

ΔΙΠΛΩΜΑΤΙΚΉ ΕΡΓΑΣΙΑ

ANDROID & JAVA ΕΕ

APPLICATION



ΑΝΤΩΝΟΠΟΥΛΟΣ ΧΡΗΣΤΟΣ
2016

The eyes of the angels



Περιεχόμενα

ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

DATABASE SCHEMA

ΟΘΟΝΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

ΕΠΙΛΟΓΕΣ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

ΤΕΧΝΙΚΗ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ

JPA

EJB

Java Servlets

JAX-RS

JAAS

FRONT-END

ajax-jquery

[Google Maps](#)

[Activitys](#)

[Services](#)

[Broadcast receiver](#)

[Intent - Filters](#)

[Fragments](#)

[Location](#)

[UI-LAYOUT](#)

[Connecting to the Network](#)

[Android wearble](#)

[ΣΥΜΠΕΡΑΣΜΑΤΑ](#)

Στόχος της εφαρμογής

Στα πλαίσια της πτυχιακής εργασίας υλοποίησα ένα web application καθώς και μία android εφαρμογή όπου υπάρχει και η επέκτασή της σε android wearable.

Η mobile εφαρμογή απευθύνεται σε ευρύ κοινό όπως παιδιά ενήλικες και ηλικιωμένους όπως και άλλες ευαίσθητες κοινωνικές ομάδες. Καταχωρώντας οι χρήστες τα στοιχεία τους στο κεντρικό σύστημα και ενημερώνοντας μέσα από το gps του κινητού την τοποθεσία σε πραγματικό χρόνο με αποτέλεσμα όταν καλέστε βοήθεια μέσα από την εφαρμογή να γνωρίζει ο κεντρικός διαχειριστής ποια είναι η τοποθεσία τους.

Με το application μπορεί ο χρήστης να καλεί σε βοήθεια με διάφορους τρόπους εκτός από τους συνηθισμένους απάντηση απλά μόνο ένα κουμπί.

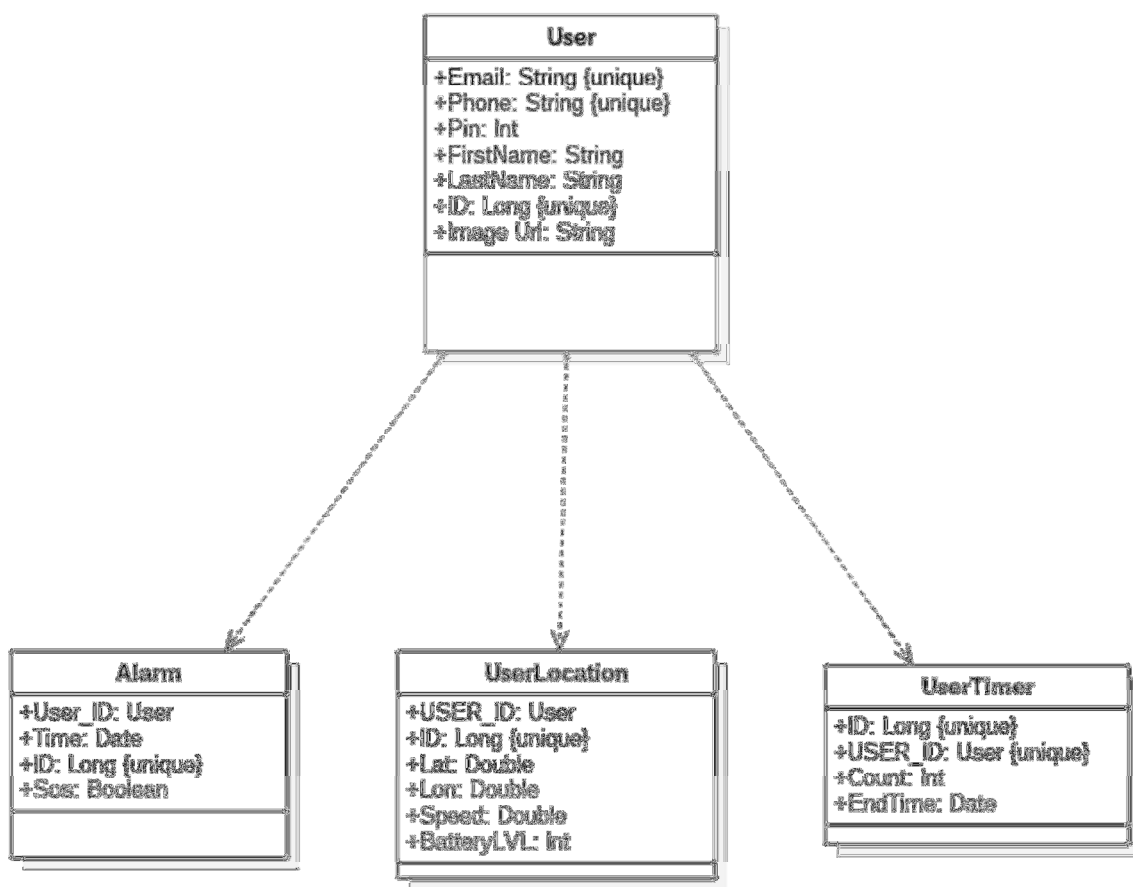
Δίνεται η δυνατότητα να καλέσει βοήθεια μέσα από το smartphone μπορούν όμως να υπάρχουν και άλλα σενάρια με το ποιους τρόπους θα καλέσει βοήθεια κάνοντας χρήση του κουμπιού από το hands-free που είναι συνδεδεμένες στο κινητό όταν ο χρήστης κάνει αυτή τη διαδικασία η εφαρμογή θα αποστείλει σήμα ότι βρίσκεται σε κίνδυνο και το κινητό ακόμα και αν είναι σε κατάσταση αναμονής θα κάνει τα παραπάνω και θα καλέσει σε έναν αριθμό που του έχουμε ορίσει ώστε να υπάρχει δυνατότητα να επικοινωνήσουμε με το χρήστη σε ένα κρίσιμο γεγονός ώστε να δούμε άμεσα τι βοήθεια μπορούμε να παρέχουμε. Η δεύτερη δυνατότητα που παρέχεται είναι μέσα από τις ρυθμίσεις της εφαρμογής ο χρήστης επιλέγοντας ένα κουμπί να ξέρει ότι όταν το έχει ενεργοποιήσει και αφαιρέσει τα ακουστικά από το κινητό τότε το κινητό θα εκτελέσει πάλι την παραπάνω διαδικασία. Μπορεί να δημιουργήσει ένα χρονόμετρο το οποίο όταν τελειώσει ο χρόνος του και δεν το έχει απενεργοποιήσει ο διαχειριστής θεωρεί ότι βρίσκεται σε κίνδυνο.

Από την άλλη πλευρά η εφαρμογή που βρίσκεται στο cloud τρέχει στον browser και παρέχει στον διαχειριστή 1 google maps τη λίστα από τους χρήστες που είναι εγγεγραμμένοι στην εφαρμογή και του δίνει τη δυνατότητα να μπορεί να δει το πού βρίσκεται ο κάθε χρήστης και το ποιος χρήστης βρίσκεται σε κατάσταση που ζητάει βοήθεια. Με στόχο να

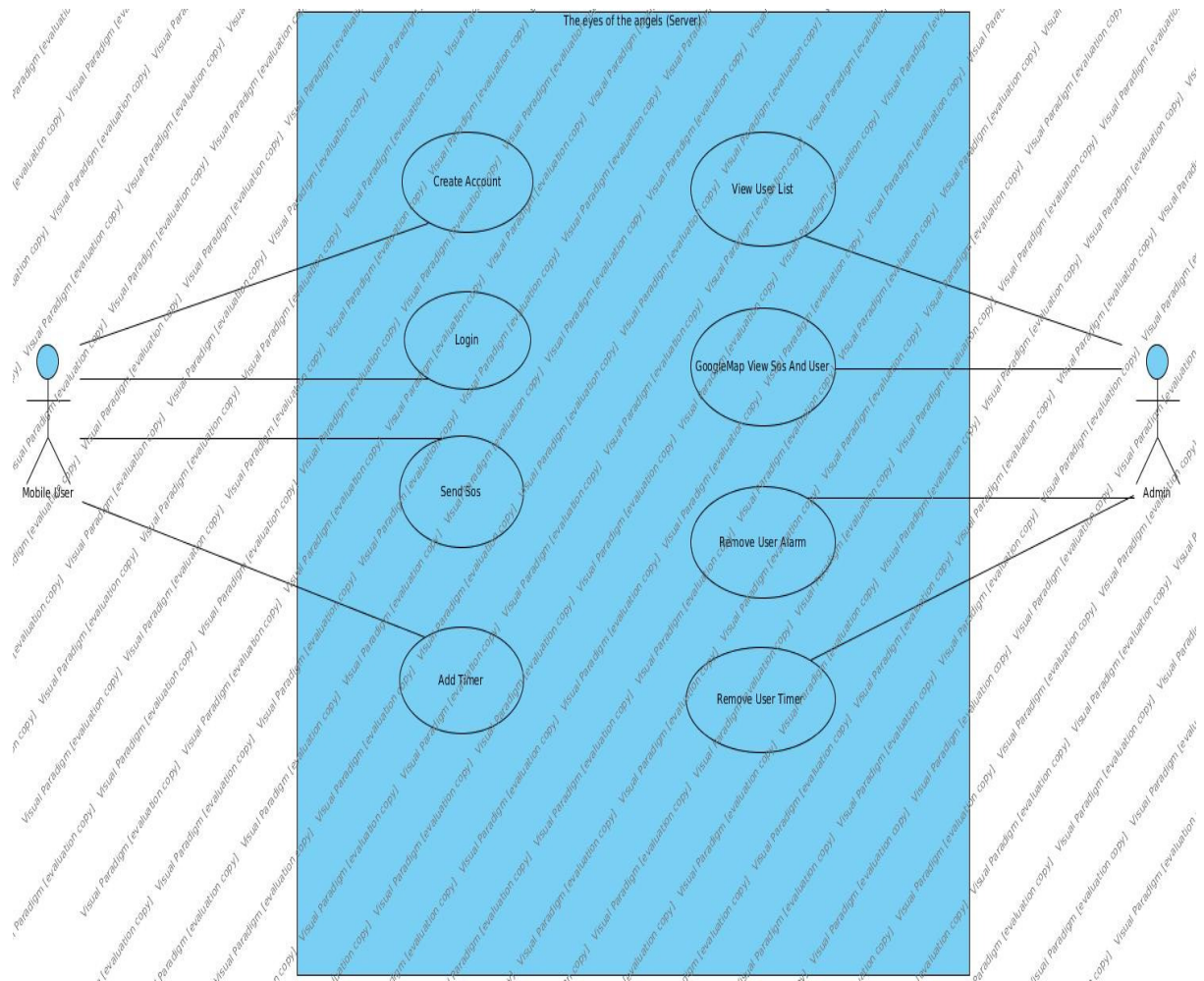
μπορέσει να βοηθήσει τους χρήστες είτε καλώντας κάποιο ασθενοφόρο η ειδοποιώντας κάποιο συγγενή ακόμα και καλώντας την αστυνομία.

ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

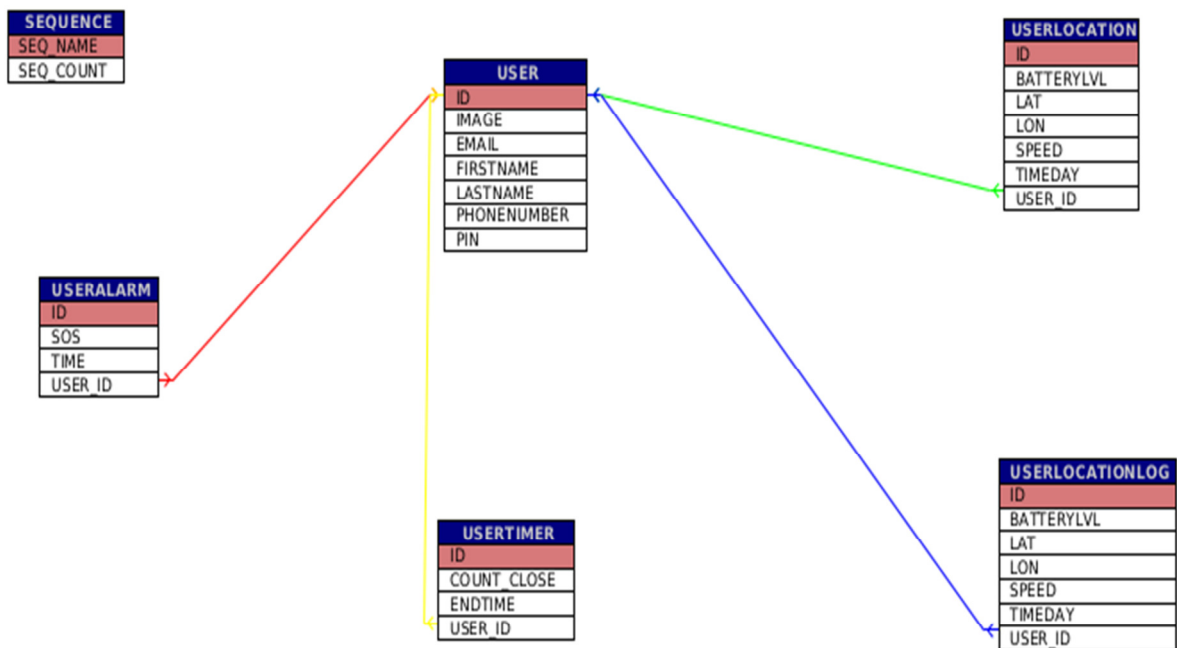
UML CLASS DIAGRAM



UML USE CASES



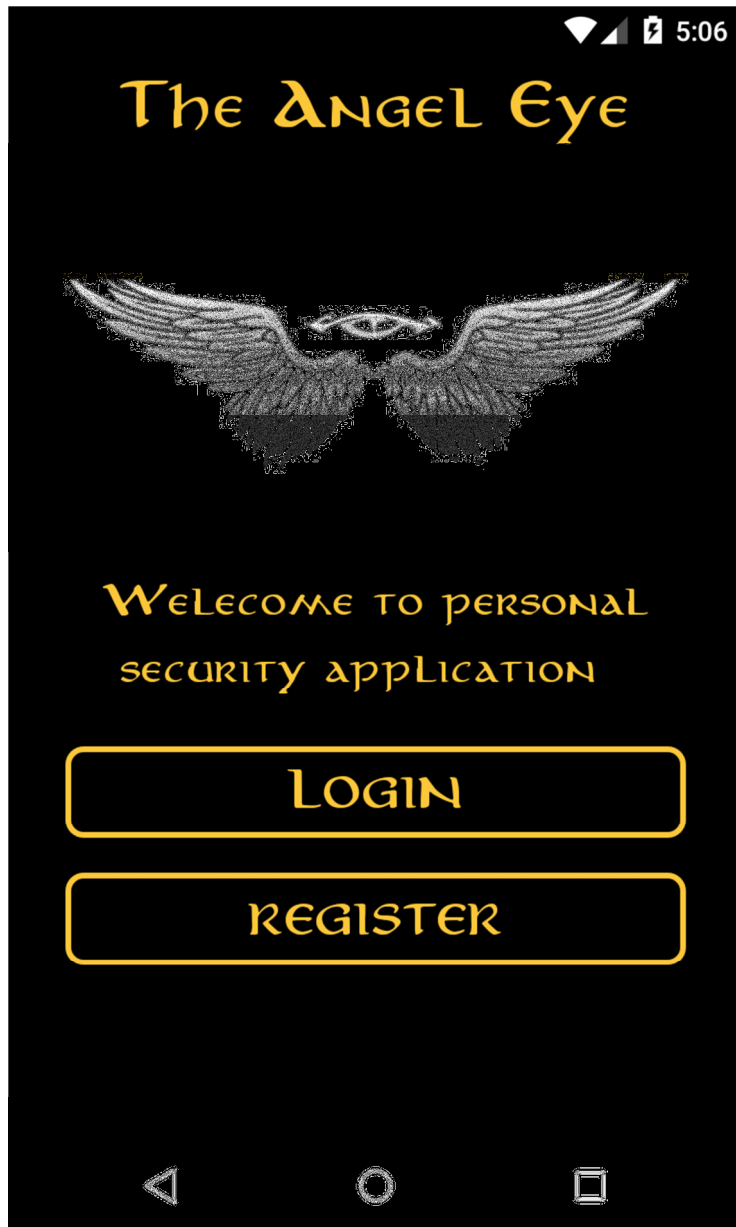
DATABASE SCHEMA



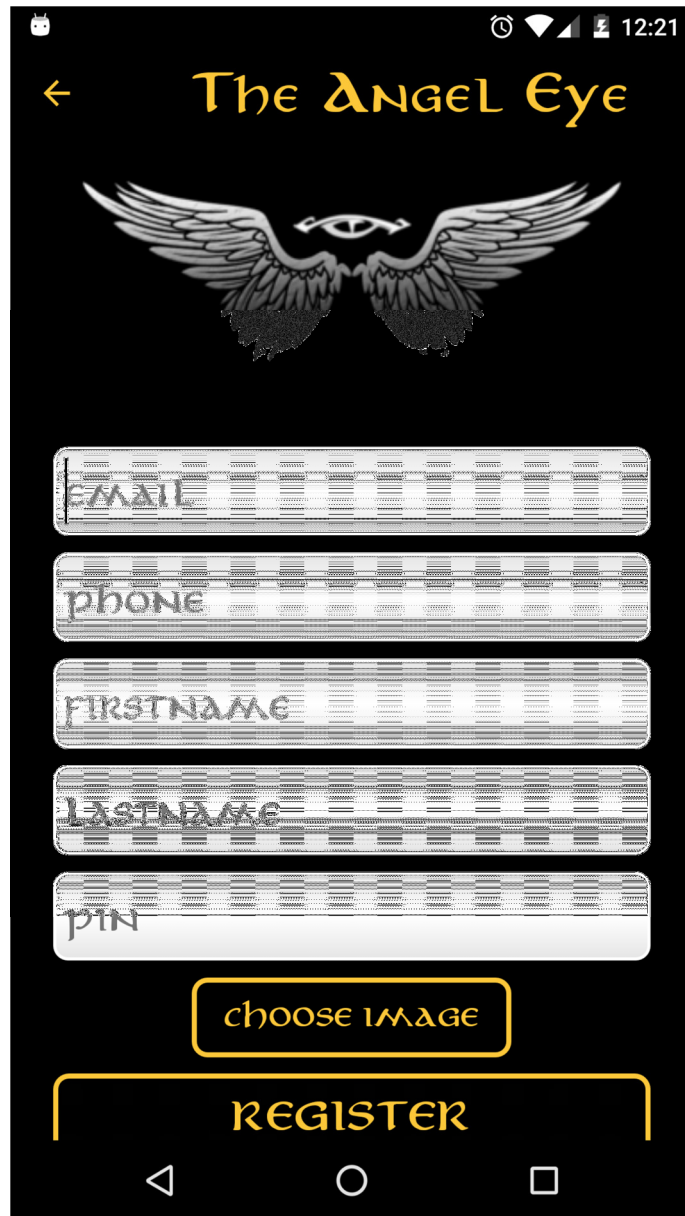
Η δημιουργία της βάσεις έγινε με την χρήση του JPA.

ΟΘΩΝΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Start Activity

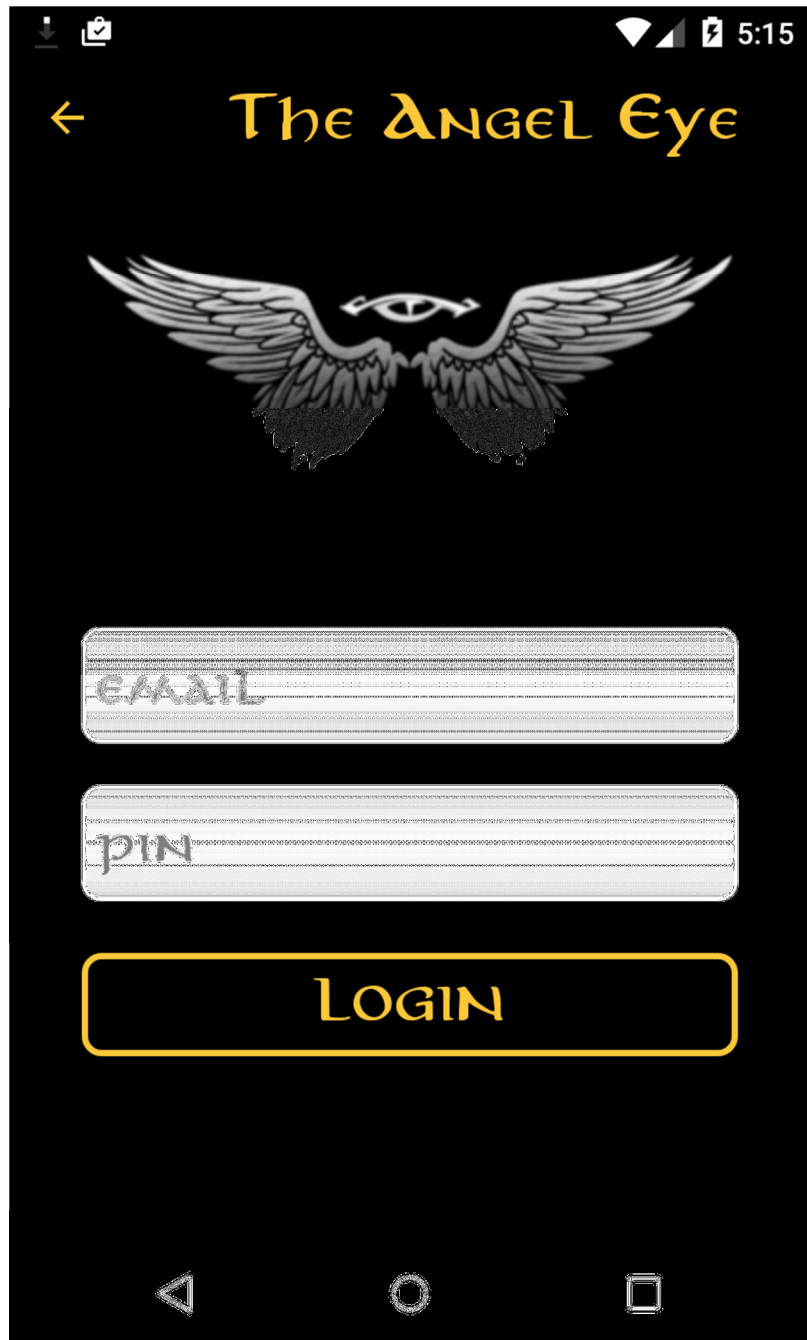


Register Activity



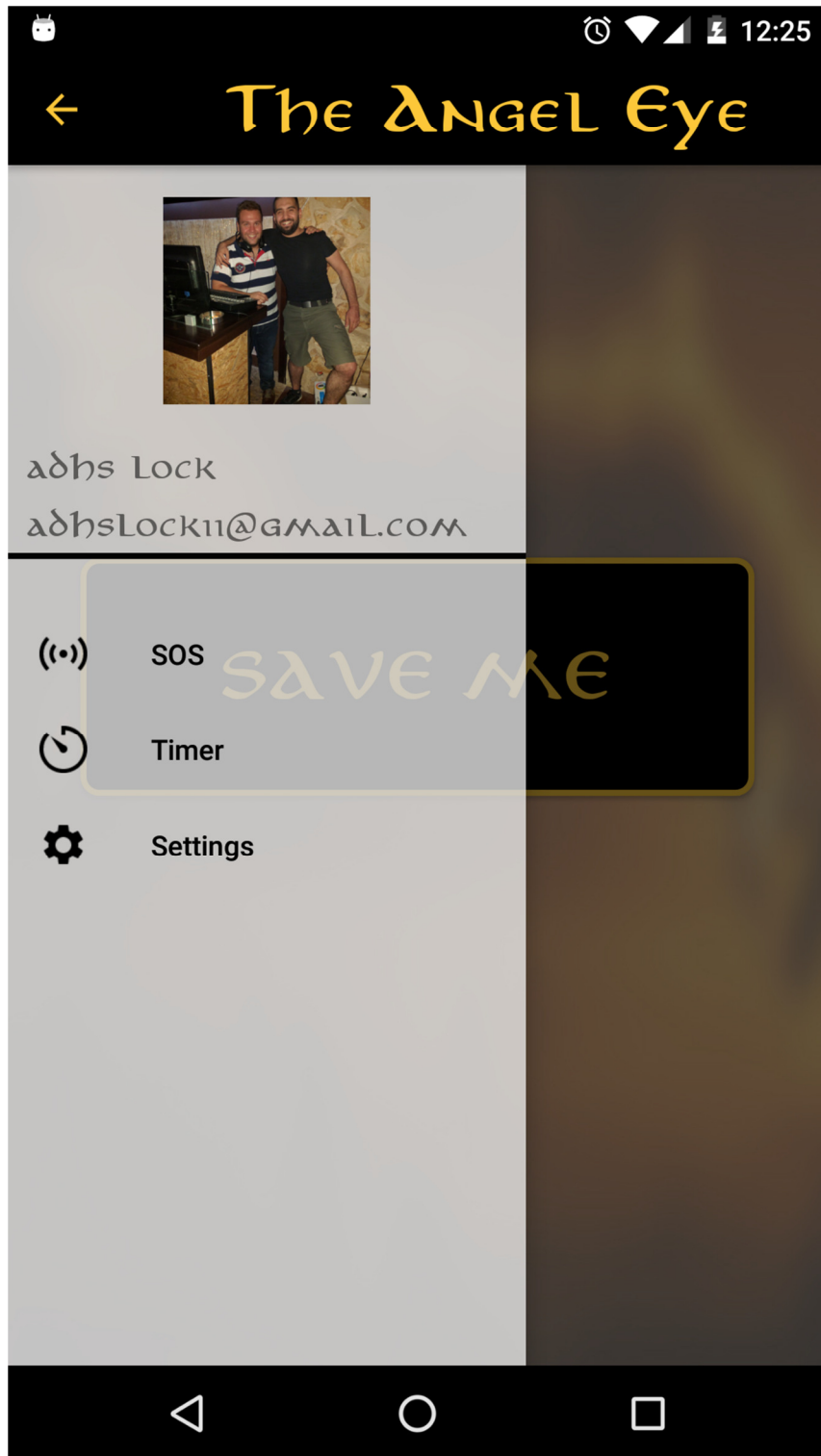
1. Ο χρήστης καταχωρεί τα στοιχεία του email ονομα επωνυμο αριθμο κινητού pin και φωτογραφια .
2. Αυτά αποθηκεύονται στο server και σε μια database.

Login Screen



1. Ο χρήστης βάζει το email και το pin του και μπαίνει στην εφαρμογή.

Main Activity

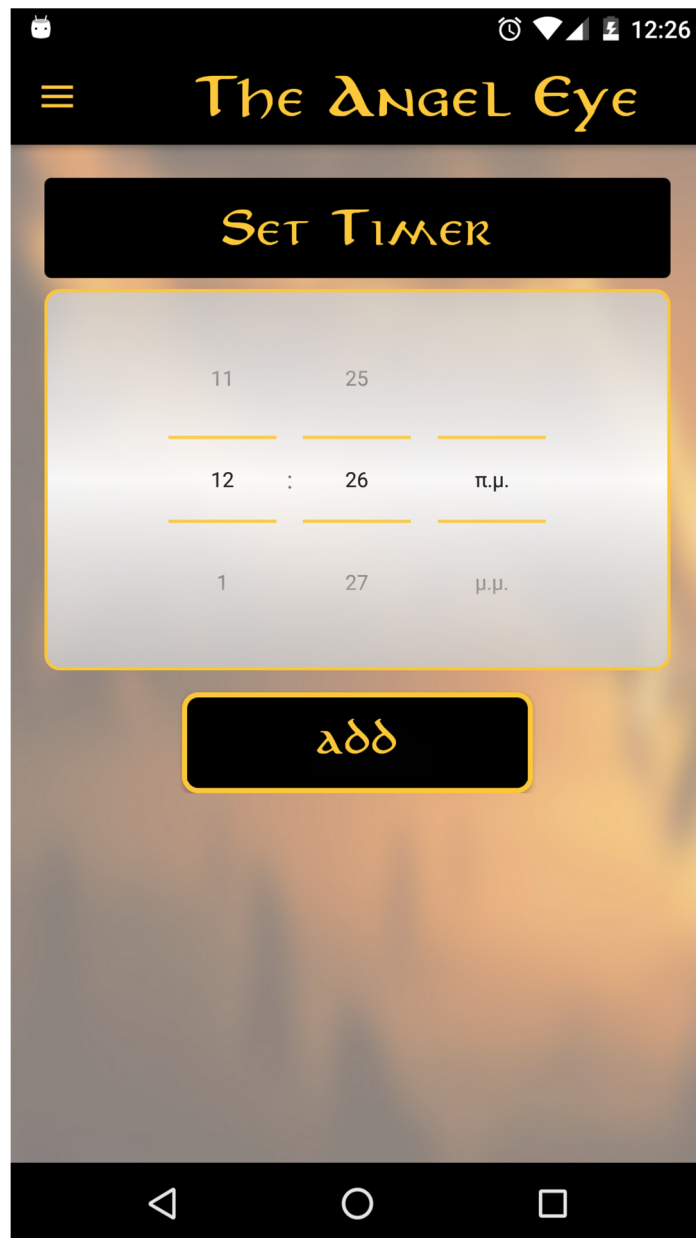


Save me screen

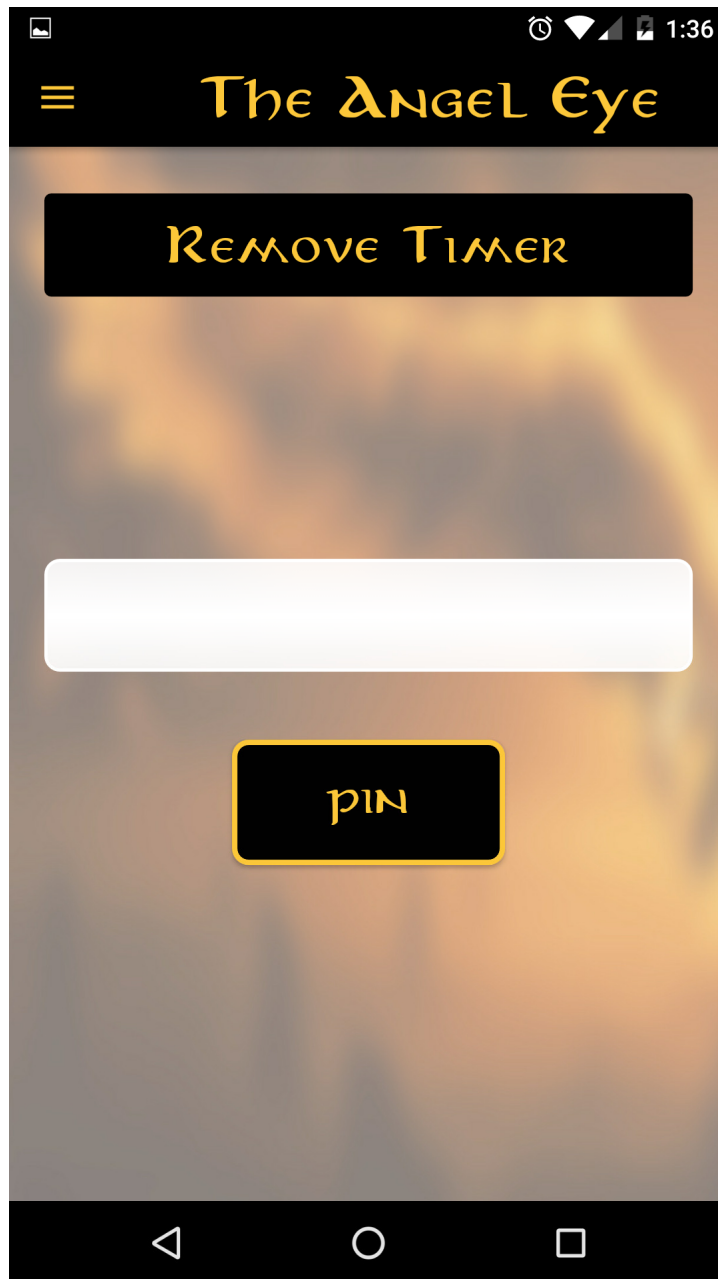


1. Με το πάτημα το κουμπί το κινητό θα στείλει στον server ένα σήμα βοήθειας και θα καλέσει αυτόματα σε έναν αριθμό.

Timer screen

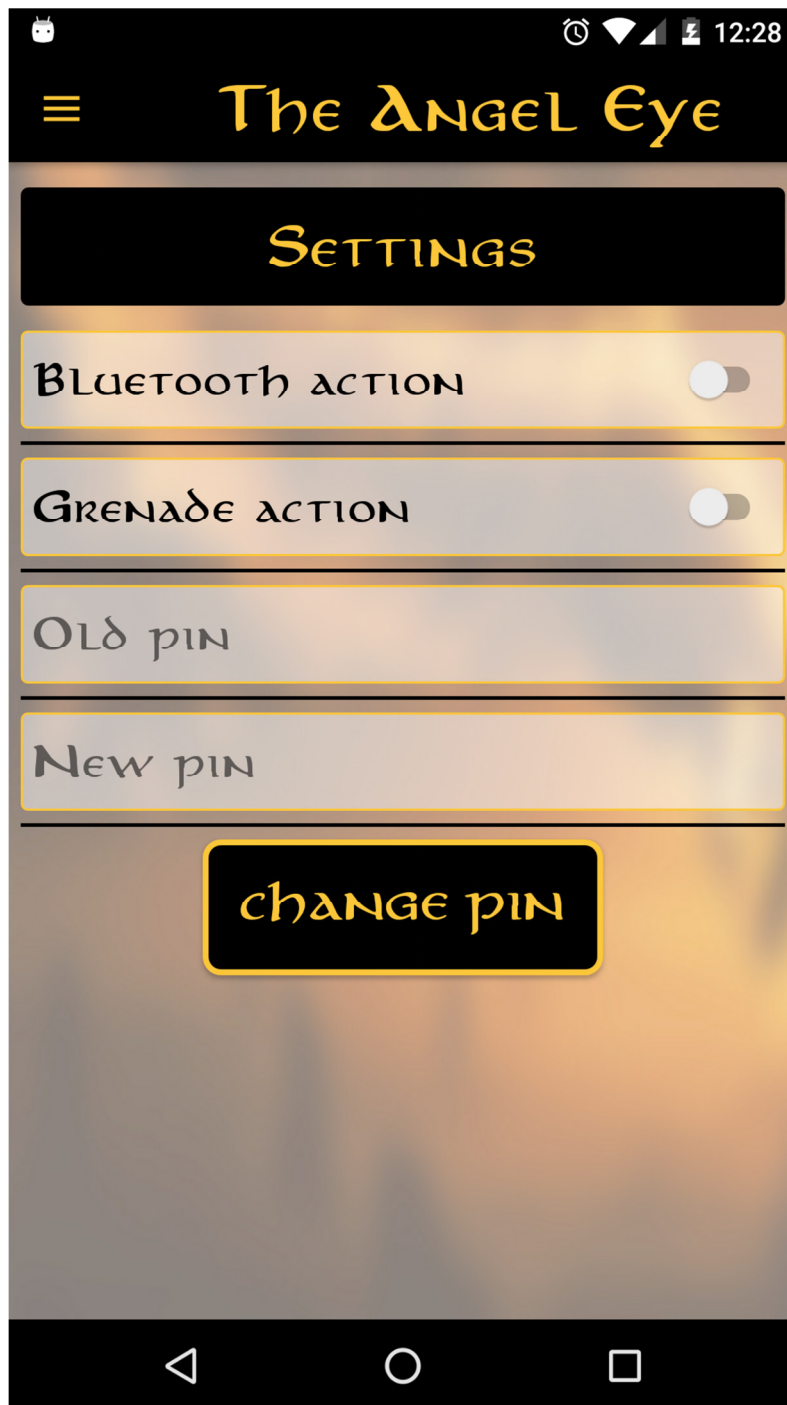


1. Ορίζουμε ένα χρονόμετρο το όπιο όταν περάσει ο χρόνος που έχουμε δηλώσει ο σύστημα θεωρεί ότι ο χρήστης βρίσκεται σε κίνδυνο.



2. Μετά εμφανίζεται ένα πεδίο όπου όταν βάλουμε το pin απενεργοποιούμε το χρονόμετρο ,αν γίνει λάθος 3 φορές το pin δεν μπορούμε να των κλείσουμε.

Settings screen



1. Bluetooth Action σε ορισμένες συσκευές όταν έχουμε το επιλέξει πατώντας το τρίτο κουμπί από το hands free στέλνει σήμα στον server και κάνει κλήση σε έναν αριθμό ακόμα αν και το κινητό να είναι κλειδωμένο.

2. Gernade Action κάνοντας την συγκεκριμένη επιλογή όταν βγάλουμε το hands free απο το κινητό θα γίνει ακριβός η ίδια διαδικασία με το bluetooth action.
3. Τελευταία δυνατότητα είναι να μπορεί ο χρήστης να αλλάξει το pin του.

ΕΠΙΛΟΓΕΣ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

Το σύστημα αποτελείται από διάφορες τεχνολογίες.

Για την υλοποίηση της βάσεις δεδομένων επέλεξα mysql που είναι δωρεάν και είναι της εταιρείας oracle. Οι λόγοι που με οδήγησαν σε αυτή την επιλογή είναι οι εξής .Ο βασικότερος λόγος είναι πάρα πολύ διαδεδομένη στην αγορά εργασίας. Έχει πάρα πολλά κοινά χαρακτηριστικά με άλλες γλώσσες το ίδιο σκοπού. Δεύτερος λόγος είναι το διανέμει δωρεάν καθώς την έχουμε διδαχθεί στη σχολή. Τέλος παρέχει την βιβλιοθήκη JDBC όπου είναι ένας οδηγός που δίνει την δυνατότητα στην java να επικοινωνεί και να διαχειρίζεται καθώς και να σχεδιάζει μία βάση δεδομένων.

Web/Application Server η επιλογή αυτή κινήθηκε με βάση πολλούς παράγοντες. Η επιλογή αυτή είναι πάρα πολύ σημαντική διότι αυτόματος επιλέγουν και τη γλώσσα προγραμματισμός που θα υλοποιεί το backend του προγράμματος που θα χειρίζεται ο διαχειριστής καθώς και θα επικοινωνούν και με την εφαρμογή του κινητού. Καθώς και διαμορφώνουν και το οικονομικό πλάνο του έργου είναι πάρα πολύ βασικό στον ανταγωνιστικό το προϊόν να αλλά δυστυχώς ίδιας κατηγορίας. Ο βασικότερος παράγοντας για να επιλέξουμε server είναι οι τεχνολογίες που μας παρέχει η γλώσσα και ο server . Διότι μία εφαρμογή δεν φτάνει να είναι οικονομική αλλά πρέπει να έχει κατασκευαστή από νέες τεχνολογίες όπου κι αυτός είναι ένας παράγοντας που τις κάνουν πιο ποιοτικές και με περισσότερες δυνατότητες .Οι επιλογές που είχα ήταν οι παρακάτω :

Windows server (iis)

Θετικά

Υπαρχή μεγάλη προσφορά στην αγορά εργασίας σε τεχνολογίες της microsoft όπως c# και .net και οι περισσότερες εμπορικές εφαρμογές στην Ελλάδα. Υποστηρίζεται από μεγάλη εταιρία.

Αρνητικά

Το visual studio χρειάζεται συνδρομή επόμενος η υλοποίηση γίνεται μόνο σε περιβάλλον windows επίσης και το OS του server. Γενικά τα windows δαπανούν περισσότερους πόρους του συστήματος σε σχέση με άλλα λειτουργικά συστήματα. Τέλος θα χρειάζονταν να μάθω c# όπου δεν έχω ξανά ασχοληθεί στο παρελθόν έτσι θα αυξανόταν ο χρόνος παραδόσεις του project.

Apache

Θετικά

Των χρησιμοποιούν οι περισσότερες εταιρείες στην Ελλάδα για web λύσεις. Η php είναι πολύ εύκολη σαν γλώσσα προγραμματισμού server είναι cross-platform τρέχει κύριος σε Linux OS με αποτέλεσμα να χρειάζεται χαμηλούς πόρους το σύστημα.

Αρνητικά

Χαμηλές δυνατότης και επέκτασης σε σχέση με άλλους server enterprise edition. Το σύστημα έχει πολλά κενά με αποτέλεσμα να είναι ευάλωτο σε κακόβουλες επίθεσης . Δεν υπάρχει support από κάποια εταιρεία αυτό μπορεί να αποτελέσει πρόβλημα σε μεγάλα έργα.

ΤΕΧΝΙΚΉ ΑΝΑΛΥΣΗ ΣΥΣΤΉΜΑΤΟΣ

Εγκαταστήσει Εργαλείων

1. Java 1.8 jdk
2. NetBeans
3. Glassfish 4.1
4. Mysql
5. Android Studio
6. Android Sdk

Δημιουργία Βασης Δεδομενων

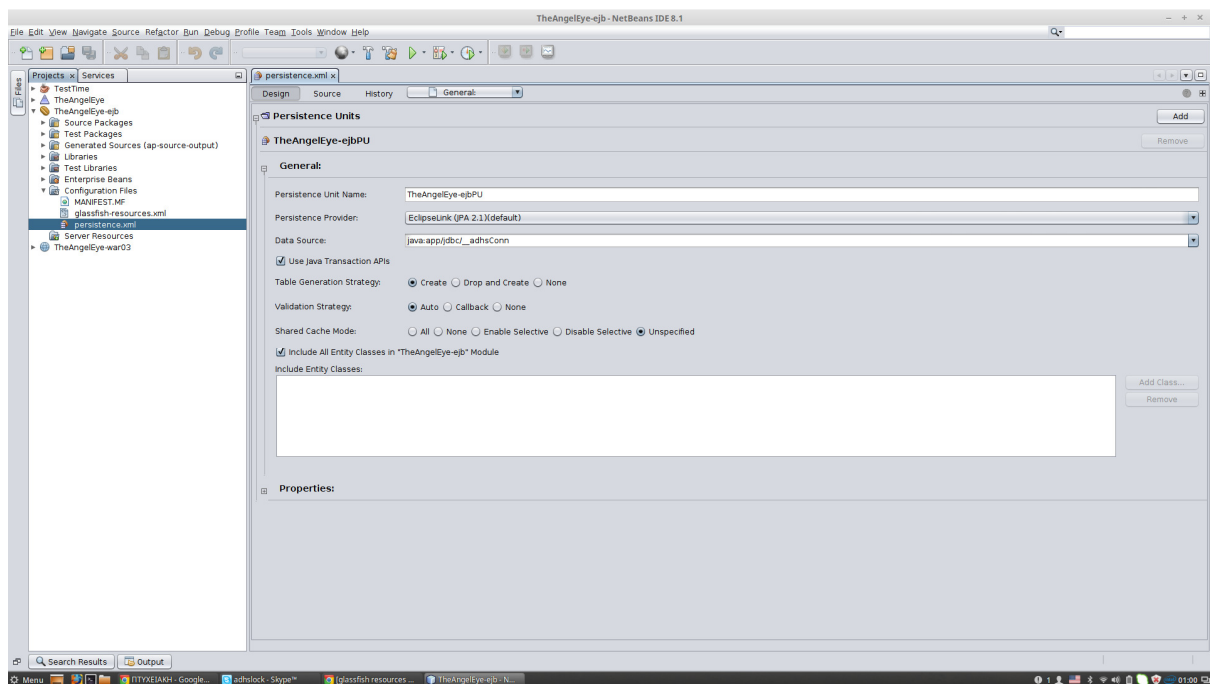
Έγινε χρήση ενός εργαλείου της java το jpa java persist api δίνει την δυνατότητα στους developer να δημιουργούν με αντικειμενοστραφής λογική τοις βάσης.

Το jpa χρησιμοποιεί έναν driver το jdbc για να γίνει η διασύνδεση με την database. Απαραίτητο είναι να δημιουργήσουμε ένα αρχείο το persistence.xml στο αρχείο σε ποια database οι κλάσεις θα γίνουν table

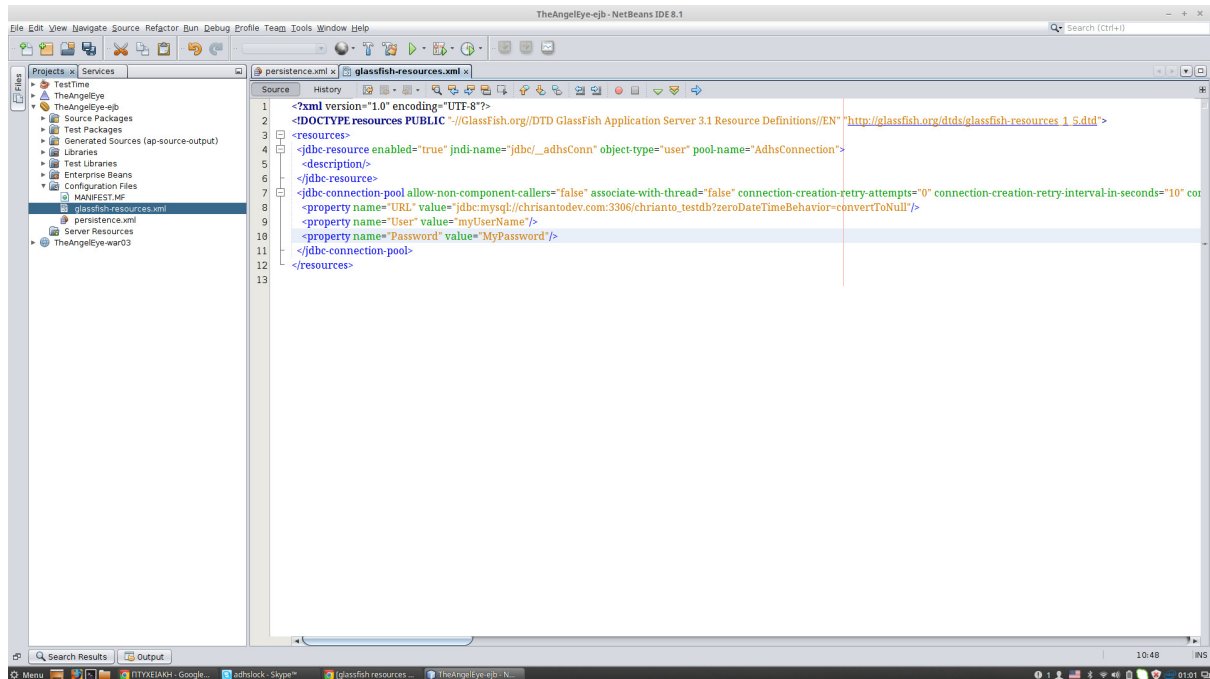
καθώς και πολλές άλλες ρυθμίσεις σε σχέση με την επικοινωνία της εφαρμογής και της βάσης. Επειδή όμως η εφαρμογή είναι web και θα τρέχει στον glassfish θα πρέπει να φτιάξουμε ακόμα ένα αρχείο JDBC Resource είναι ένα xml αρχείο που χρειάζεται ο glassfish για να γίνει το transaction με την βάση και το application το user name το password και άλλες επιλογές.

JPA

persistence.xml



glassfish resource



Παράδειγματα

Entity Class

@Entity

```
public class User implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private Long id;
```

```
    @Column(unique=true)
```

```
    private String email;
```

```
    private String firstName;
```

```
    private String lastName;
```

```
    private String phoneNumber;
```

```
    private int pin;
```

```
    private String Image;
```

```
    ....
```

```
//Getter and setter
```

OneToOne relationship

```
@Entity
public class UserTimer implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @OneToOne
    private User user;
    ....
}
```

inheritance entity

```
@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public abstract class Location {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    protected Long id;

    ...
}

@Entity
public class UserLocation extends Location implements Serializable {

    private static final long serialVersionUID = 1L;
}
```

JPQL

Java persistence query language αυτή η βιβλιοθήκη παρέχει την δυνατότητα μέσα από την java να κάνουμε ερωτήματα στην database.

```
@NamedQueries({
    @NamedQuery(name = "userCount", query = "SELECT COUNT(u.id) FROM
User u"),
    @NamedQuery(name = "findEmail", query = "SELECT u FROM User u
WHERE u.email=:email")
})
```

Πολύ σημαντικό είναι το σύμβολο : όταν κάνουμε query με παραμέτρους προστατεύει την database από κακόβουλες επιθέσεις όπως mysql injection .

```
@PersistenceContext
private EntityManager manager;

....
TypedQuery<User> query = manager.createNamedQuery("findEmail",
User.class)
    .setParameter("email", user.getEmail());
```

EJB

Enterprise Java Bean είναι από τα βασικότερα στοιχεία της JEE για την υλοποίηση μεγάλων και κατανεμημένων συστημάτων . Υπάρχουν αρκετές κατηγορίες από EJB όπως stateless , stateful , entity bean , message drive bean, Timer Session Bean και άλλα όμως όλα τα μπορούν να είναι local remote . Τα remote ejb έχουν την δυνατότητα να τρέχουν και σε απομακρυσμένα συστήματα ενώ τα local μόνο τοπικά.

Παραδείγματα

Stateful-Session-Bean

Το stateful bean δημιουργείται όταν κάνει request ο client το πιο είναι συνδέεται μαζί του και όταν τελειώσει η διαδικασία το γίνεται destroy .

```
@Stateful
@LocalBean
public class AdminServices {

    Public String getMsg(){return "hello";}

    }//call ebj

    StatefulBean statefulBean = lookupStatefulBeanBean();

statefulBean.getMsg();
...
    private StatefulBean lookupStatefulBeanBean() {
        try {
            Context c = new InitialContext();
            return (StatefulBean)
c.lookup("java:global/JavaEETutorial/JavaEETutorial-
ejb/StatefulBean!bean.StatefulBean");
        } catch (Exception e) {}
    }
}
```

Stateless-Session-Bean

Το stateless δεν έχει καμία συσχέτιση με των client υπάρχει ένα σύνολο από αυτά και κάθε φορά ένας πελάτης καλή ένα bean όπου η διαδικασία δεν συσχετίζεται με την προηγούμενη.

```
@Stateful
@LocalBean
public class StatefulBean {

    public String myFun() {
        return "Hellow Bean" ;
    }
}
//call bean

@Ejb
Private StatefulBean bean;

bean.myFun();
```


Singleton Bean

Είναι καλύτερο από θέμα αποδόσεις και για την υλοποίηση μεγάλων διεργασιών άλλα μπορεί να δημιουργήσει προβλήματα για παράδειγμα όταν επικοινωνεί με την βάση δεδομένων. Δημιουργείτε μονό μια φορά για το κάθε request .

```
@Singleton
@LocalBean
public class SingletonBean {
    public String myFun() {
        return "Hellow Bean";
    }
}
//call bean

@Ejb
Private SingletonBean bean;

...

```

Timer(services) EJB

Είναι ένα singleton ejb το όπιο τρέχει στο παρασκήνιο της εφαρμογής δεν είναι καλή πρακτική να έχει προσπαθεί ο client σε αυτά . Αποτελεί καλή επιλογή για παράδειγμα όταν μια εταιρεία κάθε πρώτη του μήνα θέλει να πληρώσει το προσωπικό της αυτή είναι μια διαδικασία συγκεκριμένη που γίνεται σε συγκεκριμένη χρονική στιγμή.

```

@Singleton
@LocalBean
@Startup
public class BackEndService {
    @PostConstruct
    private void init() {
        timerService.createTimer(2000, 60000, "IntervalTimerDemo_Info");
    }

    @Timeout
    public void execute() {
    ...
    }
}

```

Java Servlets

Τα servlets είναι είμαι java class η όποια δίνει υλοποιεί το http μεταξύ server και client κάνοντας ο client request και περνώντας response έτσι μπορεί να μιλήσει με την εφαρμογή μέσω των μεθόδων doGet και doPost ,κληρονομεί χαρακτηριστικά και μεθόδους class HttpServlet. Για να πάρουμε τοις παραμέτρους που θέλει ο χρήστης να στείλει στον server κάνουμε χρήση της request.getParameter("id"). Το url από το κάθε servlet το ορίζουμε σε ένα που ονομάζετε web.xml .

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
  <servlet>
    <servlet-name>One</servlet-name>
    <servlet-class>servlets.One</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>Two</servlet-name>
    <servlet-class>servlets.Two</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>One</servlet-name>
    <url-pattern>/One</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Two</servlet-name>
    <url-pattern>/Two</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>

```

Το response του server μπορεί να είναι μια άλλη web page ή ένα μήνυμα με μια συγκεκριμένη μορφή json ή xml. Στο project έγινε χρήση του json format το οποίο είναι ένα string αλφαριθμητικό .

```

{"employees": [
  {"firstName": "John", "lastName": "Doe"},
  {"firstName": "Anna", "lastName": "Smith"},
  {"firstName": "Peter", "lastName": "Jones"}
]}

```

JAX-RS

Είναι μια τεχνολογία η όποια ξεκίνησε από την java 6 και μεταφέρει τα δεδομένα με εύκολο και γρήγορο τρόπο μέσω του http . Είναι πολύ απλά εύκολα στην χρήση τους. Έχουν σχεδόν τοις ίδιες λειτουργίες με τα servlets. Τα Urls πρέπει και αυτά να δηλωθούν στο web.xml .Κάνουν χρήση annotation .

```
@Path("getuser")
@GET
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public User selectUserById(@QueryParam("id") String id) {
    Long uid = Long.parseLong(id);
    return adminServices.getUserID(uid);
}
```

Όταν στο url που έχουμε ορίσει δώσουμε την παράμετρο id=... θα πάρουμε ένα json από τον server. Μέσα στην ίδια κλάση μπορούμε να φτιάξουμε όσα url θέλουμε και function με διαφορετικές λειτουργίες .Επίσης μπορούμε να επιλέγουμε το είδος με το οποίο θα πάρουμε τα δεδομένα και το είδος του response που θα δώσει ο server.

JAAS

Java Authentication and Authorization Service είναι μια τεχνολογία της oracle που έχει ως στόχο να υλοποιούνται ασφαλείς εφαρμογές και να προστατεύονται δεδομένα και λειτουργίες . Πιο συγκεκριμένα να γίνεται η επαλήθευση του κάθε χρήστη , να υπάρχει έλεγχος πρόσβασης της

λειτουργίες της εφαρμογής που θα έχει πρόσβαση .Στα πλαίσια της διπλωματικής ο τρόπος που έγινε χρήση αυτής για να έχουν πρόσβαση στην εφαρμογή μονό οι χρήστες και ο διαχειριστής .

Ανάλυση βημάτων :

1. Δημιουργία ενός πίνακα με τους χρήστες (users) και ενός δευτέρου πίνακα που θα περιγράφει σε ποια ομάδα (groups) ανήκει ο κάθε ένας . Συνδέσουμε τα groups με τους users με σχέση ένα προς ένα (OneToOne).
2. Θα χρειαστεί να κάνουμε κάποιες ρυθμίσεις στον glassfish μέσα από το admin console.

The image shows two screenshots from the GlassFish Admin Console. The top screenshot displays the 'Realms' configuration page for the 'server-config' configuration name. It shows a table of four realms: JDBCRealm, admin-realm, certificate, and file. The bottom screenshot shows the configuration details for the 'JDBCRealm' realm, including fields for JAAS Context, JNDI, User Table, User Name Column, Password Column, Group Table, Group Table User Name Column, Group Name Column, and Password Encryption Algorithm.

Select	Name	Class Name
<input type="checkbox"/>	JDBCRealm	com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm
<input type="checkbox"/>	admin-realm	com.sun.enterprise.security.auth.realm.file.FileRealm
<input type="checkbox"/>	certificate	com.sun.enterprise.security.auth.realm.certificate.CertificateRealm
<input type="checkbox"/>	file	com.sun.enterprise.security.auth.realm.file.FileRealm

JAAS Context:	jdbcRealm
JNDI:	jdbc/_javaEETutorial
User Table:	USERS
User Name Column:	USERNAME
Password Column:	PASSWORD
Group Table:	GROUPS
Group Table User Name Column:	USER_ID
Group Name Column:	GROUPNAME
Password Encryption Algorithm:	MD5

3. Μέσα στο web project > web.xml θα χρειαστεί να ορίσουμε σε ποια url θα έχει πρόσβαση η κάθε ομάδα .

```

<security-constraint>
  <display-name>admin</display-name>
  <web-resource-collection>
    <web-resource-name>secureAdmin</web-resource-name>
    <description/>
    <url-pattern>/secureAdmin/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description/>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <display-name>user</display-name>
  <web-resource-collection>
    <web-resource-name>user</web-resource-name>
    <description/>
    <url-pattern>/secureUser/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description/>
    <role-name>user</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>JDBCRealm</realm-name>
</login-config>
<security-role>
  <description/>
  <role-name>admin</role-name>
</security-role>
<security-role>

```

4. New > Glassfish > descriptor .Θα δημιουργηθεί το αρχείο glassfish-web.xml στο όπιο θα οριστούν οι ομάδες και ο ρόλι που θα έχουν στην εφαρμογή .

```
<security-role-mapping>
  <role-name>admin</role-name>
  <group-name>admin</group-name>
</security-role-mapping>

<security-role-mapping>
  <role-name>user</role-name>
  <group-name>user</group-name>
</security-role-mapping>
```

Όταν θα προσπαθήσει κάποιος να χρησιμοποιήσει τα urls που έχουμε βάλει security auth θα του ζητηθεί username και password .

FRONT-END

Για να παρέχουμε την δυνατότητα να αλληλεπιδρά ο διαχειριστής με την εφαρμογή και των server υλοποιήθηκε μια διεπαφή το γραφικό περιβάλλον η user interface ,όλες αυτές οι δυνατότητες θα παρέχονται μέσω του browser .Το user interface περιέχει έναν χάρτη google maps στο όπιο εμφανίζεται η τοποθεσία του client μια λίστα με τους χρήστες που ζήτησαν βοήθεια . Έγινε χρήση από της παρακάτω τεχνολογίες : html , css , javascript (jquery , ajax) .

Με την html και η css υλοποιήθηκε το γραφικό περιβάλλον και η διαδραστικότητα του site έγινε με javascript (jquery,ajax) . HTML δημιουργεί το κείμενο και η css κάνει ακόμα πιο φιλικό στον χρήστη δίνοντας διαφορετική μορφή . Η λειτουργικότητα του site βασίζεται στην javascript η οποία είναι client side τρέχει από την πλευρά του user ώστε να υπάρχει διαδραστικότητα καθώς και για των χάρτη της google.Για να μπορεί το site να εκτελεί λειτουργίες στο παρασκήνιο χρειάζεται ajax Asynchronously Javascript and XML στην ουσία κάνει Http request σε άλλα url και το response που παίρνει μπορούμε να το αναπαραστήσουμε στο site χωρίς να χρειάζεται να ανανέωση ο διαχειριστής συνέχεια την σελίδα .

ajax-jquery

```
$.ajax({
  type: 'GET',
  url: '/TheAngelEye/webresources/adminrs/getallusers',
  dataType: 'json',
  success: function (data) {
    //data is the response of the request
  }
})
```

Όταν θέλουμε μια κομμάτι κωδικά να εκτελείται συνέχεια θα κάνουμε χρήση μια συνάρτησης της javascript setInterval .

```
setInterval( function () {
  getAlarms();
}, 1000);
```

Google Maps

Για να έχει το site χάρτη πρέπει να κάνουμε την παρακάτω διαδικασία :

1. Δημιουργία gmail.
2. Να κάνουμε καινούργιο project στο developer console της google και να δημιουργήσουμε το api του google maps μετά θα μας δώσει ένα κλειδί το οποίο πρέπει να το ορίσουμε μέσα στο παρακάτω κομμάτι κωδικά .


```

<div id="map"></div>

<script>

  var map;

  function initMap() {

    map = new google.maps.Map(document.getElementById('map'), {

      center: {lat: -34.397, lng: 150.644},

      zoom: 8

    });

  }

</script>

<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>

```

ANDROID

Το android είναι ένα λειτουργικό σύστημα σε κινητές συσκευές το οποίο είναι βασισμένο στο μοντέλο του Linux . Αυτό το κάνει γρήγορο και ασφαλές . Για την ανάπτυξη εφαρμογών γίνεται με την java και xml με εργαλεία όπως το android studio και το eclipse .Η κάθε εφαρμογή μπορεί να πάρει κάποια δικαιώματα στο κινητό όπως πρόσβαση στην camera η στις έπαθες και της κλήσης αυτά πρέπει να τα ορίσει ο προγραμματιστής σε ένα αρχείο manifest.xml .

Για το user interface η υλοποίηση γίνεται με xml άλλα μπορεί να γίνει και με java .Το back end γίνεται με java . Τέλος έγινε και χρήση από κάποια open source library .

Η εφαρμογή αποτελείται από activitys services και broadcastreiceiver όλα τα παραπάνω είναι κλάσης οι που έχει η φτιάξη η google , κάθε μια έχει διαφορετικό ρολό στο σύστημα .

Activitys

Τα activity είναι δραστηριότητες κάθε ένα από αυτά έχει και ένα αντίστοιχο xml αρχείο που εκεί μέσα παράγουμε τα UI elements κουμπιά φωτογραφίες πεδία κείμενου και άλλα .Κάθε activity πρέπει να οριστεί μέσα στο manifest.xml . Κάθε UI element για να είναι λειτουργικό πρέπει να έχει οριστεί και στο activity εκτός από το xml αρχείο και συνεχονται με το id που έχει το κάθε ένα . Περιέχει της παρακάτω μεθόδους :

Services

Σκοπός αυτής της κλάσης είναι να τρέχει στο παρασκήνιο της εφαρμογής ώστε να μην επηρεάζουν των χρήστη μεγάλες διεργασίες που μπορεί να κάνει και το κύριο νήμα της εφαρμογής .Πρέπει να οριστεί στο manifest.xml :

```
<service
  android:name=".nameOfClass"
  android:enabled="true"/>
```

Στην εφαρμογή το service θα ενημερώνει την τοποθεσία που βρίσκεται συσκευή σε πραγματικό χρόνο όταν η εφαρμογή είναι στο background .

Broadcast receiver

Είναι η απάντηση από ένα broadcast μήνυμα από την ίδια την εφαρμογή η από ένα γεγονός του συστήματος , την ώρα που έχουμε μια εισερχόμενη κλήση στο κινητό μας και εμφανίζεται στο προσκήνιο υπάρχει ενεδ broadcast receiver που μέσα του υπαρχή ένα φίλτρο που ενεργοποιεί ανάλογες διαδικασίες.

Στην εφαρμογή έγινε έπρεπε να παρακολουθήσουμε κάποια συγκεκριμένα μηνύματα του συστήματος αυτά χρειάζεται να οριστούν στο αρχείο manifest.xml .

```

<receiver
    android:name=".nameOfClass"
    android:enabled="true">
<intent-filter>
    <action
        android:name="android.intent.action.MEDIA_BUTTON"/>
</intent-filter>

</receiver>
<receiver
    android:name=".nameOfClass"
    android:enabled="true">

```

Intent - Filters

Intent είναι μια κλάση η οποία με αυτή μπορούμε να εκτελέσουμε μια διαδικασία που ορίζουμε όπως να ξεκινήσει ένα activity ένα service η ακόμα και ένα broadcast receiver .Παράδειγμα μέσα από μια την εφαρμογή μας πάντοτες εντά κουμπι να καλέσει το κινητό έναν αριθμό που έχουμε ορίσει μέσα στην εφαρμογή .

Fragments

Είναι μια υποκατηγορία του activity που έχει το δικό του layout μέσα σε ένα activity άλλα εξαρτάται από αυτό και έχει της δικαίες του ανεξάρτητες κατάστασης onCreate onPause, onDestroy .

Location

Το πλέον σχεδόν όλες οι android συσκευές υποστηρίζουν την τεχνολογία του location όπου αυτή παρέχεται μέσω του gps του έχει η συσκευή αλλά και με άλλους πιο εξελιγμένους τρόπους cell-id και wi-fi. Για να έχει η εφαρμογή αυτή την δυνατότητα πρέπει να δηλωθεί μέσα στο αρχείο manifest.xml . Μπορούμε να πάρουμε και άλλα στοιχεία με το Location API όπως την ταχύτητα του χρήστη .

UI-LAYOUT

Το για να φτιαχτεί ένα user interface υπάρχει η κλάση View όπου μπορούμε να δημιουργήσουμε view objects όπως button textview . Όλα τα ui objects μπορούν να δημιουργηθούν σε αρχεία xml που είναι στην φάκελο res/layout χρειάζεται και το αντίστοιχο αντικείμενο μέσα στο activity η σύνδεση μεταξύ του γίνεται μέσω του id , αλλά και δυναμικά μέσω της java .

Connecting to the Network

Η εφαρμογή επικοινωνεί με των server μέσω του πρωτοκόλλου http.Στέλνει δεδομένα με post και get και το response είναι σε μορφή json. Ωστόσο αυτό για τις εφαρμογές δεν είναι μια απλή διαδικασία καθώς δεν γίνεται να εκτελείτε το mian thread που διαχειρίζεται το UI .Είναι απαραίτητο να γίνεται στο background και αφού ολοκληρωθεί να εμφανίζονται να τα αποτελέσματα . Για γίνει ακόμα πιο γρήγορη αυτή η διαδικασία έγινε χρήσιμη μια open source βιβλιοθήκης από την square με το όνομα retrofit . Στην παρακάτω διεύθυνση υπάρχουν παραδείγματα για το πως μπορούμε να χρησιμοποιήσουμε [retrofi home page](#) .

Android wearble

Η εφαρμογή επεκτείνεται και σε wearble devices . Αυτό υλοποιήθηκε για να δίνει την δυνατότητα στους χρήστες να χρησιμοποιούν πιο γρήγορα και εύκολα κάποιες λειτουργίες της εφαρμογής . Παράδειγμα αν χρειάζεται κάποιος άμεσα βοήθεια να μην χρειαστεί να βγάλει το κινητό του από την τσέπη του .

Η σύνδεση μεταξύ των δυο συσκευών γίνεται μέσω του bluetooth protocol . Με την βοήθεια της εφαρμογής Android Wear [Download](#) η επικοινωνία του application στο mobile και στο wearble γίνεται σε πιο υψηλό επίπεδο για των προγραμματιστή .

Η χρειάστηκε να χρησιμοποιήσουμε την βιβλιοθήκη google play services και πιο συγκεκριμένα την τεχνολογία google cloud messaging . Το επόμενο βήμα είναι ότι υλοποίησα έναν sender και ένα receiver όπου είναι απαραίτητη η χρήση του google api (WearableListenerService ,GoogleClientApi).

WearableListenerService με αυτή την κλάση έχουμε την δυνατότητα επέκτασης έτσι ώστε να λαμβάνει το mobile τα πακέτα που προέρχονται από το wearble .

GoogleClientApi υλοποιεί την διασύνδεση του wearble με το mobile και την μετάδοση πακέτων . Έτσι ο χρήστης όταν πατήσει το κουμπί από την εφαρμογή στο wearble θα σταλθεί ένα πακέτο στο κινητό και αυτό θα συνέχιση την διαδικασία .

ΣΥΜΠΕΡΑΣΜΑΤΑ

JPA έγινε πολύ γρήγορα η δημιουργία της βάσης δεδομένων καθώς και ο τρόπος που γίνονται τα ερωτήματα .

Ejb είναι καλύτερη υλοποίηση του business logic επίσης δεν χρειάζεται να διαχειριζόμαστε το life cycle του κάθε ejb και έχουν πολλές δυνατότητες όπως τα ejb service .

JAX-RS είναι αρκετά γρήγορα διότι δεσμεύουν λίγους πόρους από των server ο εναλλακτικός τρόπος είναι με websocket .

Το gps στο android καταναλώνει πολύ γρήγορα την μπαταρία .

Όσο πιο πολλά γραφικά έχουμε στο android όπως background images , custom buttons , animations και άλλα αυτό κάνει την εφαρμογή να δεσμεύει περισσότερη μνήμη και να έχει πιο αργή ανταπόκριση .

Αν ξανά έκανε την εφαρμογή η επιλογή του server θα ήταν διαφορετική θα επέλεγα των wildfly της jboss επειδή χρησιμοποιείται από μεγάλες εταιρείες στην αγοράς εργασίας .

Κάποια module της εφαρμογής δεν λειτουργούν σε όλες τις συσκευές (bluetooth action) .