ.μ.	Enabled
38-5941 x32389	curt.toy@email.com	4/9/15	44404 Zemlak Streets	18291	Cormierbury	Minnesota	United States	20/2/21, 7:24 μ.μ.	Enabled
5-6461 x112	damien.emmerich@email.com	16/11/79	953 Ebert Mountains	21870-2440	Connellystad	California	United States	20/2/21, 7:24 μ.μ.	Disabled
31-6787 x0372	danny.kunze@email.com	13/12/20	81401 Conn Mill	53230-4394	East Kytelown	Idaho	United States	20/2/21, 7:24 μ.μ.	Disabled
25-4474 x46854	elisha.prohaska@email.com	8/10/87	72908 Carleen View	80918	Lake Faustino	Rhode Island	United States	20/2/21, 7:24 μ.μ.	Enabled
315-6423	galina.jacobs@email.com	27/11/11	937 Conroy Mill	36977-5895	Port Wilmer	Virginia	United States	20/2/21, 7:24 μ.μ.	Disabled
5-3097 x30619	haley.yost@email.com	20/2/71	2424 Larson Canyon	73890-1634	Wisokychester	Maine	United States	20/2/21, 7:24 μ.μ.	Disabled
1.3186 x21117	harland.cartwright@email.com	29/10/09	867 Beatty Stream	90130	West Lincoln	Georgia	United States	20/2/21, 7:24 μ.μ.	Enabled
308-2695 x67822	jarrett.mueller@email.com	11/12/73	7455 Carmen Groves	54312	Lake Tuanfort	Idaho	United States	20/2/21, 7:24 μ.μ.	Enabled
5-5089 x14021	kecia.howell@email.com	28/9/16	4248 Rosalind Alley	72174	South Eddy	Michigan	United States	20/2/21, 7:24 μ.μ.	Disabled
0-0325	lee.kuphal@email.com	26/8/19	423 Collier Square	60186-8826	New Briceview	Maryland	United States	20/2/21, 7:24 μ.μ.	Enabled
3-5488 x92387	leonel.bahringer@email.com	7/9/11	778 Jones Run	70556	North Octavio	Wyoming	United States	20/2/21, 7:24 μ.μ.	Disabled
326-4595 x35236	lucy.dibbert@email.com	18/10/70	75798 Evan Coves	19351	Corkeryhaven	Washington	United States	20/2/21, 7:24 μ.μ.	Enabled
0-9140 x9254	mario.harris@email.com	5/8/00	573 Hyacinth Parks	60732	North Zachary	Louisiana	United States	20/2/21, 7:24 μ.μ.	Enabled
8.2959	marty.deckow@email.com	3/9/85	277 Jaye Fords	42490	New Colin	Florida	United States	20/2/21, 7:24 μ.μ.	Disabled



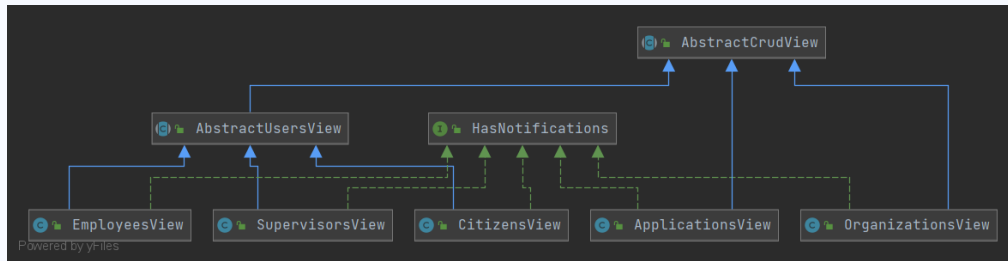
- Ready UI Components – 100% Java Compatible





SOME OF OUR PROJECT'S DESIGN GUIDELINES

Abstract Views



90%-95% of The Code Is
In The Abstract Classes !

Paginated Grid

```
@Override
public void setupDataProvider()
{
    this.getGrid().setDataProvider(DataProvider.fromCallbacks(
        fetch -> presenter.findAll(fetch.getOffset(), fetch.getLimit()).stream(),
        count -> presenter.count()));
}
```

Every Grid Fetches Data Lazily

TEST DRIVE



Ώρα για δοκιμή !

Όλοι **οι φοιτητές (πολίτες)** μπορούν να συνδεθούν στο **external.hermesapp.xyz** και να κάνουν αιτήσεις στο πανεπιστήμιό μας.

Είναι καταχωρημένοι όλοι οι it217XXX και it218XXX χρήστες της σχολής μας. Αν είστε από παλιότερο έτος απλά επιλέξτε τυχαία έναν μητρώο εντός αυτού του εύρους.

Το Εσωτερικό Σύστημα βρίσκεται στην διεύθυνση **internal.hermesapp.xyz**

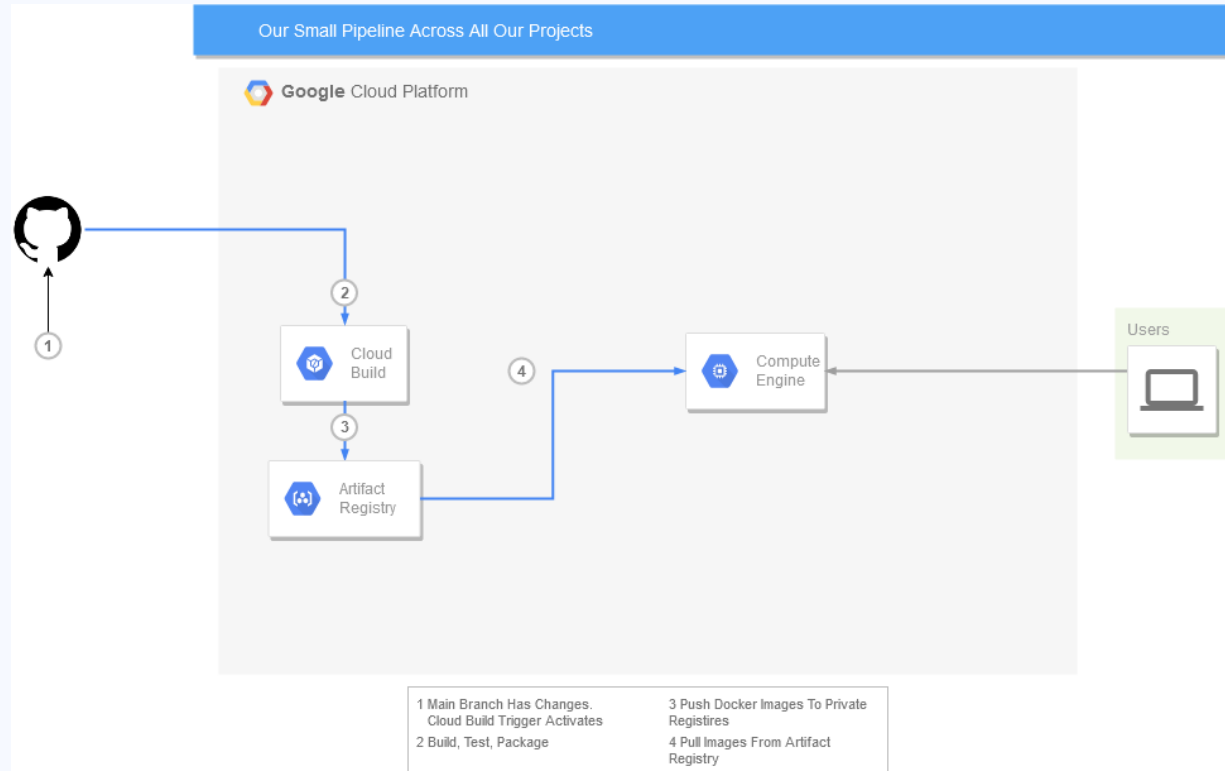
Ο κ. Τσαδήμας είναι **διαχειριστής οργανισμών**. Αν θελήσει, μπορεί να δημιουργήσει έναν νέο οργανισμό και στην συνέχεια να διαχειριστεί τους/τις προϊσταμένους/ες του.

Η κ.Μάρα είναι **υπάλληλος του ΗΥΑ**. Μπορεί να δει τις αιτήσεις που καταθέτουμε εμείς οι φοιτητές και να ενημερώνει την κατάστασή τους.

Αν θελήσετε να δείτε το interface ενός **προϊσταμένου**, μπορείτε να μπείτε με το account του κ.Μιχαήλ (υπάρχουν και άλλοι).

Κωδικός Πρόσβασης σε κάθε λογαριασμό είναι το όνομα του account (πχ για το it21821 κωδικός είναι το it21821).

A BIT DEVOPS

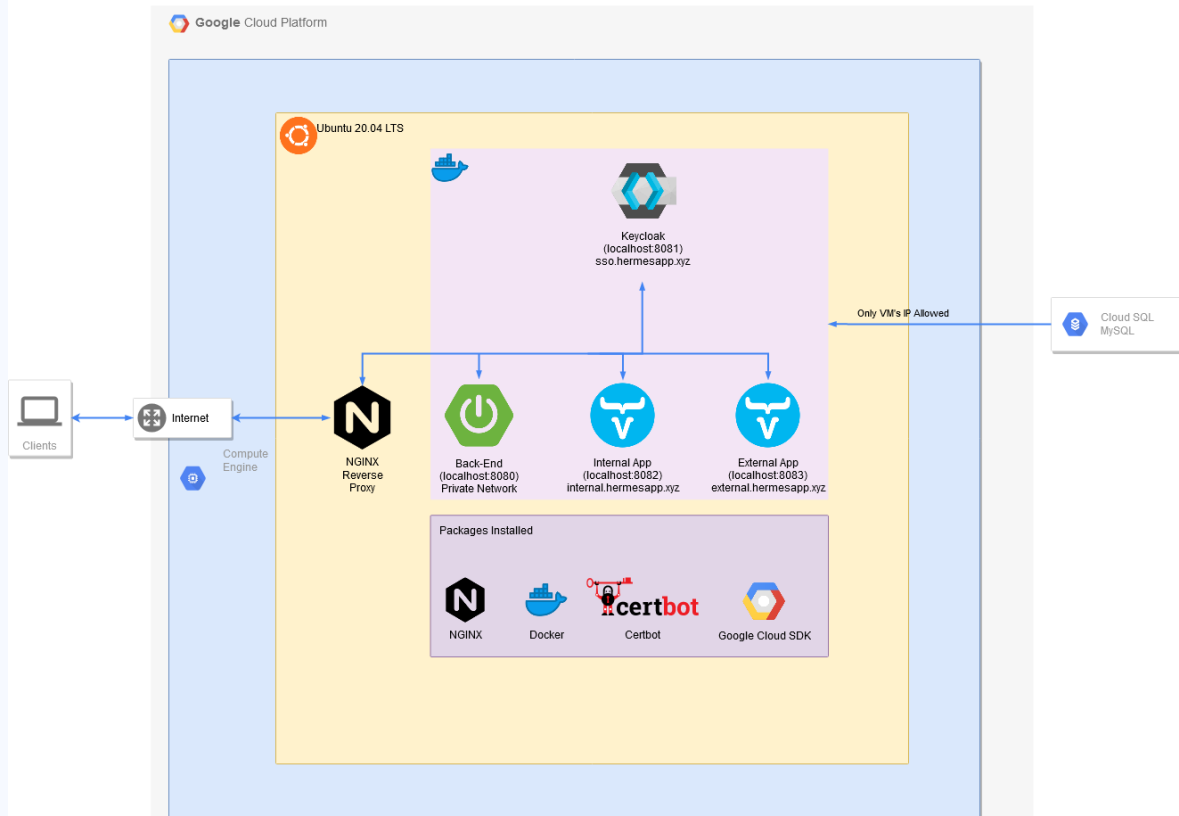


Some Benefits:

- Easy Local Development Setups With docker-compose
- Easier Deployment On Production (Continuous Deployment - CD)
- Ensure That Changes Don't Break The System By Continuous Integration (CI) – Not Present Here

DEPLOYMENT: GCP

Our Deployment



Basic Idea:

- Compute Engine VM With Following Software Installed:
 - Docker & Docker-Compose (Containers)
 - NGINX (Reverse Proxy – Our Ticket To Easy SSL)
 - Certbot (Certificates)
 - Google Cloud SDK (Preinstalled – Easy Access To Google Cloud's Products)
- Docker-Compose Setup For Our 4 Images
- Cloud SQL MySQL 5.7 Instance As DB
- Purchased Domain hermesapp.xyz And Used Cloud DNS

DEPLOYMENT: GCP

Cloud DNS Config

hermes-dns

DNS name hermesapp.xyz.






Type Public

RECORD SETS

[ADD RECORD SET](#)

[DELETE RECORD SETS](#)

 **Filter** Filter record sets

<input type="checkbox"/>	DNS name	Type ↑	TTL (seconds)	Data	
<input type="checkbox"/>	external.hermesapp.xyz.	A	300	• 35.246.117.37	
<input type="checkbox"/>	internal.hermesapp.xyz.	A	300	• 35.246.117.37	
<input type="checkbox"/>	sso.hermesapp.xyz.	A	300	• 35.246.117.37	
	hermesapp.xyz.	NS	21600	• ns-cloud-a1.googledomains.com. • ns-cloud-a2.googledomains.com. • ns-cloud-a3.googledomains.com. • ns-cloud-a4.googledomains.com.	
	hermesapp.xyz.	SOA	21600	• ns-cloud-a1.googledomains.com. cloud-dns-hostmaster.google.com. 1 21600 3600 259200 300	

Equivalent [REST](#)