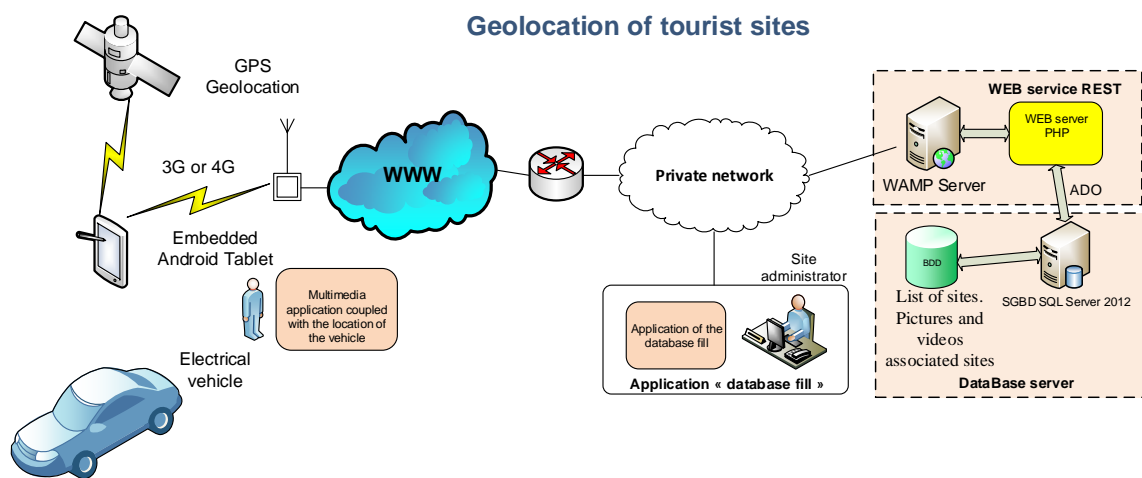


### Presentation file of the project topic

<b>High school or Training Center: Institut Lemonnier</b>
<b>Town: Caen</b>
<b>Project name : « LiveTourism » application on Android</b>

#### 1. Presentation of the project

##### Operating configuration



#### 2. Expression of the requirements

The project will consist in equipping the headrest backsides of the front seats of a car with digital tablets provided with an application enabling to discover the places of interest located in the direct environment of the vehicle.

The GPS function of the tablets will enable the application to locate the vehicle, while the 3G or 4G modem function will enable the same application to request a remote server to obtain information about the places of interest in the immediate environment of the vehicle.

The search radius of the tourist attractions around the vehicle, the type of the sites, as well as the mode of dissemination of the information on the sites will be configurable. Either the application will automatically select the display of the presentation of the closest site, or it will propose, in the form of icons, the sites present in the immediate environment of the vehicle and leave the passengers the choice of discovering one site. Optionally, it will be possible to select one of the proposed sites, in order that the application navigation guides you to the selected site.

The presentation of the site will be displayed in the form of a Web content in a fun way with images and a possibility of short videos.

The data on tourist attractions will be stored on a database server, SQL server 2012, and made accessible from the on-board application thanks to a web service hosted by a web server. The data returned through the web service will be encapsulated into a JSON format.

A so-called "provider" application will be used for the site manager to populate the database with the tourist information (site, description of the site, pictures and videos of the site, etc...).

The development of the multimedia contents will be carried out by themes with as objective the discovery of the 3 regions of the 3 schools concerned by the Erasmus project. For the Institut Lemonnier, an almost mandatory theme would be for example, the discovery of the landing beaches of June 6<sup>th</sup> 1944 in Normandy.

Features of the embedded application Android.

In summary, the embedded Android application should make it possible to:

- ✓ Configure the settings for accessing the web service.
- ✓ Select the region, type of sites, and the radius of search for the sites that you want to be able to discover "in live" as the vehicle moves.
- ✓ Locate the vehicle on a map with a display in road or mixed mode (road and satellite).
- ✓ Display the sites according to the type selected in the radius of search on the map.
- ✓ Display, by clicking on a site, the information available on that site, the viewing of the pictures and if possible reading of the videos.
- ✓ Activate a mode of automatic display of the information on the nearest site.

In addition, the application should be able to detect the presence or not of an active network access via the 3G or 4G connection. If no access is possible, the application should inform about it and prohibit any attempt to access the network services.

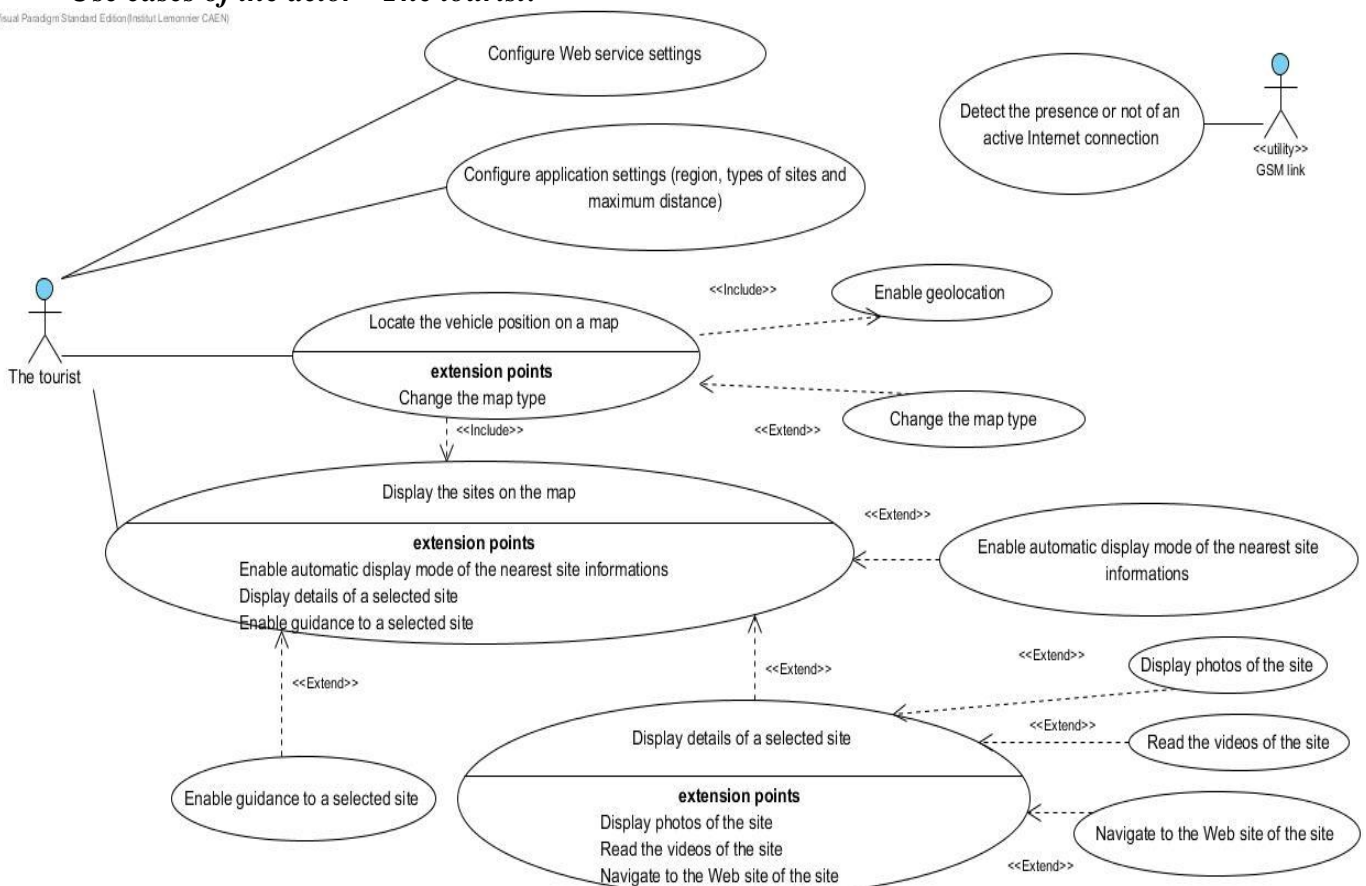
The device will be installed with a landscape mode orientation, the application must therefore ensure that the view is locked in a landscape mode exclusively.

It should also take into account the language specified on the operating system: French (default language) as well as English and Italian.

The embedded Android application highlights a single actor that we shall call "the tourist". This actor is involved in the following scenarios:

**Use cases of the actor "The tourist":**

Visual Paradigm Standard Edition (Institut Lemonnier CAEN)



**Use case "Configure Web service settings":**

**Description:** the tourist configures his application with his parameters for accessing Web services (IP address or DNS name and port listener) in order to consult the information related to the tourist sites.

**Actor concerned:** the tourist.

**Use case "Configure application settings":**

**Description:** the tourist configures the tourist settings corresponding to the sites which he wants to see, the region of visit, the type of site (Museum, natural, cultural, historic sites, etc...) as well as the maximum distance of the site from the vehicle.

**Actor concerned:** the tourist.

**Use case "Locate the position of the vehicle on a map":**

**Description:** the tourist can see his position (that one of the vehicle) on a map. He can change the type of map, road map or mixed (road map and satellite view). If the geolocation is not enabled, he will be required to activate it.

**Actor concerned:** the tourist.

**Use case "Select the map type":**

**Description:** the tourist can choose the type of map displayed, road map, or mixed (road map and satellite view).

**Actor concerned:** the tourist.

**Use case "Enable Geolocation":**

**Description:** the tourist activates the geolocation necessary to display his location on the map.

**Actor concerned:** the tourist.

**Use case "Show tourist sites on the map":**

**Description:** the tourist enables the display of the tourist sites on the map. Optionally an automatic reading mode of the closest site can be enabled.

**Actor concerned:** the tourist.

**Use case "Enable the automatic display mode of the nearest site":**

**Description:** the tourist activates the automatic mode, the display of the information associated to the nearest site will appear automatically.

**Actor concerned:** the tourist.

**Use case "View detailed information on a selected site":**

**Description:** the tourist selects a site on the map to view detailed information, description, theme, list of photos and videos associated, etc.

**Actor concerned:** the tourist.

**Use case "Enable guiding to a selected site":**

**Description:** the tourist activates the display of the route guidance to the selected site.

**Actor concerned:** the tourist.

**Use case "Show pictures of the site":**

**Description:** the tourist, after having viewed the detailed information on the site, can choose to visualize the photos of the list associated to this site.

**Actor concerned:** the tourist.

### Use case "Read the videos of the site":

**Description:** the tourist, after having viewed the detailed information on the site, can choose to visualize the videos of the list associated to this site.

**Actor concerned:** the tourist.

### Use case "Navigate to the Web site of the site":

**Description:** the tourist, after having visualized the details of the site can, if there is any, choose to browse to the official Web site of the tourist site.

**Actor concerned:** the tourist.

## 3. Preliminary arguments available and constraints of realization

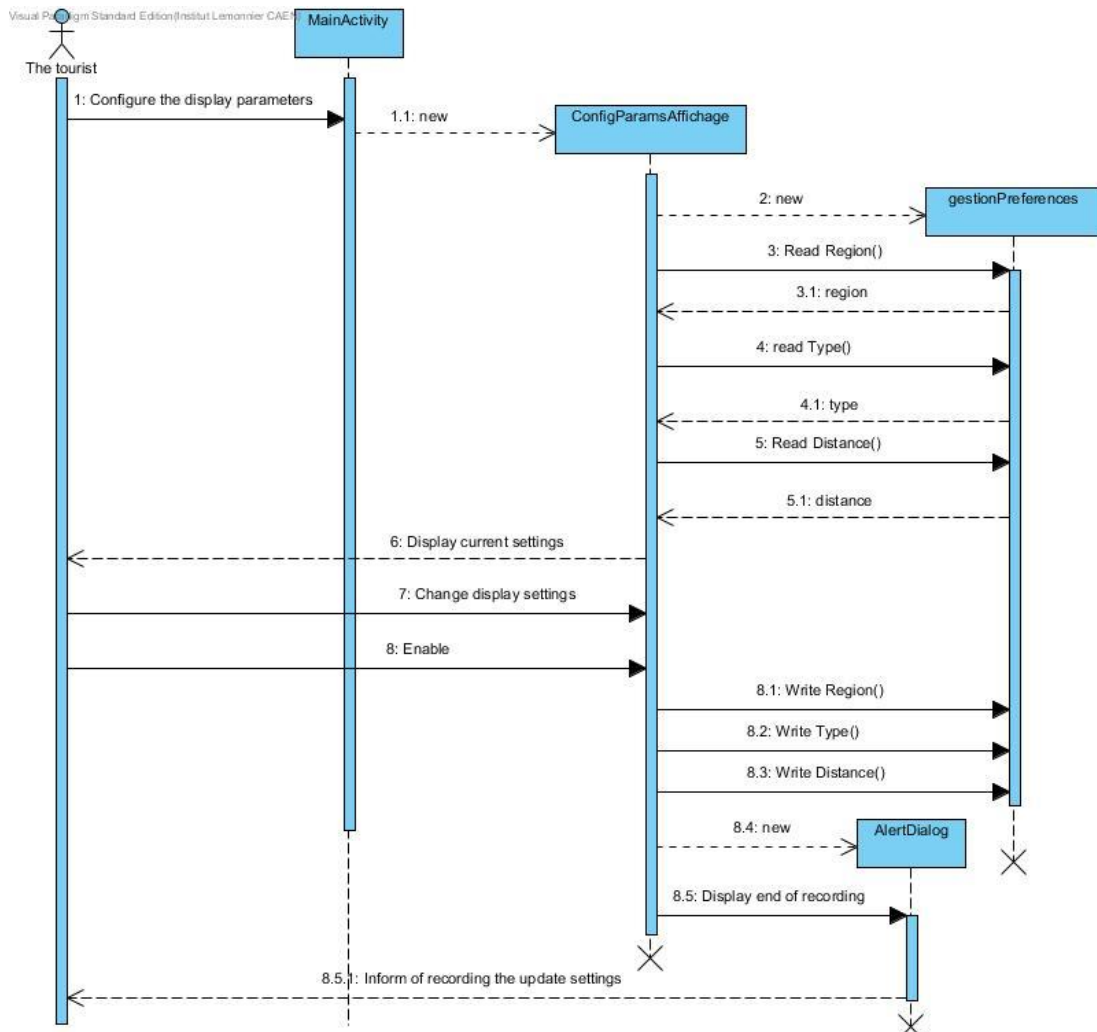
Mobile application written for Android 4 API 14 Ice Cream Sandwich or later.

### 3.1. Specifications

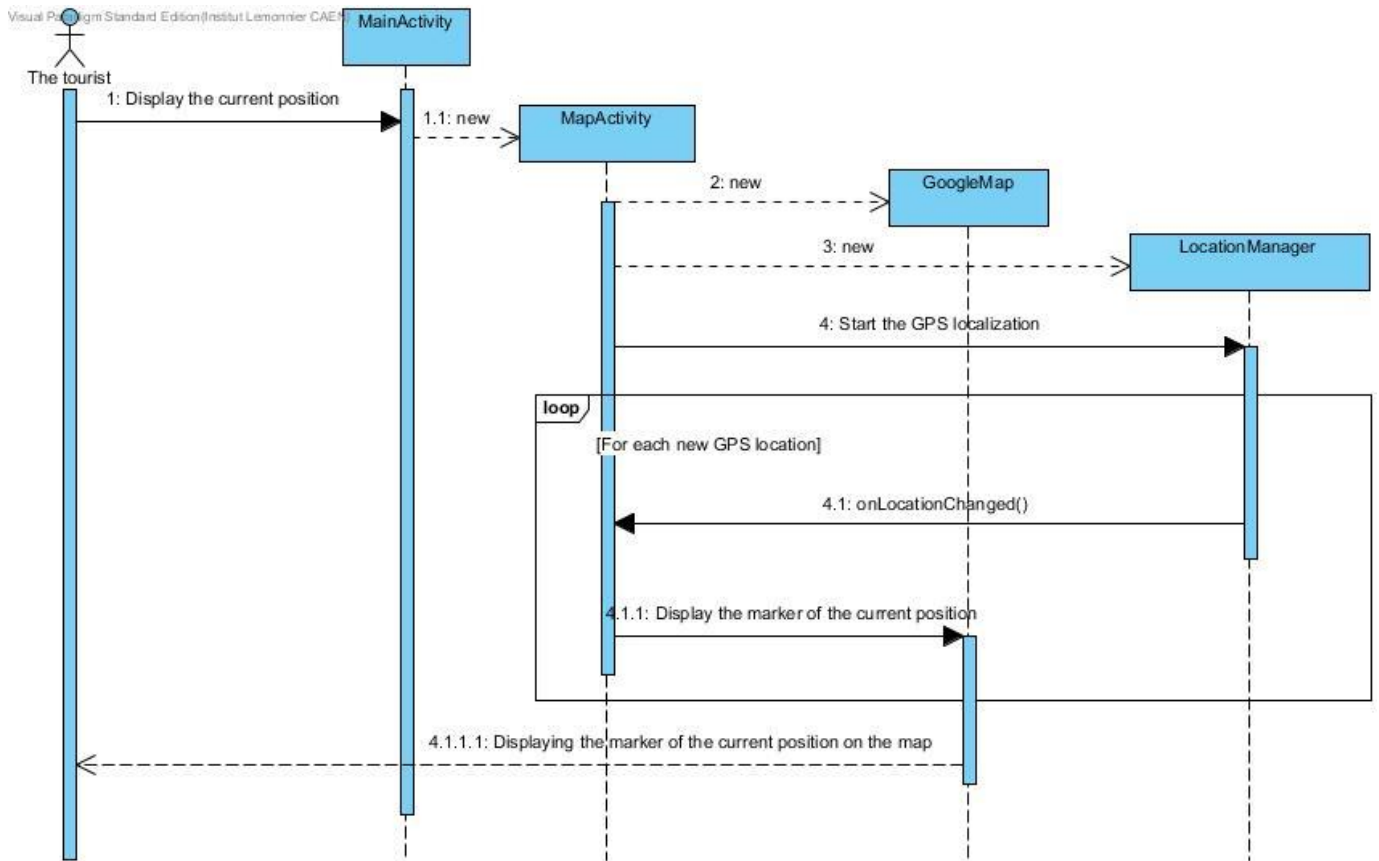
#### Starting point of the analysis.

The starting point of the analysis includes use cases above and sequence diagrams provided below. Only sequences "Configure application settings" and "Locate the position of the vehicle on the map" are presented here for the purpose of a better understanding of the system exchanges. The research concerning other sequences rely upon the responsibility of the project team.

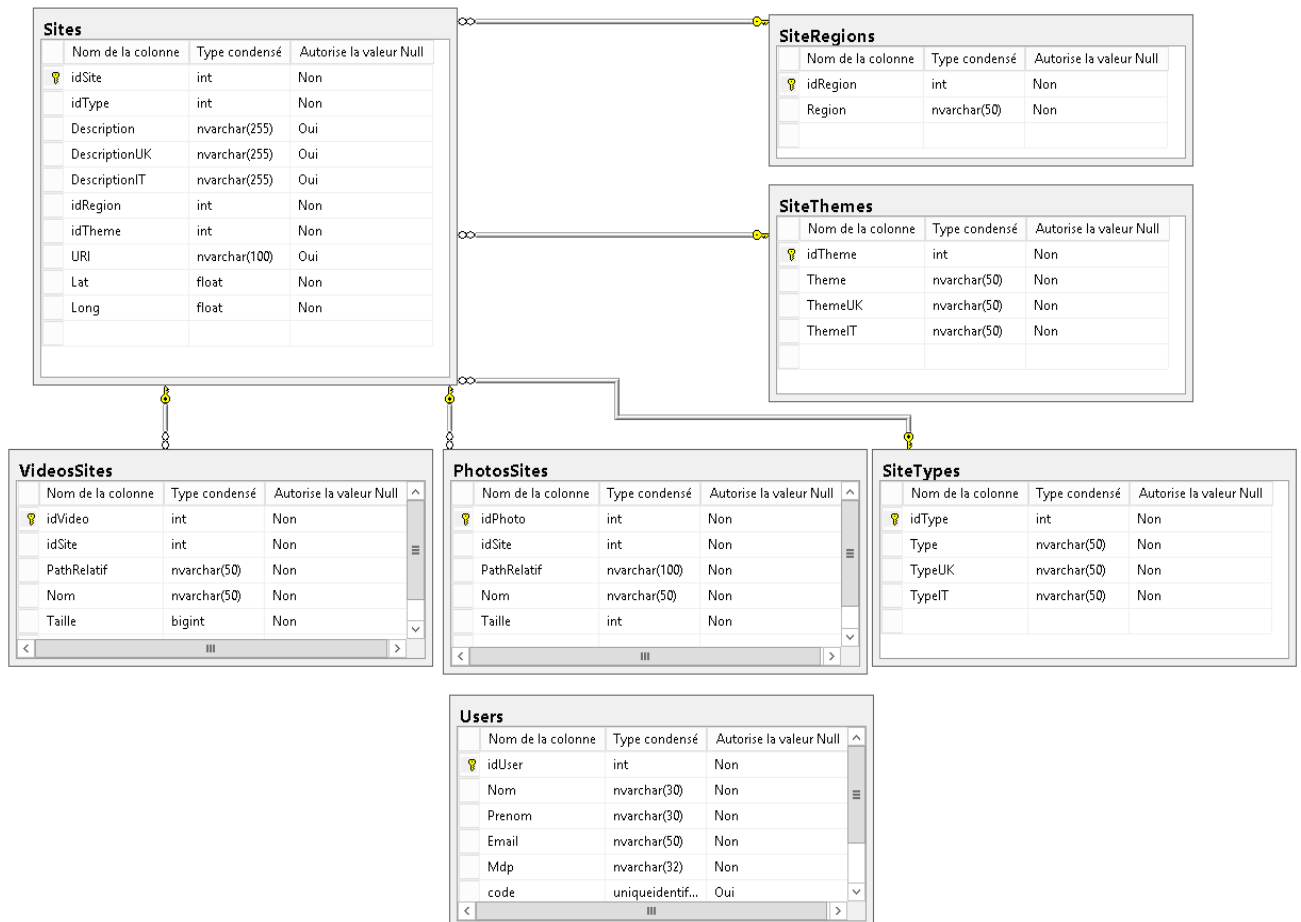
#### Sequence "Configure application settings":



## Sequence " Locate the position of the vehicle on the map " :



## Diagram of the database « BDD ProjERASMUS ».



## The Web service classes :

Sites
-autorise : boolean -pays : string
+Sites(email : string, code : string, pays : string) +selectSite(id : int) : string +selectAllSites() : string +getTypesOfSites() : string +getSitesAroundMeByRegion(region : string, lat : string, lon : string, dist : string) : string +getSitesAroundMeByRegionAndType(region : string, Type : string, lat : string, lon : string, dist : string) : string

VideosSites
-autorise : boolean
+VideosSites(email : string, code : string) +lireVideos(id : int) : string

PhotosSites
-autorise : boolean
+PhotosSites(email : string, code : string) +lirePhotos(id : int) : string

Users
+inscription(email : string, mdp : string, nom : string, prenom : string) : int +connexion(email : string, mdp : string) : string

## The commands and settings of the HTTP Web service :

Android customers should use the URI **/eras/index.php** to access the functionality of the Web service. Commands and parameters of the (case sensitive) Web service accessible via this URI are described below:

The various commands or queries (query) must be specified by the parameter “q”:

- ✓ Query for the user registration on the service: **q = registration**
  - Required query parameters:
    - **email** = user email,
    - **mdp** = desired password for the user,
    - **nom** = username,
    - **prenom** = user’s firstname.
  - Optional query parameter: none.
  - Return value: the identifier of the user registered in JSON format.

*Example of query :*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=inscription&email=paul.auchon@free.fr&mdp=Password1234&nom=auchon&prenom=paul>

**JSON result: {"id":10}**

- ✓ Service connection query from a recorded user: **q = GetCode**
  - Required query parameters:
    - **email** = user email,
    - **mdp** = desired password for the user.
  - Optional query parameter: none.
  - Return value: the temporary code to access the service for the connected user in JSON format.

*Example of query :*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=GetCode&email=paul.auchon@free.fr&mdp=Password1234>

**JSON result:** {"code":"553f2dac7baf6"}

- ✓ Query for the list of all the sites: **q = Sites**
  - Required query parameters:
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter:
    - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
  - Return value: list of all the sites referenced in JSON format.

*Example of query :*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Sites&email=paul.auchon@free.fr&code=553f2dac7baf6>

**JSON result:**

```
{"SiteListe":[{"idSite":1,"Type":"Monument","Description":"Eglise 11° siècle",  
"Theme":"Entre Thue et Mue", "Region":"Calvados",  
"URI":"http://www.calvados.fr","Lat":49.5,"Long":-0.45},  
{"idSite":2,"Type":"Monument  
historique","Description":"Moulin","Theme":"Entre Thue et Mue",  
"Region":"Calvados","URI":"http://www.moulin.fr","Lat":49.3,"Long":-0.352},  
{"idSite":3,"Type":"Site naturel","Description":"A voir  
absolument","Theme":"Côte de  
Nacre","Region":"Calvados","URI":"http://www.maregion.com","Lat":49.3,"Long  
":-0.352}]}
```

- ✓ Query for a site with a specified site identifier: **q = Site**
  - Required query parameters:
    - **idSite** = searched site identifier,
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter:
    - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
  - Return value: the site with the specified identifier in JSON format.

*Example of query :*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Site&email=paul.auchon@free.fr&idSite=3&code=553f2dac7baf6>

**JSON result:**

```
{"Type":"Site naturel","Description":"A voir absolument","Theme":"Côte de  
Nacre","Region":"Calvados","URI":"http://www.maregion.com","Lat":49.3,"Long  
":-0.352}
```

- ✓ Query for the list of all the types of sites: **q = TypesOfSite**
  - Required query parameters:



- **email** = user email,
  - **code** = access code of the connected user.
- Optional query parameter:
  - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
- Return value: list of all the types of sites referenced in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=TypesOfSite&email=paul.auchon@free.fr&code=553f2dac7baf6>

*JSON result:*

```
{"TypeListe":[{"idType":2,"Type":"Monument"}, {"idType":1,"Type":"Monument historique"}, {"idType":3,"Type":"Site naturel"}]}
```

✓ Query for the list of all the themes of sites: **q = ThemesOfSite**

- Required query parameters:
  - **email** = user email,
  - **code** = access code of the connected user.
- Optional query parameter:
  - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
- Return value: list of all the themes of sites referenced in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=ThemesOfSite&email=paul.auchon@free.fr&code=553f2dac7baf6>

*JSON result:*

```
{"ThemeListe":[{"idTheme":1,"Theme":"Débarquement du 6 juin 1944"}, {"idTheme":2,"Theme":"Guillaume le conquérant"}]}
```

✓ Query for the list of all the regions: **q = Regions**

- Required query parameters:
  - **email** = user email,
  - **code** = access code of the connected user.
- Optional query parameter: none.
- Return value: list of all the regions referenced in JSON format

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Regions&email=paul.auchon@free.fr&code=553f2dac7baf6>

*JSON result:*

```
{"RegionListe":[{"idRegion":1,"Region":"Calvados"}]}
```

✓ Query for the list of the sites of a specified region around the specified position: **q = SitesAroundMeByRegion**

- Required query parameters:
  - **region** = region of the sites localized,
  - **lat** = current latitude,



- **lon** = current longitude,
- **dm** = max. great circle distance in meters (search radius) of the sites around the current position,
- **email** = user email,
- **code** = access code of the connected user.
- Optional query parameter:
  - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
- Return value: list of all the sites meeting the query criteria in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=SitesAroundMeByRegion&email=paul.auchon@free.fr&code=553f2dac7baf6&region=1&lat=49.0&lon=-0.3&dm=35000>

*JSON result:*

```

{"SiteListe":{"1":{"idSite":2,"idType":1,"Type":"Monument historique",
>Description":"Moulin","Theme":"Entre Thue et Mue","Region":"Calvados",
>URI":"http://www.moulin.fr","Lat":49.3,"Long":-
>0.352,"DistanceM":32822.474212517},"2":{"idSite":3,"idType":4,"Type":"Site
>naturel","Description":"A voir absolument","Theme":"Côte de Nacre",
>Region":"Calvados","URI":"http://www.maregion.com","Lat":49.3,"Long":-
>0.352,"DistanceM":32822.474212517}}}

```

- ✓ Query for the list of the sites of a specified region and a specified type around the specified position: **q = SitesAroundMeByRegionAndType**
  - Required query parameters:
    - **region** = identifier of the region of the localized sites,
    - **type** = identifier of the type of searched site, the existing types are listed upon the query TypesOfSite
    - **lat** = current latitude,
    - **lon** = current longitude,
    - **dm** = max. great circle distance in meters (search radius) of the sites around the current position,
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter:
    - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
  - Return value: list of all the sites meeting the query criteria in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=SitesAroundMeByRegionAndType&email=paul.auchon@free.fr&code=553f2dac7baf6&region=1&lat=49.0&lon=-0.3&dm=35000&type=3>

*JSON result:*

```

{"SiteListe":[{"idSite":3,"idType":4,"Type":"Site naturel",
>Description":"A voir absolument",
>Theme":"Côte de Nacre",
>Region":"Calvados",
>URI":"http://www.maregion.com",
>Lat":49.3,"Long":-0.352,
>DistanceM":32822.474212517}]}

```

- ✓ Query for the list of the sites of a specified region and a specified theme around the specified position: **q = SitesAroundMeByRegionAndTheme**
  - Required query parameters:
    - **region** = identifier of the region of the localized sites,
    - **theme** = identifier of the theme of searched site, the existing types are listed upon the query ThemesOfSite
    - **lat** = current latitude,
    - **lon** = current longitude,
    - **dm** = max. great circle distance in meters (search radius) of the sites around the current position,
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter:
    - **pays** = country code for the language to be used in the reply message (FR, UK or IT; Default FR).
  - Return value: list of all the sites meeting the query criteria in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=SitesAroundMeByRegionAndTheme&email=paul.auchon@free.fr&code=553f2dac7baf6&region=1&lat=49.0&lon=-0.3&dm=35000&theme=1>

*JSON result:*

```
{"SiteListe":[{"idSite":3,"idType":4,"Type":"Site naturel","Description":"A voir absolument", "Theme":"Côte de Nacre","Region":"Calvados", "URI":"http://www.maregion.com", "Lat":49.3, "Long":-0.352, "DistanceM":32822.474212517}]}
```

- ✓ Query for the list of the URLs of the pictures associated to a specified site identifier: **q = Photos**
  - Required query parameters:
    - **idSite** = site identifier,
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter: none.
  - Return value: list of the URL of the pictures associated to a specified site in JSON format.

*Example of query:*

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Photos&idSite=1&email=paul.auchon@free.fr&code=553f2dac7baf6>

*JSON result:*

```
{"PhotoListe":[{"Nom":"Phare","Path":"http://82.233.223.249:8081/eras/images/Phare.jpg", "vignettePath":"http://82.233.223.249:8081/eras/images/Phare_TB.jpg", "Taille":36273}, {"Nom":"Desert", "Path":"http://82.233.223.249:8081/eras/images/Desert.jpg", "vignettePath":"http://82.233.223.249:8081/eras/images/Desert_TB.jpg", "Taille":64160}]}
```

- ✓ Query for the list of the URLs of the thumbnail associated to a specified site identifier: **q = Photos**
  - Required query parameters:
    - **idSite** = site identifier,
    - **email** = user email,
    - **code** = access code of the connected user.

- Optional query parameter: none.
- Return value: list of the URL of the thumbnail associated to a specified site in JSON format.

**Example of query:**

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Vignettes&idSite=1&email=paul.auchon@free.fr&code=553f2dac7baf6>

**JSON result:**

```
{ "PhotoListe": [{"idPhoto":72,"vignettePath":"http://82.233.223.249:8081/eras/images/arromanches_1TB.jpg"}, {"idPhoto":73,"vignettePath":"http://82.233.223.249:8081/eras/images/arromanches_2TB.jpg"}, {"idPhoto":74,"vignettePath":"http://82.233.223.249:8081/eras/images/arromanches_360_1TB.jpg"}, {"idPhoto":75,"vignettePath":"http://82.233.223.249:8081/eras/images/arromanches_360_2TB.jpg"}, {"idPhoto":83,"vignettePath":"http://82.233.223.249:8081/eras/images/Arromanches_360_3TB.JPG"}]}
```

- ✓ Query for the list of the URLs of the videos associated to a specified site identifier: **q = Videos**
  - Required query parameters:
    - **idSite** = site identifier,
    - **email** = user email,
    - **code** = access code of the connected user.
  - Optional query parameter: none.
  - Return value: list of the URL of the videos associated to a specified site in JSON format.

**Example of query:**

<http://serveurErasmus.institut-lemonnier.fr:8081/eras/index.php?q=Videos&idSite=1&email=paul.auchon@free.fr&code=553f2dac7baf6>

**JSON result:**

```
{ "VideoListe": [{"Nom":"Faune","Path":"http://82.233.223.249:8081/eras/videos/Faune.wmv","Taille":26246026}, {"Nom":"KitKat","Path":"http://82.233.223.249:8081/eras/videos/kitkat.3gp","Taille":2818508}]}
```

### 3.3. Environment constraints

Android API 14 Ice Cream Sandwich minimum.  
Development environment: Eclipse ADT and Android SDK or Android Studio.  
Languages: Java, XML.

### 3.4. Economic duress

Minimum footprint of the mobile Android application in energy usage and bandwidth.

### 3.5. Documents and technological tools at disposal

#### **Material resources**

Materials needed (available):

- ✓ development PCs,
- ✓ Operating server: Windows Server 2012R2 with WAMP Server and SQL Server 2012

- ✓ Nexus 7 Android mobile terminal.

### **Software resources**

Development environments (available):

- ✓ Visual Paradigm standard edition
- ✓ Visio 2013,
- ✓ Eclipse ADT and Android SDK or Android Studio.

### **Documentary resources**

Available manuals (Web Site):

- ✓ Slideshows and practicals Android M.M.Corporation:  
<http://sitemm.Michel-Marie.NET/siteMM/>

Available manuals (paper):

- ✓ Book "L'art du développement Android" edited by PEARSON.

### **Web server specifications**

IP addresses, Web service listener:

- ✓ private: **172.16.32.235**,
- ✓ public: **82.233.223.249** or **serveurErasmus.institut-lemonnier.fr**,
- ✓ listening port: **8081**.

## **4. Quality requirements to comply with**

### **4.1. Quality requirements for the product realization**

The HMI applications must be ergonomic and user-friendly (usable with one hand with the thumb). All the forms will be submitted to the customer for agreement.

Applications must be:

- ✓ robust, ensuring the monitoring of the validity of the data to another system,
- ✓ secured through the availability and continuity of treatment,
- ✓ structured, while fostering the modular development in order to facilitate the reuse of modules,
- ✓ serviceable, providing a great ease of localization and correction of residual errors, as well as the adding or removing of features.

### **4.2. Quality requirements for the development**

The object development will be based on the UML 2 modeling using the analysis tool "Visual Paradigm".

Coding under Eclipse ADT or Android Studio will be implemented in Java language and will use Android API 14 minimum.

### **4.3. Quality requirements for the documentation to be produced**

Quality requirements to comply with regarding documents:

- ✓ on their form: compliance with standards and standards of representation, workability, consistency, readability, maintainability,
- ✓ on their content: completeness, consistency and precision.

### **4.4. Quality requirements for the delivery**

The deliverables of the project are:

- ✓ the application projects,
- ✓ the installation package (.jar) of the application.

The project deliverable documentation must consist of:

- ✓ a technical dossier containing the files of specifications, preliminary design, detailed design
- ✓ separate technical annexes concerning the manufacturers documents and test worksheets.

The technical folder includes:

- ✓ the specification files, preliminary design, detailed design,
- ✓ the test worksheets,

The projects and application installer must be delivered on independent media supports.

### **4.5 Quality requirements for operating environment**

Some validation tests in situation should be carried out.

## 5. Distribution of functions or use cases

The project is to be undertaken by teams of 3 students, the following table indicates the distribution of the work to be performed by each student:

Student	Module or Use case
<u>Student 1 (E1) :</u>	<p>Android Nexus 7 tablet: <b>'LiveTourism' application</b></p> <ul style="list-style-type: none"><li>✓ Use case "<b>Configure Web service settings</b>".</li><li>✓ Use case "<b>Configure application settings</b>".</li><li>✓ Use case "<b>Locate the vehicle position on a map</b>".</li><li>✓ Use case "<b>Enable geolocation</b>".</li><li>✓ Use case "<b>Change the map type</b>".</li></ul> <p><b>Design, coding and testing of all the classes necessary to realize the previous use cases.</b></p>
<u>Student 2 (E2) :</u>	<p>Android Nexus 7 tablet: <b>'LiveTourism' application</b></p> <ul style="list-style-type: none"><li>✓ Use case "<b>Enable automatic display mode of the information on the nearest site</b>".</li><li>✓ Use case "<b>Display the sites on the map</b>".</li><li>✓ Use case "<b>Detect the presence or not of an active Internet connection</b>".</li><li>✓ Use case "<b>Enable guidance to a selected site</b>".</li></ul> <p><b>Design, coding and testing of all the classes necessary to realize the previous use cases.</b></p>
<u>Student 3 (E3) :</u>	<p>Android Nexus 7 tablet: <b>'LiveTourism' application</b></p> <ul style="list-style-type: none"><li>✓ Use case "<b>Display detailed information on a selected site</b>".</li><li>✓ Use case "<b>Display photos of the site</b>".</li><li>✓ Use case "<b>Read the videos of the site</b>".</li><li>✓ Use case "<b>Navigate to the Web site of the site</b>".</li></ul> <p><b>Design, coding and testing of all the classes necessary to realize the previous use cases.</b></p> <ul style="list-style-type: none"><li>✓ Management of the localization in French (default language), English and Italian.</li></ul>

## 6. Educational exploitation

### 6.1 Methods of projects assessment

- ✓ **An individual oral examination**

The oral test of 45 minutes will be divided into three parts:

- ✓ - 15 mns for the presentation of your work (content specified on next page),
  - ✓ - 15 mns of questions on issues relating to your presentation and your workbook,
  - ✓ - 15 mns for the validation of your work (demonstration, implementation: choice from a list of proposals).
- ✓ **A personal workbook about your work (see pages 14 and 15)**

Notation grid:

<b>IT project evaluation</b>		Duration	Note
Technical workbook	<ul style="list-style-type: none"> <li>- Ability to write a report on the work (plan, summary, paging, location of the appendices, vocabulary, clarity, correct spelling).</li> <li>- Ability to describe a personal work in a team effort.</li> </ul>		/10
Quality of realization (technical workbook)	<ul style="list-style-type: none"> <li>- Setting of the context.</li> <li>- Presentation of the work to be performed (analysis, formal description, etc.).</li> <li>- Design, production and unit tests.</li> <li>- Planning.</li> </ul>		/20
Oral presentation of the subject of the study (quality of presentation)	<ul style="list-style-type: none"> <li>- Plan, quality of the supports, time management, elocution, clarity of the presentation.</li> </ul>	20 mn	/10
Oral presentation of the subject of the study (technical quality)	<ul style="list-style-type: none"> <li>- Ability to synthesize the whole of one part of the project.</li> <li>- Ability to develop a critical point in the study.</li> <li>- Ability to conclude.</li> </ul>		/5
Questioning	<ul style="list-style-type: none"> <li>- Ability to answer (any question relating to the project) with relevance, precision, and accuracy.</li> </ul>	20 mn	/20
Presentation of the realization	<ul style="list-style-type: none"> <li>- Ability to validate upon request the different scenarios developed in the study.</li> <li>- Good usage of the tools (hardware and/or software).</li> </ul>	20 mn	/15
<b>TOTAL :</b>			<b>/80</b>

Remark: **Give the jury a copy of the proposed documents.**

### **Planning**

Beginning of the Mini project: October 5, 2015.

Rendering workbooks: no later than December 12, 2015.

Individual oral test: week from December 14 to 18 2015.



# Dossiers de projet IRIS

## Organisation minimale attendue

Le travail de rédaction relatif au projet IRIS doit donner lieu à :

- ✓ un dossier de présentation générale du projet, commun à l'équipe de projet,
- ✓ un dossier technique individuel par étudiant du projet.

### Dossier de présentation générale

Du point de vue de sa forme globale, le dossier de présentation générale du projet doit contenir :

- ✓ un sommaire paginé,
- ✓ un pied de page commun aux membres d'un même projet contenant :
  - le numéro de page,
  - l'intitulé du projet,
  - le rappel de l'aspect général de ce dossier.

Du point de vue des rubriques à présenter, le dossier de présentation générale doit contenir à minima :

- ✓ La présentation des objectifs du projet, ce projet s'adresse à qui, pour faire quoi, en utilisant quelles ressources matérielles et logicielles. Dans ce cadre au moins un synoptique est nécessaire pour une présentation claire des interactions entre les utilisateurs du projet et les différents matériels et logiciels (diagramme de déploiement).
- ✓ Une présentation de l'analyse générale du projet, à savoir :
  - quels sont les acteurs de ce projet et leur rôle,
  - quels sont les principaux cas d'utilisation : diagramme des cas d'utilisation de premier niveau,
  - quelles sont les séquences et scénarii associés aux précédents cas d'utilisations : diagrammes de séquence de premier niveau.
- ✓ La description des différentes tâches de développement à accomplir au sein de ce projet, ainsi que la répartition de ces tâches entre les étudiants du projet. Dans ce cadre également, il est conseillé de faire appel à un synoptique pour situer chaque étudiant dans le projet d'ensemble (diagrammes de déploiement et/ou de cas d'utilisation).
- ✓ Une conclusion générale sur l'état d'avancement du projet, son intégration entre les différentes parties et sa validation finale.

### Dossier technique individuel

Du point de vue de sa forme globale, le dossier technique du projet doit contenir :

- ✓ un sommaire paginé,
- ✓ un pied de page commun aux membres d'un même projet contenant :
  - le numéro de page,
  - l'intitulé du projet,
  - le nom de la partie traitée,
  - le nom du ou des auteur(s) de cette partie.
- ✓ des annexes vers lesquelles des références seront faites au sein du dossier, par exemple :
  - lors de la présentation d'un matériel spécifique, référence vers la documentation technique de celui-ci en annexe,
  - lors de la présentation d'une classe métier dans la partie conception, référence vers les fichiers de code source de celle-ci en annexe,
  - lors de la présentation d'un test de validation dans la partie des tests, référence vers les fichiers de code du programme de test correspondant en annexe.

**Remarque :** dans le cas où le volume du dossier technique avec ses annexes deviendrait trop important il est conseillé de séparer le dossier technique du dossier des annexes. **Les références du premier vers le second doivent cependant rester valides.**

Du point de vue des rubriques à présenter, le dossier technique individuel doit contenir à minima :

- ✓ Un rappel de la situation de l'étudiant dans le projet d'ensemble et des parties qu'il doit traiter.
- ✓ Une présentation des matériels, logiciels, protocoles spécifiques utilisés dans sa partie de projet, par exemple :
  - si l'étudiant doit développer un composant logiciel afin de faire communiquer un ordinateur PC avec une carte d'entrées/sorties interconnectés par un réseau de terrain de type CAN, il devra :
    - présenter brièvement le protocole CAN : ses spécificités, la constitution de la trame CAN, son mode d'adressage,
    - présenter le matériel correspondant à la carte CAN à intégrer à l'ordinateur PC : ses fonctionnalités, son type d'interfaçage matériel, son type d'interfaçage logiciel avec les outils de développement,
    - présenter le matériel correspondant à la carte d'entrées/sorties déportées via le bus CAN : ses fonctionnalités, etc...

*Dans le cadre de ce projet il faudra par exemple présenter le protocole HTTP avec paramètres Get, les objets Java permettant d'exécuter ces requêtes de façon synchrone ou asynchrone...*

- ✓ Une présentation de l'analyse de sa partie individuelle, à savoir :
    - quels sont les cas d'utilisation et séquences à traiter, en responsabilité totale ou partielle,
    - rappel des diagrammes de séquence de premier niveau correspondants et diagrammes de séquences détaillés (de second niveau) faisant apparaître clairement les objets des classes que l'étudiant devra développer,
    - diagrammes de classes avec les relations entre classes en faisant ressortir **clairement les classes dont le développement revient à l'étudiant auteur du dossier**,
    - diagramme de déploiement des modules logiciels à développer et de leurs dépendances,
    - diagramme de planning prévisionnel représentant la répartition prévue sur la durée totale du projet des différentes tâches à réaliser :
      - analyse,
      - conception de la classe x,
      - tests unitaires de la classe x,
      - intégration de la classe x avec l'application de l'étudiant y,
      - tests d'intégration avec l'étudiant y,
      - tests de validation avec l'ensemble des étudiants du projet.
  - ✓ La conception détaillée et les tests individuels de chacune des parties à développer, par exemple :
    - conception de la classe client CAN en justifiant à partir d'un diagramme de la classe :
      - du choix des attributs et de leur visibilité,
      - du choix des propriétés,
      - du choix des méthodes, paramètres passés et retournés,
      - des types de relations utilisées entre cette classe et les autres classes du projet, cardinalité.
    - méthodes de test utilisées pour valider individuellement cette classe :
      - quel type de test, pour valider quoi,
      - conditions du test, avec quels matériels,
      - résultats attendus dans les conditions précitées,
      - résultats obtenus et modifications éventuelles apportées.
- Rappel : le code source de la classe et des programmes de test est à placer dans la partie des annexes mais des références paginées à ceux-ci doivent être présentes.*
- ✓ L'intégration avec les autres étudiants du projet et les tests d'intégration associés :
    - intégration de quelle partie, avec qui,
    - conditions du test, avec quels matériels,
    - résultats attendus dans les conditions précitées,
    - résultats obtenus et modifications éventuelles apportées.

✓ Les tests d'ensemble.

✓ Le planning réalisé (diagramme de Gantt) représentant la répartition effective sur la durée totale du projet avec le comparatif au planning prévisionnel et la justification des écarts.

✓ Une conclusion individuelle sur le travail réalisé, les modifications à apporter, les points positifs et négatifs, etc...

	Etudiant 1	Etudiant 2	Etudiant 3	Professeur responsable
Équipe 1				Mme TBER
Équipe 2				M GOUVILLE
Équipe 3				M NEVEU
Équipe 4				M MARIE
Équipe 5				M NEVEU/M GOUVILLE
Équipe 6				M MARIE/ Mme TBER