forecAsting System for urban heaT Island effect

“Implementation of a forecAsting System for urban heaT Island effect for the development of urban adaptation strategies” (LIFE ASTI)

Action A.1 **Preliminary design of the pilot operational UHI forecasting systems**

Thessaloniki December 2018

The project Implementation of a forecAsting System for urban heat Island effect for the development of urban adaptation strategies - LIFE ASTI has received funding from the LIFE Programme of the European Union”.

www.lifeasti.eu

Document Information			
Grant agreement number			LIFE17 CCA/GR/OOO108
Project acronym			LIFE ASTI
Project full title			Implementation of a forecAsting System for urban heaT Island effect for the development of urban adaptation strategies
Project's website			Under construction
Project instrument			EUROPEAN COMMISSION - Executive Agency for Small and Medium-sized Enterprises
Project thematic priority			Climate Change Adaptation
Deliverable type			Report
Contractual date of delivery			31/12/2018
Actual date of delivery			31/12/2018
Deliverable title			Architectural and Functional Report
Action			A.1 Preliminary desing of the pilot operational UHI forecasting systems
Authors			Panagiotis Symeonidis, Dimitris Karakoutas, Elena Athanasopoulou, Simeon Taskaris, Pantelis Mitropoulos, Nikolaos Kallitsis
Version History			
Issue Date	Version	Author	Partner
31-12-2018	V.1	Panagiotis Symeonidis, Dimitris Karakoutas, Elena Athanasopoulou, Simeon Taskaris, Pantelis Mitropoulos, Nikolaos Kallitsis	GET

Table of contents

I Architectural and Functional Report

The LIFE ASTI project focuses on addressing the impact of Urban Heat Island (UHI) effect on human mortality by developing and evaluating a pilot system of numerical models that will lead to the short-term forecasting and future projection of the UHI phenomenon in two Mediterranean cities: Thessaloniki (Greece) and Rome (Italy).

The phenomenon of UHI has an impact on human health, which is becoming more intense as the duration of the heat wave episodes is expected to increase due to climate change. The spread of urban areas has become alarming in recent years: almost 73% of Europe's population lives in cities, a rate which is expected to reach 80% by 2050. Extensive urbanization is triggering significant changes to the composition of the atmosphere and the soil, which result in the modification of the thermal climate and the temperature rise in urban areas, compared to neighboring nonurban ones.

The modeling system which will be developed in the framework of the LIFE ASTI project, will produce high-quality forecasting products, such as bioclimatic indicators and heating and cooling degree days, which assess the energy needs of buildings. In addition, it will guide the Heat Health Warning System to be implemented in both cities and will aim at informing the competent authorities, the general population and the scientific community.

This report presents the architecture and the functional requirements of the LIFE ASTI system platform. The LIFE ASTI system platform is a collection of IT software and applications that will provide access to the information produced in the project.

The platform is designed to follow the n-tier architecture to provide maintainability, scalability, flexibility, and availability. It consists of:

- **Central Database**
  It will store all the data produced during the project (forecasting models results as well as meteorological station data and other supporting datasets). It will be developed using the most advanced open-source Relational DataBase Management System with spatial data support (PostgreSQL with PostGIS).
- **Services**
  Access to data and information derived from project will be performed using web services. Since the project generated spatial data several of the services are related to spatial data access and are based on standards defined by the OGC: WMS, WFS, WCS. In addition, custom services will automate the ETL process as well as other functionalities of the provided applications.
- **Applications**
  For the dissemination of the LIFE ASTI project results a web and a mobile application will be developed. The web user interface will be developed based on the GET SDI PORTAL. GET SDI PORTAL is a reach web GIS client offering advanced visualization and analysis capabilities. Concerning the mobile app, state-of-the-art technologies like ionic, openlayers, angularJS and Apache Cordoba will be applied so that the mobile app could be provided as a native app for both Android and iOS devices.

In this report:

- **Chapter 1: System Design Concept**, presents the system architecture (General Architecture, Physical Architecture, Logical Architecture) and the data flows,
- **Chapter 2: Software components**, presents the software tools and libraries that will be used for the system development,

- **Chapter 3: Non-Functional Requirements**, presents the basic rules that the system should comply, while
- **Chapter 4: Functional Requirements**, presents the user requirements for both web and mobile applications. Mockups of the user interfaces are presented and the main use cases are also identified.

i. System design concept

a. System architecture and data flow

The implementation of the proposed system will be based on an **open**, **multi-tier web** architecture. This will facilitate the development and future escalation of the system, while it will allow the interconnection with other systems, as open protocols and communication standards will be used in line with the interoperability framework (e.g. WSDL, SOAP, XML).

An important role in the expandability and scalability will also have the utilization of virtual machine technology and virtualization environment in general.

In order for the system to serve the purpose for which it will be created, it will have the following characteristics:

- Ease of use, operation, maintenance
- Flexibility in the adoption and integration of new technologies
- Use of open standards
- Scalability
- Performance
- Availability
- Utilization of existing resources (human resources and know-how, software / hardware)

More specifically, the requirements that will be overall satisfied by the architecture are as follows:

- It is based on high-level integration of state-of-the-art **open source software** and **open standards**.
- Easy to **scale** and adapt to the application requirements.
- Comply with the multi-level **security** rules necessary for the safe operation of modern web applications and their protection against malicious practices.
- Access to the system is provided to unlimited number of users.
- Supports role-based security.
- Remote access is supported.
- Data and content will be stored in appropriate **database** formats. The applications will access the data through **services**.
- Provides multiple applications to access the data customized to the users (web, mobile).
- Compatibility with many different forms of geospatial data
- Link / import data from external data sources
- The web application will be compatible with major web browsers, such as Microsoft Edge, Mozilla Firefox, and Google Chrome.

b. General architecture

The following figure presents the overall system architecture.

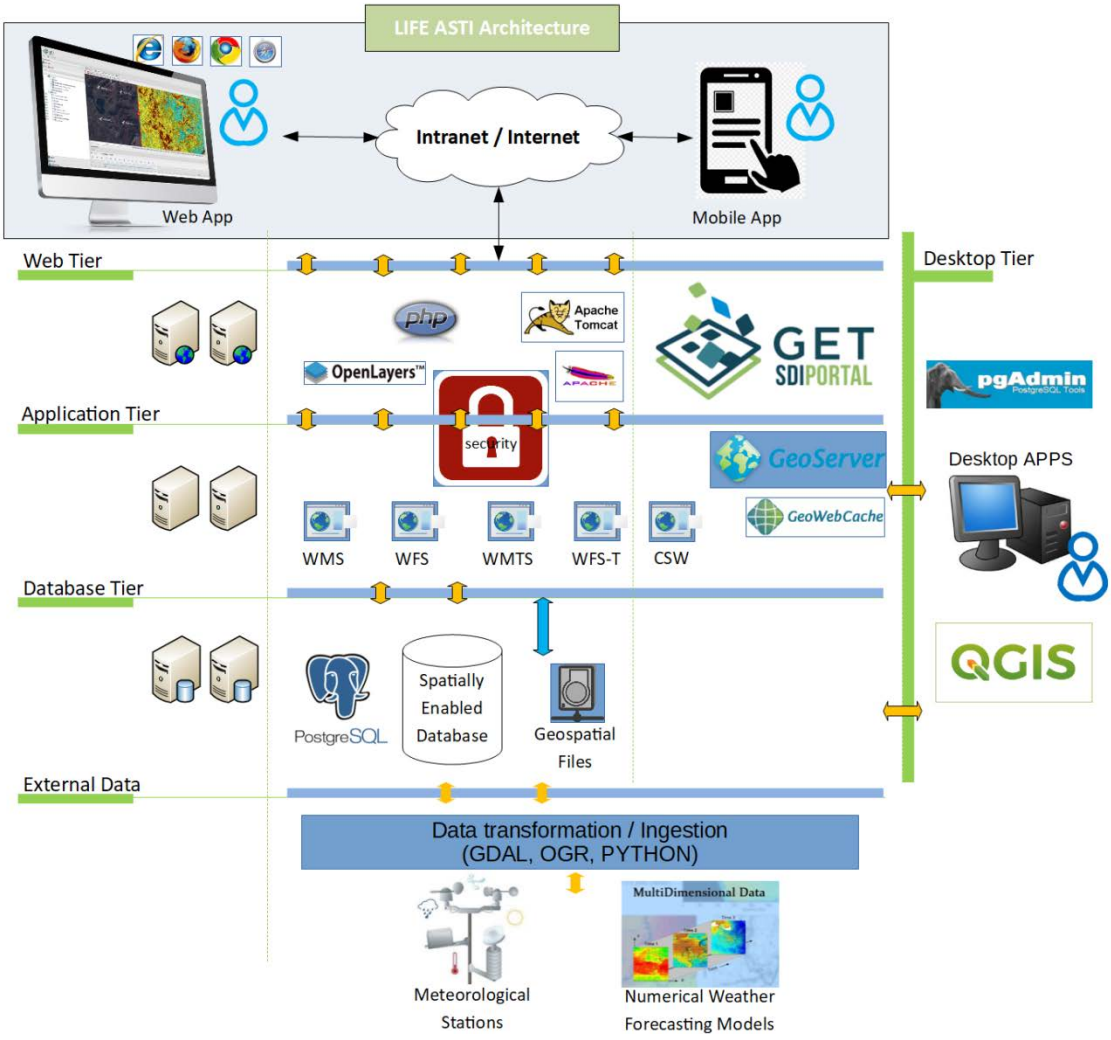
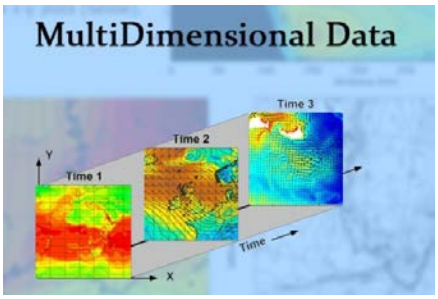


Figure 1. System architecture

The foreseen data flow is presented below:

Meteorological Station



Numerical Weather Forecasting Models

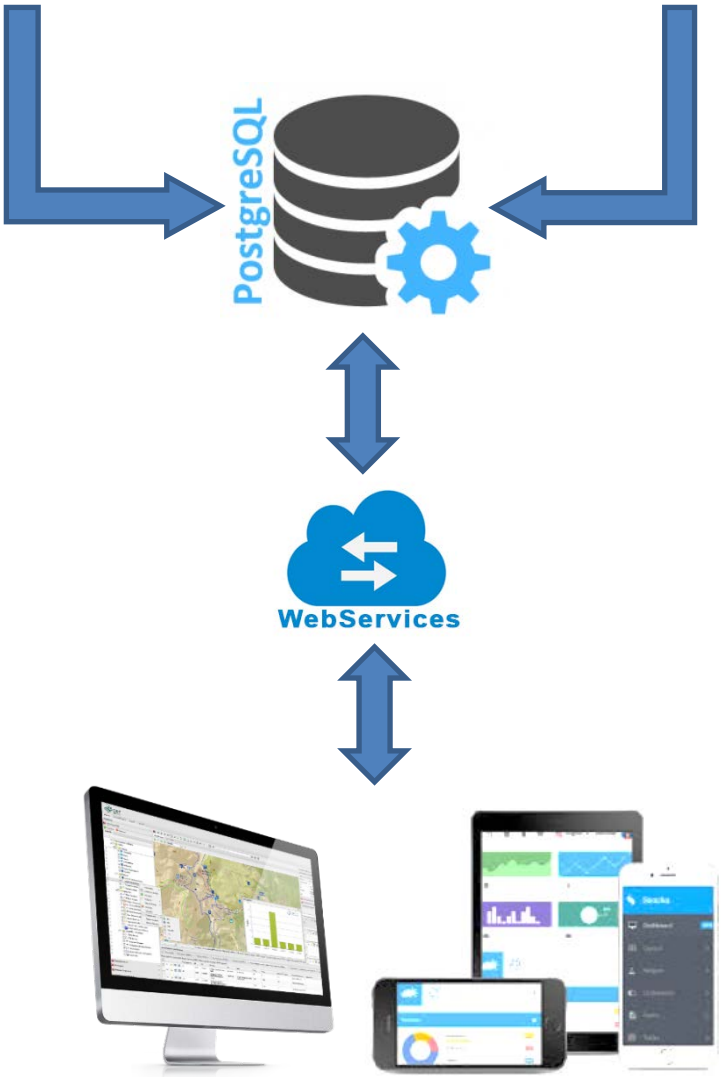


Figure 2. Data flow

c. Physical architecture

The web platform will be installed in a **Virtual Machine** (VM) hosted in a Cloud Server.

The basic requirements of the hosting machine and the VM are presented in the table below.

	Host Server	VM
CPU	8	2
Memory	64 GB	8 – 10 GB
Disk	2 TB	200 – 500 GB
Network	1 Gbit	1 Gbit

Table 1. Hardware requierements

Among the advantages of the VM deployment are:

- **Reduced hardware cost**. This estimated around 40% by maximizing the existing hardware capabilities.
- **Faster system provisioning and deployment**. VM allows to easily clone an existing VM and set different starting points, reducing dramatically the effort needed for these tasks.
- **Improved disaster recovery**. VMs can be easily moved from one server to another. Thus, back up can be done quickly and more effectively.
- **Energy cost saving**. Energy consumption is directly related to the number of physical servers. Thus, VMs can reduce energy cost up to 80%.
- **Increased productivity**, due to less time spend for maintenance and deployment.

d. Logical architecture

The identification of several layers is based on clustering the business operations and needs of the LIFE ASTI project. The layers identified are:

- **External data Layer**: this layer includes the raw data produced by the meteorological forecast models and the IoT sensors
- **Management Layer**: this layer includes the infrastructure for data storing, management and editing
- **Service Layer**: this layer encapsulates the LIFE ASTI business logic operations, by developing and deploying the respective software components and workflows
- **Presentation Layer:** this layer implements the system's interface with its users
- **Back-office application Layer**: this layer includes software programs (desktop) that are used from data administrators and expert users for administrative purposes.

To implement the logical architecture, the proposed solution uses the **n-tier** model. The solution consists of the following tiers:

- External data tier
- Database tier
- Application tier
- Web tier
- Desktop tier

A short description for each tier is provided in the next sections.

d.1. External data tier

This tier provides access to external raw data provided by models or sensors in the form of services or data. Custom ETL (Extract – Transform – Load) modules will be developed in order to facilitate the ingestion of these data to the central database. These modules will use open source applications and libraries like GDAL and OGR, which will be integrated in python.

d.2. Database tier

This tier lies down at the lower level in the physical and logical design. This is where all the information (spatial, attribute data, documents) is stored. It includes:

- (geo)spatial database(s) hosting the data,
- the Enterprise Open Source Object-Relational Spatial Database Management System (O-RDBMS) that will be used,
- data directories organized in a coherent way

Direct access to this tier shall be available only for internal system administrator(s), while rest of the users will access the database through the Application Tier.

d.3. Application tier

This is an intermediate tier that will practically implement the required server-side functionality. Application tier will be accessed through the web tier. Additionally, the layer includes security issues, user management etc. This tier includes:

- Open Source Map Server Software
- Open Source Catalogue Server Software
- Custom Server-side applications, scripts and procedures implementing the required functionality

d.3.1. Spatial network services

This system will be developed to consume spatial data services, used to disseminate the spatial data through the Web. Network Services connect to the database and serve data, through the Network, to end users. Services will be developed by using the most interoperable standards in the geospatial domain, namely Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) and Catalogue Service for the Web (CSW).

d.3.1.1. View service

The data dissemination through the View Services is based on the OGC's Web Map Service (WMS) standard, version 1.3.0. The WMS standard is a standardized method for exchanging georeferenced images (maps) using the HTTP. This standard is supported not only by web application, but also by desktop GIS applications (both commercial and open source), offering increased interoperability in viewing geospatial data.

By default, View Services make possible, as a minimum, to display, navigate, zoom in/out, pan, or overlay viewable spatial data sets and to display legend information and any relevant content of metadata.

d.3.1.2. Download service

The Download Services, for vector data, are based on the Open Geospatial Consortium's (OGC) Web Feature Service (WFS) standard. The WFS standard provides the interface for sending requests for spatial data through the

web. The difference with the View Services is that through the Download Services the user gains access to “real” data and not to “images” of the data. The Download Services make it possible to download copies of spatial data sets, or parts of such sets, to be downloaded and, where practicable, accessed directly.

The Download Services may regard the access to predefined data sets and/or parts of such sets, but also to direct access download services with the ability to perform queries. The former approach is recommended for data sets with more “static” content and of large relative area (e.g. a country), while the latter for more dynamically changing over time data sets, serving specific needs and applications.

For raster datasets the Web Coverage Service (WCS) will be provided in order to define web-based access to coverages which are digital geospatial information representing space/time varying phenomena.

d.3.1.3. Catalogue (discovery) service

The Discovery Services are based on the Open Geospatial Consortium’s (OGC) Catalogue Service for the Web (CSW) standard. These services make it possible to search in metadata catalogues based on criteria like key-words, area of interest, data themes, time extent, creation date, scale etc.

d.3.2. Data transformation

A custom service will be developed in order to automate the process of data ingestion from the models and the sensors to the central database.

For model data an automation script is expected to ingest the multidimensional forecast data to the central database following these steps:

1. Export model binary outputs to netCDF format
2. Slice netCDF files in space / time and for each parameter
3. Perform index calculations
4. Export results to GeoTiff format
5. Create / update an image mosaic store to GIS Server software
6. Update the OGC Services for data access

For the meteorological station data, the ingestion workflow will follow these steps:

1. Creation of a data import script from ascii / csv files to database
2. Setup a listener to a folder where the station data files will be delivered
3. The import script will be automatically executed every time a new file will be available

For this workflow we assume that meteorological measurements will be communicated over the internet as ascii delimited files. In case the data are transferred with different method or with another API the data import script will be modified accordingly.

d.4. Web tier

This is the higher level accessible both through intranet and internet, where the users will have access to the data and operations. The applications will operate in a web browser environment and in mobile environment.

d.5. Desktop tier

This layer provides to the administration users the tools to perform advanced data management functionalities. Using desktop tools like pgadmin, for database management, and QGIS, for geospatial data management (both at DB and GIS Server) the users are able to design / develop the database, create spatial indexes, setup backup procedures etc. that are necessary for the optimal system performance.

e. Security

The system will support three user categories:

- Public users
- Registered users
- Administrators

Public users will have access to all public data produced in the LIFE ASTI project.

The registered users will have access to additional functionalities in the applications (like Favorite locations in the Mobile app) as well as personalized information. For example, elderly people will receive special instructions with precaution actions in case of an extreme Heatwave event.

Administrators will have direct access to the data and the related software through dedicated web or desktop applications. These applications will allow them to perform data management tasks (such as database optimization, database backup etc.) necessary for the optimal operation of the system.

The system will provide a single sign-on security. Users will be able to register in the web or the mobile app. Upon the successful registration, no additional registration will be required (in the other app).

Personal data will be stored in the Registered Users Database. All information will be protected according to the EU and National regulations. Thus, personal information and files will be encrypted as well as the users' database backup which will be taken regularly to prevent accidental data loss.

ii. Software components

a. Core components

a.1. Postgresql

PostgreSQL (latest release 11.1, 8/11/2018), is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. It is available in several OS like macOS, Microsoft Windows and Linux.

PostgreSQL is ACID-compliant and transactional. PostgreSQL has updatable views and materialized views, triggers, foreign keys; supports functions and stored procedures, and other expandability.

PostgreSQL is free and open-source, released under the terms of the PostgreSQL License, a permissive software license.

The main features of the software are:

- Strong scalability, high availability, security and reliability to facilitate the implementation and management of centralized systems that are able to respond effectively to the needs of service large numbers of users and manage large volumes of data.

- Support the most popular operating systems (Windows, Linux, MacOS).
- Localization support
- Compatibility with ANSI-SQL standard
- Unicode/UTF 8 support
- Views/ Queries support
- ODBC interface support
- Database Schema creation and data management using SQL
- Multiple Database Schemas implementation support
- The software can be managed using GUI, via intranet / internet without requiring physical access to the server that is installed.
- Embedded data processing and analysis functions.
- Trigger, cast support
- Authorized data access, even at record level
- Role-User Policy
- Encryption Capabilities
- New data type definition support
- "Full Text search" support
- Indexing of complex objects support
- Auto maintenance and optimization database tools.
- Support various data types
- Referential Integrity Rules support
- Indexing algorithms, such as B-tree, R-tree Support
- Embedded languages for writing functions
- Using tables as data type support
- Table inheritance support
- Transaction control mechanisms
- Performance control mechanisms
- Communication with commercial, open source Desktop GIS software
- Backup and restore capabilities
- Programming interfaces C/C++, Java, .Net, Perl, Python, Ruby SDK's
- Unlimited database size support
- Unlimited number of records per table support
- Geospatial Data support with or without additional software (extension, add-on)
- Spatial Indexing support
- Embedded geometry creation, processing and extraction functions for geospatial data

PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium.

The main features of the software are:

- Geometry types for Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultiPolygons and GeometryCollections.
- Spatial predicates for determining the interactions of geometries using the 3x3 DE-9IM (provided by the GEOS software library).
- Spatial operators for determining geospatial measurements like area, distance, length and perimeter.

- Spatial operators for determining geospatial set operations, like union, difference, symmetric difference and buffers (provided by GEOS).
- R-tree-over-GiST (Generalized Search Tree) spatial indexes for high speed spatial querying.
- Index selectivity support, to provide high performance query plans for mixed spatial/non-spatial queries.
- For raster data, PostGIS WKT Raster (now integrated into PostGIS 2.0+ and renamed PostGIS Raster).

The PostGIS implementation is based on "light-weight" geometries and indexes optimized to reduce disk and memory footprint. Using light-weight geometries helps servers increase the amount of data migrated up from physical disk storage into RAM, improving query performance substantially.

PostgreSQL will be used for the development of the central database of the system.

a.2. Geoserver

GEOSERVER is a Java-based software server that allows users to view and edit geospatial data. Using open standards set forth by the Open Geospatial Consortium (OGC), GeoServer allows for great flexibility in map creation and data sharing. GeoServer is a OGC compliant implementation of a number of open standards such as Web Feature Service (WFS), Web Map Service (WMS), and Web Coverage Service (WCS). Additional formats and publication options are available including Web Map Tile Service (WMTS) and extensions for Catalogue Service (CSW) and Web Processing Service (WPS). Geoserver might be used in order to provide serviced based access, using OGC standards like WMS, WFS, and WCS.

The main features of GEOSERVER are:

- Implements OGC services including Web Map Service (WMS 1.1.1 and 1.3.0), Web Feature Service (WFS 1.0.0 and 1.1.0), WFS-Transactional (WFS-T 1.0.0), and Web Coverage Service (WCS 1.0.0 and 1.1.1)
- Java J2EE application, works with Jetty, Tomcat, WebLogic, WebSphere, JBoss.
- Supports the n-tier architecture.
- Support the most popular operating systems (Windows, Linux).
- Support data publishing from the most popular (commercial and FOSS) RDBMSs (namely like MS SQL PostgreSQL/PostGIS, MySQL, Oracle Spatial, ESRI ArcSDE).
- Support file-based geospatial data formats (e.g. ESRI Shapefile, Vector Product Format, MapInfo MIF/MID, TIFF, GeoTIFF, BigTIFF, GTOPO30, ECW, MrSID, and JPEG2000).
- Support in JPEG and PNG formats, with SLD support integrated.
- Offers support for reference systems specified in the INSPIRE Directive and dynamic transformations.
- Support for data export in various formats such as ESRI Shapefile, KML, GML, PDF, and JPEG.
- Support the geospatial data dissemination in JPEG, GIF, PNG, PDF, SVG, KML, and GeoRSS formats.
- Image antialiasing.
- Caching and Tiling
- Fully-featured and intuitive web administration interface with REST API for programmatic control
- Full map styling support with Styled Layer Descriptor (SLD), including text-based and graphical editor
- User- and role-based security subsystem based on Spring Security

GEOSERVER will be used for the provision of the spatial data access services, WMS, WFS, WCS and CSW (through the CSW extension).

Multidimensional data will be processed using the GEOSERVER **Image Mosaic** data store which supports both time and elevation dimension as well as additional custom dimensions.

a.3. GET SDI portal

GET SDI Portal, is a web mapping application providing a comprehensive platform for viewing, downloading, analyzing, querying, editing and styling data originating from multiple geospatial sources. It constitutes a simple and ready to deploy solution for any organization who wants to setup a Geoportal based on ISO/OGC Standards.

GET SDI Portal has been developed by GET in order to support the implementation of the Infrastructure for Spatial Information for Europe (INSPIRE) Directive (2007/2/EC), as well as to serve needs of organizations targeting to share their geospatial resources. Its modular architecture allows implementing widgets for the realization of specific functionalities, integrated smoothly in a configurable and easy to use web application.

At its core, GET SDI Portal is based on open source tools such as **Openlayers**, **ExtJS**, **GeoExt** and **Proj4js.**

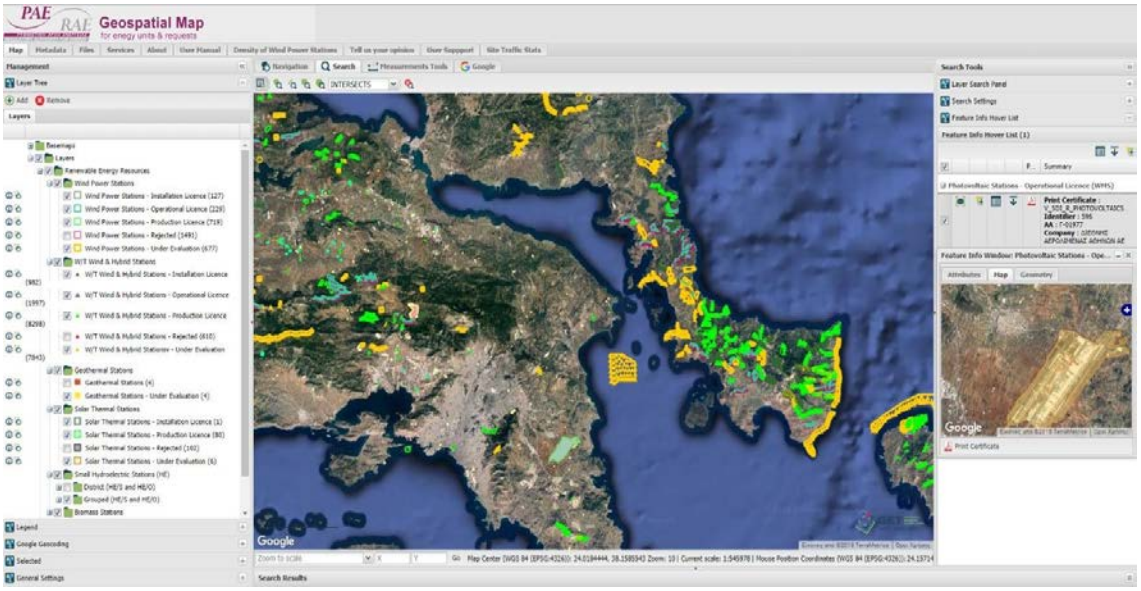


Figure 3. The GET SDI portal main user interface

The main features of GET SDI Portal are:

- User friendly GUI.
- Operates under any of the well-known web browsers and at least on the followings: Internet Explorer, Firefox , Google Chrome ,Safari.
- It provides an interactive map control with the following functionalities: Layer management (Add/ Remove/ Reorder)., Pan, Zoom in/out, Zoom by area, Measure Distance, Measure Area, Get/Set Coordinates, Change base layer, Show scale info, Coordinate Display
- It can consume any third party INPSIRE/OGC compliant service using the WMS, WFS, and CSW standards.
- View / overlay multiple layers from different WMS sources on the map.
- Add/remove third party WMS (versions 1.1.1 to 1.3.0) service.
- Add/remove legend entries from the added WMS services.
- Reorder layers.
- Hide/show added layers.
- Show the legend icon based on the used SLD of the layer.

- Change the opacity of each layer on the map.
- List of the WMS services that the user added or are predefined.
- Zoom to max extent of the layer.
- Provide layer information to the user.
- Error message where the service is unavailable.
- Add/remove third party WFS (v.1.1.0, 2.0) service.
- Add/remove legends from the added WFS services.
- Change the order that the layers are shown on the map.
- Hide/show the added layers.
- List of the WFS services that the user added or are predefined.
- Zoom to max extent of the layer.
- Provide layer information to the user
- View multiple layers from different WFS sources on the map.
- Should provide error message where the service is unavailable
- Support searching is CSW Catalogues
- Metadata Search criteria as:Resource Title,Resource Abstract,Resource type,Unique Resource Identifier,Temporal Reference.
- Discovering resources based on the geographic location of the resource, the spatial operator 'Intersects' is supported
- Add/remove third party CSW (v2.0.2) service.
- Provides results from CSW search queries in list form.
- Provides simple or advanced search on different GUIs.
- Provides simple search for service queries.
- For each returned record, the user is able to view the metadata info of the record. It also visualizes on the map the bounding boxes that correspond to the chosen record.
- Provide error messages where the service is unavailable.
- Gives the ability to the user to select the protocol to be used to perform queries (ISO AP CSW, DC, INSPIRE).
- Search results (returned record) is a WMS Service, it provides user with the ability to add it automatically to the WMS list registered services of the module.
- The result list from the query is dynamic using - Bounding box of the current map, Bounding box of a drawn rectangle from the user
- Supports reference systems and grids set by the INSPIRE Directive (ETRS89, LAEA etc.).
- Includes map printing component, allowing printing in PDF format where title, scale bar, and legend are displayed.

GET SDI Portal will be used as the base for the development of the web application (for expert users with advanced capabilities / functionalities).

a.4. THREDDS data server

The **THREDDS Data Server** (TDS) is a web server that provides metadata and data access for scientific datasets, using OPeNDAP, OGC WMS and WCS, HTTP, and other remote data access protocols. The TDS is developed and supported by Unidata, a division of the University Corporation for Atmospheric Research (UCAR), and is sponsored by the National Science Foundation.

Some of the technology in the TDS:

- THREDDS Dataset Inventory Catalogs are used to provide virtual directories of available data and their associated metadata. These catalogs can be generated dynamically or statically.

- The Netcdf-Java/CDM library reads NetCDF, OpenDAP, and HDF5 datasets, as well as other binary formats such as GRIB and NEXRAD into a Common Data Model (CDM), essentially an (extended) netCDF view of the data. Datasets that can be read through the Netcdf-Java library are called CDM datasets.
- TDS can use the NetCDF Markup Language (NcML) to modify and create virtual aggregations of CDM datasets.
- Provides OPeNDAP access to any CDM dataset.
- Provides bulk file access through the HTTP protocol.
- Provides data access through the OpenGIS Consortium (OGC) Web Coverage Service (WCS) protocol, for any "gridded" dataset whose coordinate system information is complete.
- Provides data access through the OpenGIS Consortium (OGC) Web Map Service
- (WMS) protocol, for any "gridded" dataset whose coordinate system information is complete.
- The integrated ncISO server provides automated metadata analysis and ISO metadata generation.
- The integrated NetCDF Subset Service allows subsetting certain CDM datasets in coordinate space, using a REST API. Gridded data subsets can be returned in CF-compliant netCDF-3 or netCDF-4. Point data subsets can be returned in CSV, XML, or CF-DSG netCDF files.

TDS will be examined to be used in conjunction with GEOSERVER in order to provide OPeNDAP access to CDM datasets (including virtual aggregations) as well access using OGC WMS and WCS standards.

b. Data processing

b.1. netCDFr

NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for C, FORTRAN, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

The netCDF libraries will be used to access netCDF files using different programming languages (C, Java, python etc).

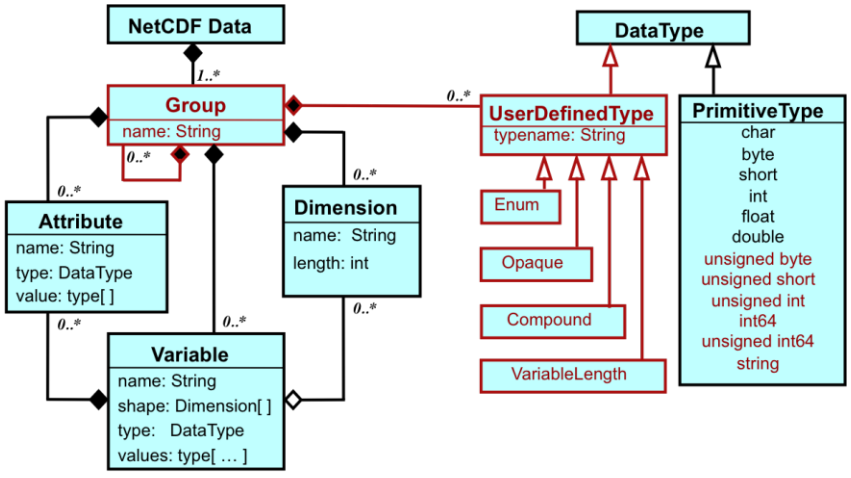


Figure 4. The netCDF data model

The output of the meteorological forecasting models will be provided in netCDF format. The ETL modules will use this format as the input format while the output format will depend on the GIS server that will be used. GeoTiff or even netCDF (sliced in space and/or dimensions) are possible alternatives.

b.2. numPY

Numpy is a powerful and robust Python library for defining handling and executing functions on large multi-dimensional arrays. It also provides the means to define such arrays as well as a large collection of high-level mathematical functions that operate on these arrays. Several other scientific tools and Python libraries and distributions use the NumPy library. In the context of the current project, NumPy shall be used for low-level data manipulation and data extraction from information that is formatted in 1D, 2D and 3D arrays. Tools for extracting simple statistics from array data can also be developed.

b.3. GDAL

The **Geospatial Data Abstraction Library** (GDAL) is a computer software library for reading and writing raster and vector geospatial data formats. It is released under the permissive X/MIT style free software license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It may also be built with a variety of useful command line interface utilities for data translation and processing. Projections and transformations are supported by the PROJ.4 library. The related OGR library (OGR Simple Features Library), which is part of the GDAL source tree, provides a similar ability for simple features vector graphics data. Currently GDAL provides at least partial support for 154 raster and 93 vector geospatial data formats including geoTIFF, netCDF, HDF4&5, JPEG2000, RASDAMAN, Sentinel 1 SAR SAFE, and Sentinel 2. GDAL library, together with other geospatial processing libraries, will be used for raster data transformation and processing.

GDAL will be used in the ETL tools that will be developed for the ingestion of the meteorological multi-dimensional models to the database and to the GIS Server.

b.4. netCDF4-python

netcdf4-python is a Python interface to the netCDF C library. netCDF version 4 has many features not found in earlier versions of the library and is implemented on top of HDF5. This module can read and write files in both the new netCDF 4 and the old netCDF 3 format, and can create files that are readable by HDF5 clients. It will be used in order to access, process and analyze netCDF datasets from Python environment.

c. Data processing

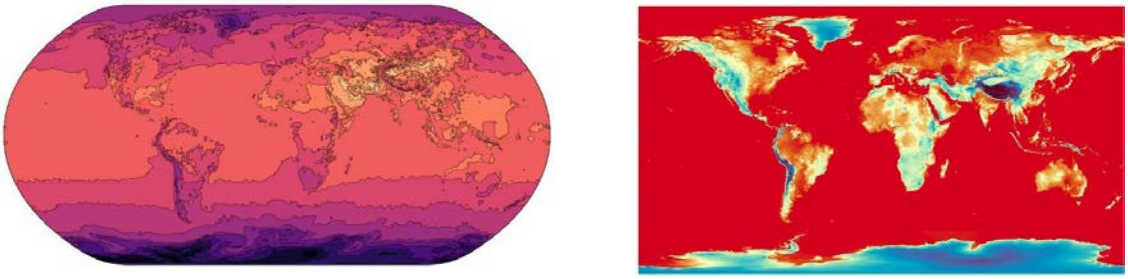
c.1. Openlayers

OpenLayers is an open source (provided under the 2-clause BSD License) JavaScript library for displaying map data in web browsers. It provides an API for building rich web-based geographic applications similar to Google Maps and Bing Maps. OpenLayers supports several geospatial standards like GeoRSS, KML, GML, GeoJSON, WMS and WFS. The first release of OpenLayers was in 2006 while the latest stable release was 4.6.5 (20/3/2018). OpenLayers will be used for the visualization of geospatial data in the web.

Figure 5. Example of OpenLayers use for the visualization of multidimentional data

c.2. D3.JS

D3.js is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. D3 uses web standards like HTML, SVG, and CSS. D3's emphasis on web standards allows web applications to get the most from the capabilities of modern browsers, combining powerful visualization components and a data-driven approach to DOM manipulation. The development of D3 started in 2011 and currently the latest version is 5.1.0 (17/4/2018). The software is open source with a BSD license. D3 will be used for the creation of web data visualizations.



(a) (b)

Figure 6. Example D3.js visualizations: (a) Contours created from a geotiff raster file, (b) Raster visualization using custom color scale

c.3. CESIUMJS

CesiumJS is an open-source JavaScript library for creating 3D globes and 2D maps in a web browser without a plugin. It uses WebGL for hardware-

accelerated graphics, and is cross-platform, cross-browser, and tuned for dynamic-data visualization. It creates 3D globes and maps from static and time dynamic content, with the best possible performance, precision, visual quality, platform support, community, and ease of use. Cesium adopts several open formats and standards like: 3D Tiles, glTF, quantized-mesh, and CZML. It is licensed under the Apache 2.0 license and it is free both for commercial and non-commercial use. CesiumJS will be used in order to provide 3D visualization (globe) to datasets.

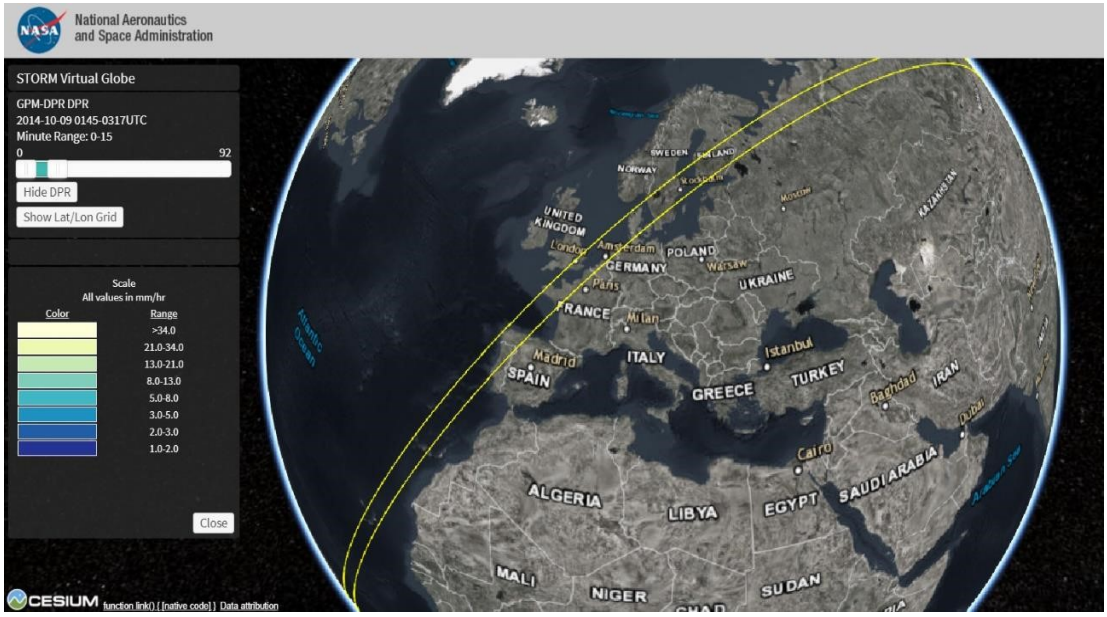


Figure 7. Example of 3D globe data visualization using CesiumJS

d. Data processing

d.1. AngularJS

AngularJS is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, a framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular application platform.

The AngularJS framework works by first reading the Hypertext Markup Language (HTML) page, which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code or retrieved from static or dynamic JSON resources.

d.2. Apache Cordova

Apache Cordova (formerly PhoneGap) is a mobile application development framework originally created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as PhoneGap, and later released an open source version of the software called Apache

Cordova. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone. It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device. It extends the features of HTML and JavaScript to work with the device. The resulting applications are hybrid, meaning that they are neither truly native mobile application (because all layout rendering is done via Web views instead of the platform's native UI framework) nor purely Web-based (because they are not just Web apps, but are packaged as apps for distribution and have access to native device APIs).

d.3. Ionic

Ionic is a complete open-source SDK for hybrid mobile app development. The original version was released in 2013 and built on top of AngularJS and Apache Cordova. The more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova.

iii. Non-functional requirements

The system shall comply with the following Non-Functional Requirements (NFR):

- Performance requirements
  The system should be optimized regarding the hardware use.
  It should also have quick response to user requests without delays that might affect the accepted usability.
- Operating constraints
  For the web application no add-on software should be requested. The application should be available from a web browser.
- Platform constraints
  The web application should be platform independent. Thus, it should be available regardless of the OS of the user.
  The mobile app should be provided as a native app for Android and iOS.
- Data integrity
  Data should be accurate stored in the database. No modification of the data is allowed during the ETL procedures. The database should also ensure data integrity by setting the necessary indexes and quality rules.
- Modifiability
  The application should be easily modified to adopt to changes and new requirements.
- Portability
  The application should be easily transferred to new Server or to the Cloud
- Availability
  The application should have high level of availability.
- Security
  Data should be protected from un-authorized users.
- Usability
  The applications (web and mobile) should provide UIs that facilitate the user interaction. It should allow users to retrieve the necessary information without using the manual.
- Legal
  Personal data should be protected according to European and National law.

iv. Functional requirements

a. Web application

The web application will target the public in order to provide information about the UHI in the pilot areas.

A first page will introduce the people to the pilots and the available services. From this the users will be able to access:

- A **dashboard**, for each pilot area, in which summarized information and alerts from the forecast model will be presented. The information will be presented using easy to understand infographics and it will target to non-expert users.
- For more advanced users a **cartographic application**. In this application all spatiotemporal data will be presented as map overlays. The user will be able to interact with the map and the provided tools in order to examine the evolution of a n event in space and time in order to understand the impact and the necessary measures.

a.1. User requirements

- The application should provide an UHI dashboard where summary information related to UHI forecast in each pilot area will be presented.
- The application should provide an advanced web map-centric page where data visualization and analysis could be performed.
- The application should have a dynamic / interactive map in which the geospatial datasets will be visualized. The users should be able to navigate (zoom in/out, pan) and request information at any point)
- The application should have a layer list in which the available datasets will be listed. For each dataset the user should be able to:
    - Turn on / off the visibility
    - Change the transparency
    - Zoom to extent
    - Change the visualization method
- The application should present a legend of the active layers
- The application should display both raw data and indexes as thematic maps.
- The application should display temporal data and provide an interface for easy selection of time.
- The application should provide time series plots at any given point. The charts should be interactive and easy to use.
- The application should allow to easily compare the forecast maps of any two parameters in different reference time
- The application should display alerts
- The application should generate a forecast report at any given point
- The application should enable data overlays from external data sources that supports OGC services like WMS and WFS
- The application should provide QUERY capabilities based on attribute and spatial data
- The application should provide easy access to metadata information. The users should be able to perform search on the metadata using any of the available data.
- The application should provide easy access to user manual

a.2. Mockups

Figure 8. Web application – Cartographic app mockup

Figure 9. Web application – Map comparison mockup

a.3. Use cases

The table below summarize the main use cases which will be presented in detail.

A/A	Use case	Users
1	Visualization of data	Public
2	Change Map Layer Style	Public
3	Get current meteorological conditions	Public
4	Get meteorological conditions history	Public
5	Get a meteorological forecast report	Public
6	Search Metadata	Public
7	Select reference day / time for forecast products	Public
8	Add external data sources	Public
9	View Alerts	Public
10	Compare forecast maps	Public
11	User registration	Public

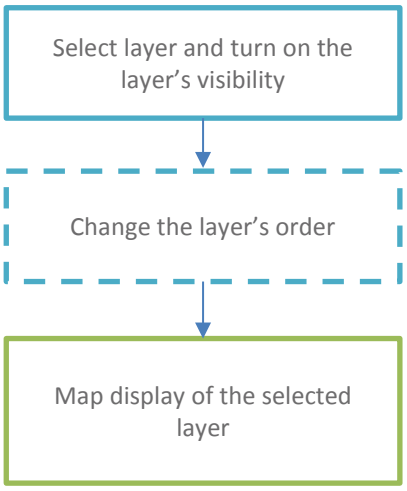
Table 2. List of web applications' use cases

In the detailed description of the use cases a flow diagram is also presented. The colors and the styles of the boxes are related to the actions characteristics as explained below:

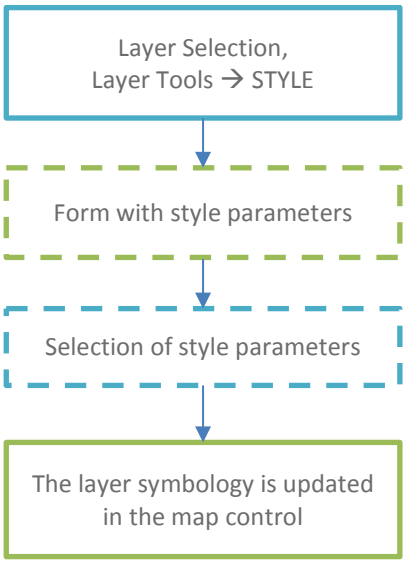
User action (Mandatory):

User action (Optional):

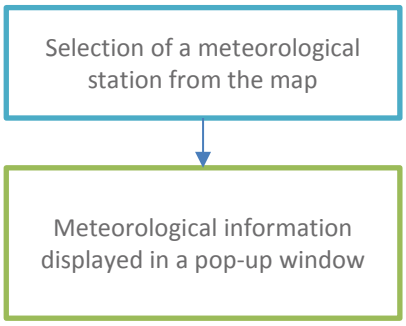
System action:

Use Case #	**1**	
Name	**Visualization of data**	
Users	Public	
Scope	Visualization of the geospatial data produced in the LIFE ASTI	
Abstract	The user selects the data that will be presented in the interactive map.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user turns on one of the preloaded layers available in the layer control	
	OPTIONAL: The user changes the order of the selected layer in the layer control by dragging it to the desired position	
		The system displays the requested layer in the map control
ACTIVITY DIAGRAM		

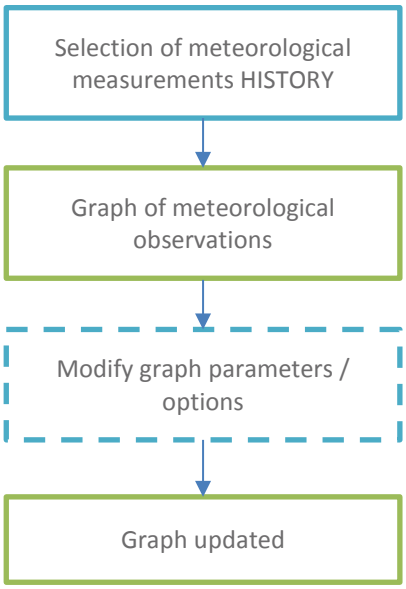
Use Case #	**2**	
Name	**Change Map Layer Style**	
Users	Public	
Scope	Change the symbology of a map layer	
Abstract	The user selects an alternative symbology for a specific map layer. This action will change the symbology of the layer. For example, a different color scale might be applied.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**

	The user selects one of the layers available in the layer control	
	The user clicks on the layer tools. From the available options selects STYLE and from the list of styles selects one.	
		OPTIONAL: If the selected style requires additional parameters to be defined like data range, color scale etc. a pop-up form appears
	OPTIONAL: The user provides the necessary inputs	
		The layer is updated in the map control using the new selected style
ACTIVITY DIAGRAM		

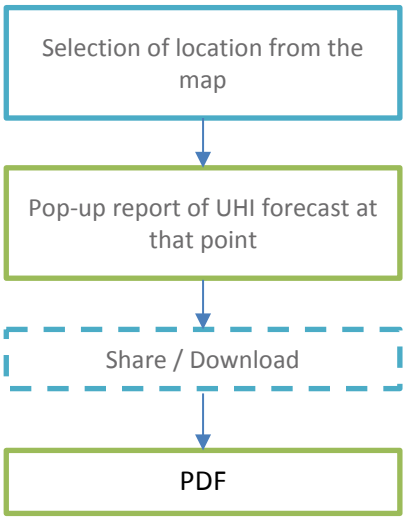
Use Case #	**3**	
Name	**Get current meteorological conditions**	
Users	Public	
Scope	Get information about the current meteorological conditions	
Abstract	The user interacts with the application in order to get the current meteorological condition in one of the available meteorological stations in the pilot area	
Prerequisites	The meteorological stations layer is visible	
Description of primary flow	**User action**	**System action**
	The user selects from the map one meteorological station	
		A pop-up window with the current (latest) meteorological information is displayed

ACTIVITY DIAGRAM	

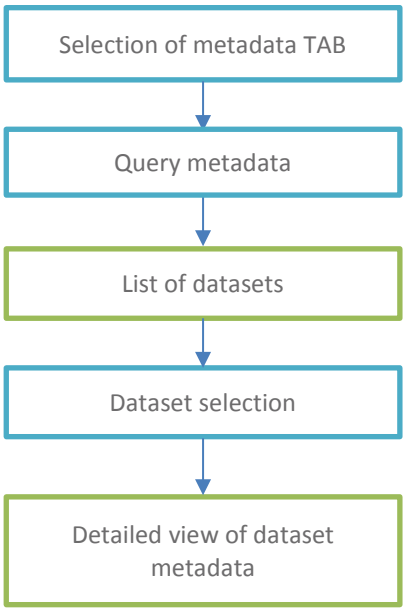
Use Case #	**4**	
Name	**Get meteorological conditions history**	
Users	Public	
Scope	Display a time series of meteorological parameters observed in a meteorological station	
Abstract	The user accesses the history of the measurements in a meteorological station	
Prerequisites	Execution of use case 3	
Description of primary flow	**User action**	**System action**
	The user selects the TAB history in the pop-up form	
		The system displays a graph of the observed meteorological parameters
	OPTIONAL: The user changes the graph options: Parameters on/off, date / period	
		The system updates the graph

ACTIVITY DIAGRAM	

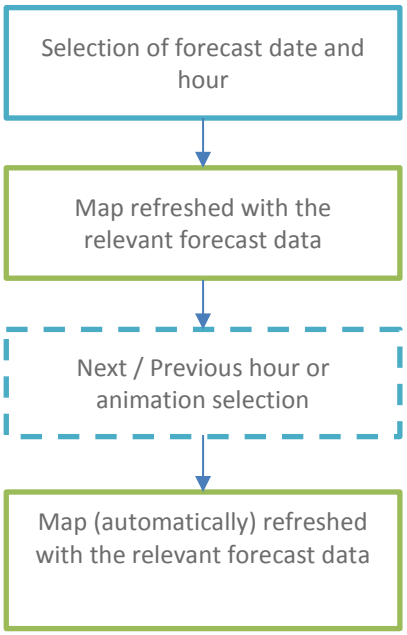
Use Case #	5	
Name	**Get a meteorological forecast report**	
Users	Public	
Scope	Get a forecast report with information related to UHI	
Abstract	The user selects any point at the map to view a UHI forecast report generated by the system for the specific point.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user clicks on the map and selects UHI report	
		The system, based on the selected location generates a report which presents information related to UHI. The information is presented in the most suitable way, similar to the dashboard app.
	OPTIONAL: The user wants to share or download the report	
		The system exports the report as PDF

ACTIVITY DIAGRAM	

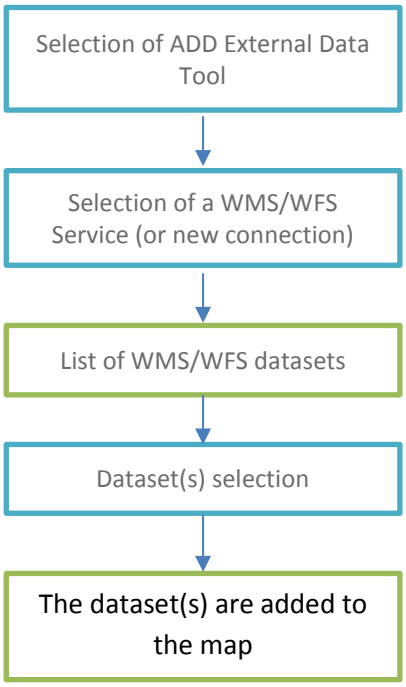
Use Case #	6	
Name	**Search Metadata**	
Users	Public	
Scope	Search for data / data services availability based on the available metadata	
Abstract	The user search for data or data services based on the available metadata. The search is performed using the metadata fields.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the Metadata TAB	
	The user defines the metadata search filters from the available metadata fields (category, agency, domain, keyword etc.)	
		The system displays a list of the datasets
	The user selects one of the datasets	
		The system displays detailed information about the dataset, including the extent (in a map view) and the OGC services endpoints.

ACTIVITY DIAGRAM	

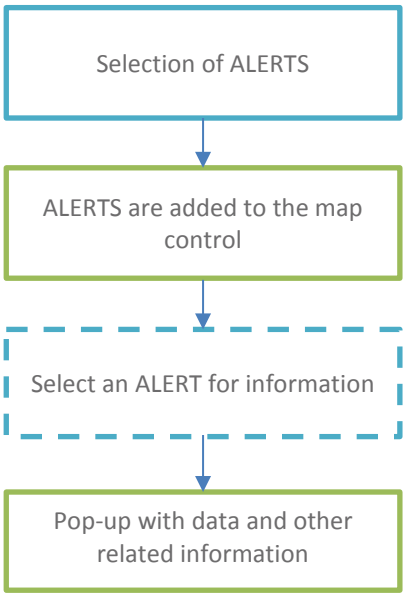
Use Case #	**7**	
Name	**Select reference day / time for forecast products**	
Users	Public	
Scope	Display information for a specific date and time	
Abstract	The user sets the date and time for which forecast date will be presented. The system will allow the transition between the available forecast times. An “animation” option will also be available.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the reference date and time from the provided control	
		The system updates in the map control all time depending datasets
	OPTIONAL: The user selects next / previous time or animation	
		The system updates in the map control all time depending datasets. If “play animation” is selected, the map is automatically refreshed every 5 seconds with the next hour data

ACTIVITY DIAGRAM	

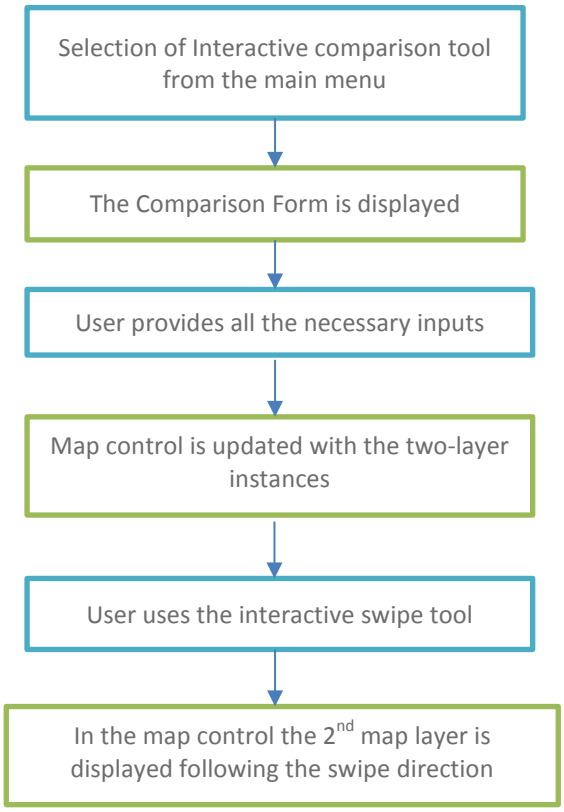
Use Case #	**8**	
Name	**Add external data sources**	
Users	Public	
Scope	Add other spatial data as map overlays	
Abstract	The user connects to an external data source using OGC WMS or WFS services. From the available datasets select one or more datasets which then are added to the layer list and to the map as overlays.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the add external data tool	
	The user selects one of the predefined WMS/WFS services or provides the necessary information for another service	
		The system connects to the service and provides a list of datasets
	The user selects the dataset(s) that will be added to the map	
		The system adds the selected dataset(s) to the layer list and to the map

ACTIVITY DIAGRAM	

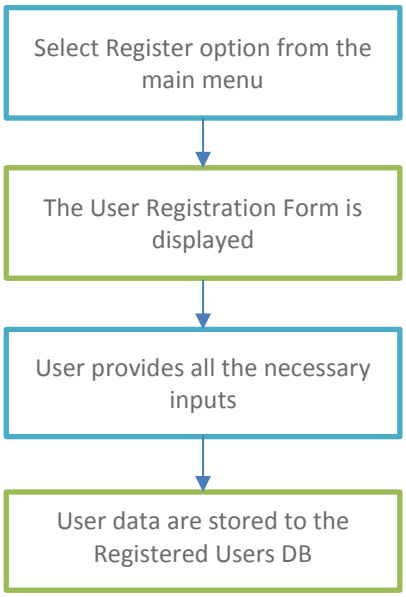
Use Case #	**9**	
Name	**View Alerts**	
Users	Public	
Scope	View UHI alerts	
Abstract	In case of UHI events alerts will provide information about the most affected areas.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the Show Alerts button	
		The alert layer will be automatically added to the map.
	OPTIONAL: The user clicks at any Alert point	
		The system displays a pop-up with the related information

ACTIVITY DIAGRAM	

Use Case #	**10**	
Name	**Compare forecast maps**	
Users	Public	
Scope	Compare forecast maps for different prediction time	
Abstract	Users will be able to select two different periods (date and hour) in order to interactively examine the spatial patterns of the predicted measure.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the interactive comparison tool	
		The interactive comparison form appears
	The user provides the required parameters (date and hour for two different periods)	
		The system updates the map control of the form with the two instances of the layer.
	The user uses the swipe tool to compare the two datasets	
		The map control shows / hides the 2nd map layer

ACTIVITY DIAGRAM	

Use Case #	**11**	
Name	**User registration**	
Users	Public	
Scope	Register user to system for personalized information	
Abstract	Users can be registered to the system in order to get personalized information. During the registration process some personal data should be provided.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the Register option	
		A registration form is displayed
	The user provides the necessary information	
		The system registers user to the Registered Users Database

ACTIVITY DIAGRAM	

b. Web application

The mobile application will provide an easy to use interface for personalized information. The application will use the GPS of the mobile devices to provide data for the exact location of the user.

The application will have the following pages:

1. Main page
   The main page will provide an overall view of the UHI in the pilot area for the forecasting period. The information will be presented in a simple to understand way (index with appropriate color scale). Information about precaution actions will be also provided.
2. Location information page
   The location will be defined by the device GPS, predefined areas available in a list, or favorite places which the users will have the option to define.
   For those areas information similar to the one provided in the main page will be displayed
3. Map View page
   This page will provide a map visualization of the results. The UI will be simple (previous hour, next hour, play, pause) will facilitate the users interaction.
4. Register page
   This page will allow users to register to the system (similar to the web user registration page). The system will support single sign-on, thus, if a user is registered in the web or the mobile app no additional registration is required.

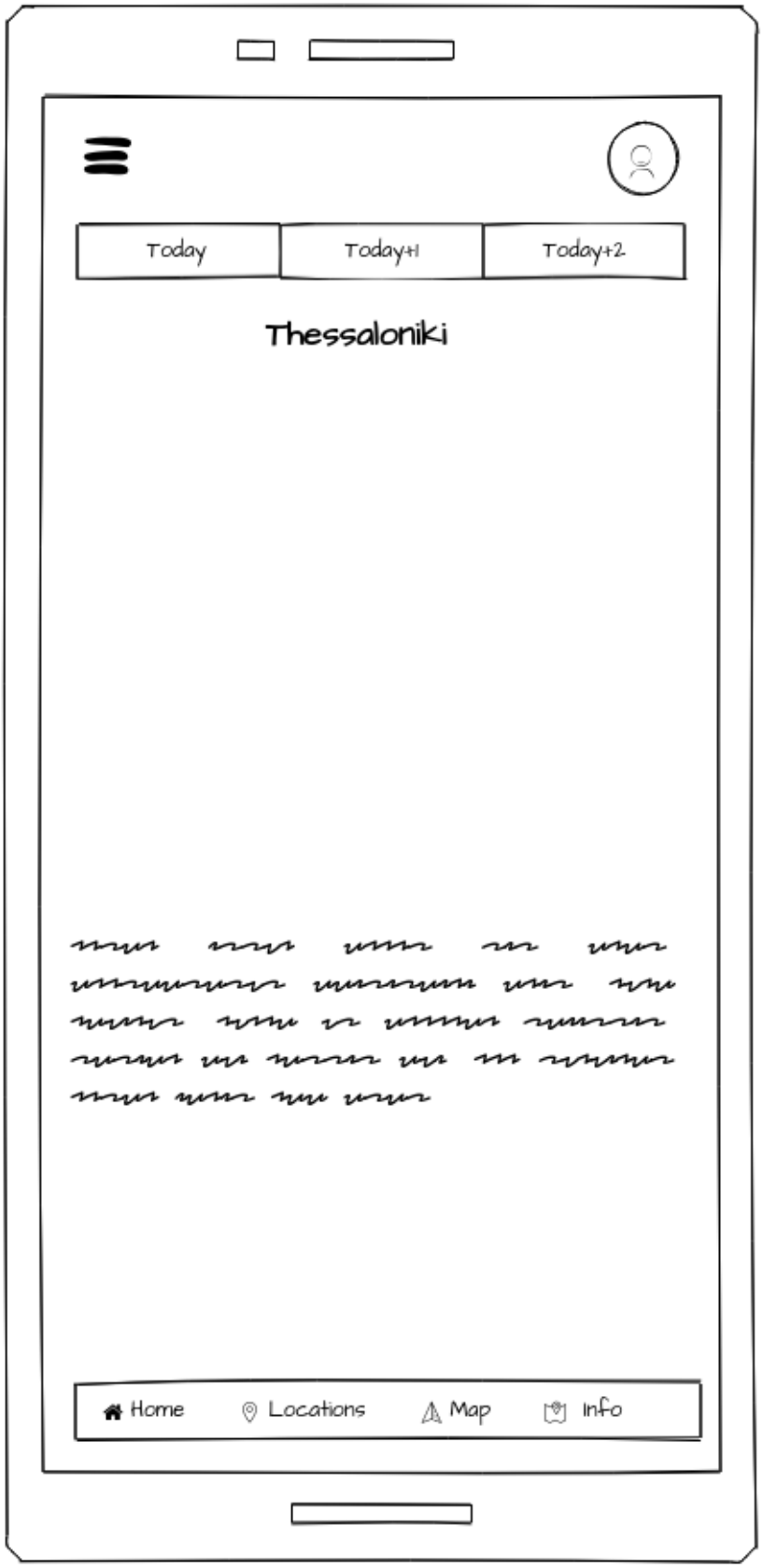
b.1. User requirements

- The application should provide a simple UHI dashboard where summary information related to UHI forecast in each pilot area will be presented.
- The application should provide a simple UHI dashboard for the current user location

- The application should provide a list of locations for which UHI predictions are available
- The application should allow the users to add locations in a favorite list
- The application should allow users to add favorite locations interactively from a map, using the GPS or by map selection
- The application should present the forecast results in a map with simple in time navigation tools
- The application should provide forecast information in a simple index form
- The application should provide precaution actions information
- The application should allow users to share the results to social media channels
- The application should allow users registration

b.2. Mockups

Figure 10. Login page


Today
Today+1
Today+2
Thessaloniki
Home
Locations
Map
Info

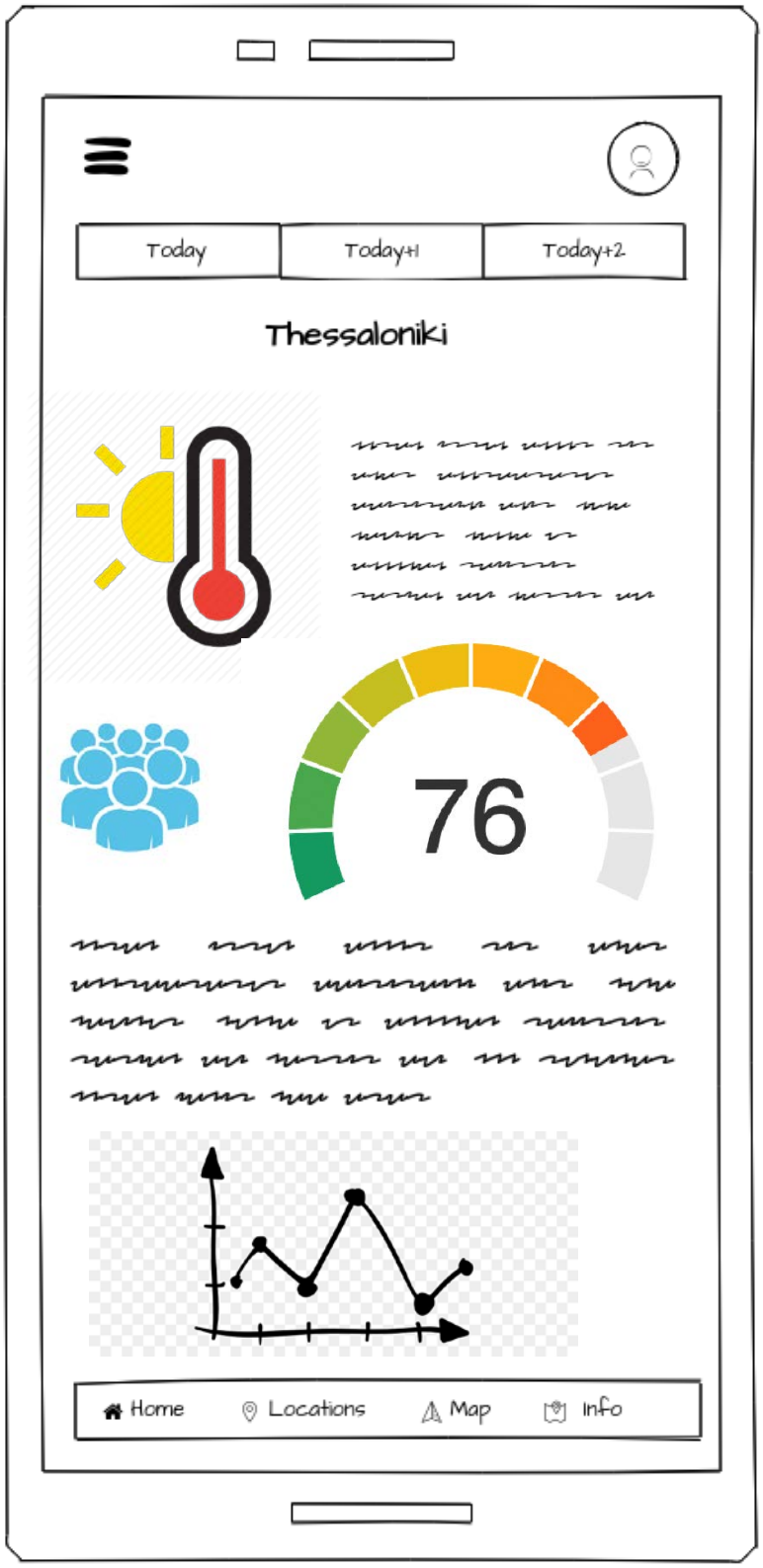


Figure 11. Main page – Overview dashboard

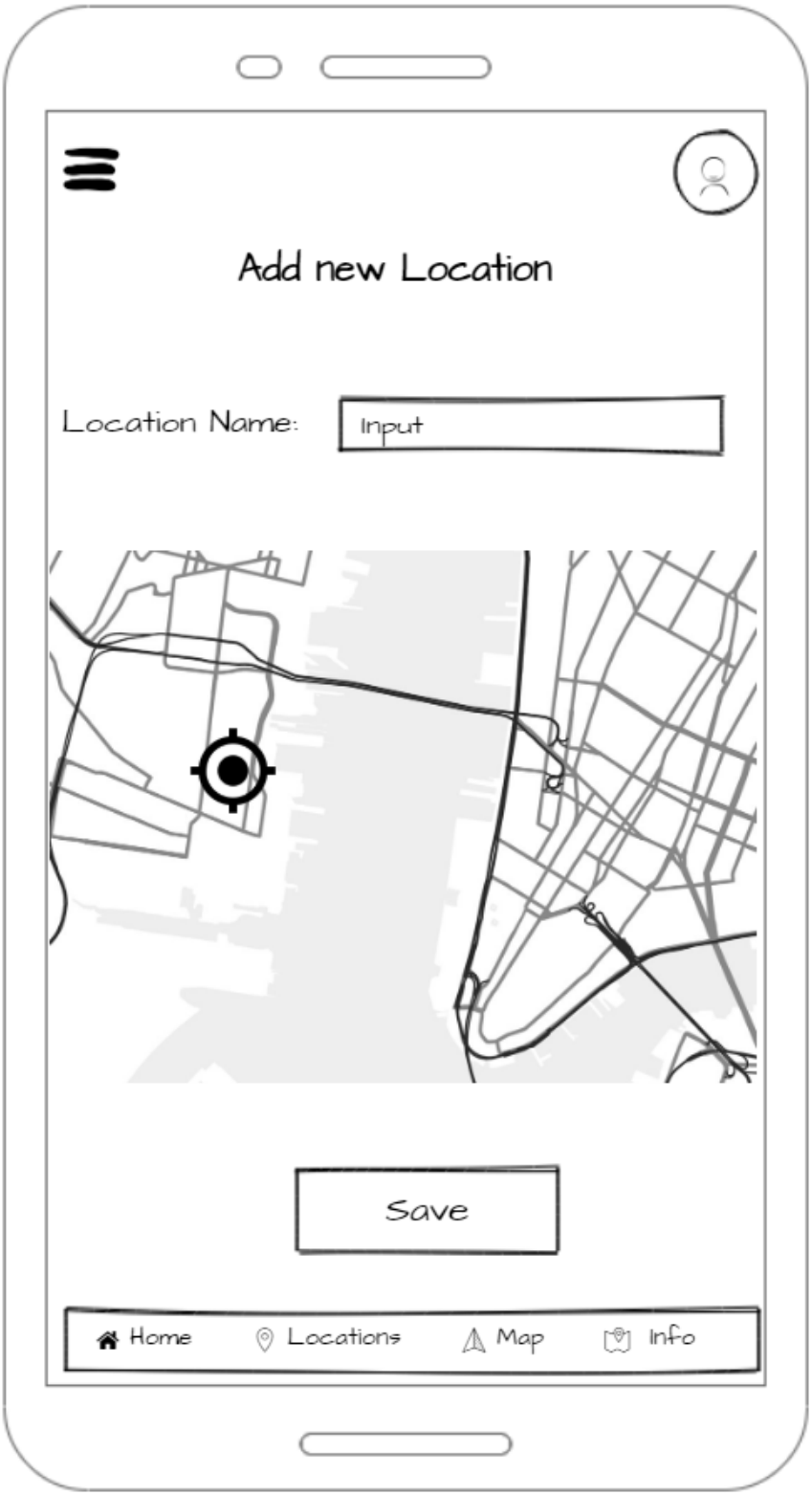


Figure 12. Add new location to favorite list

Figure 13. Map view of forecast data

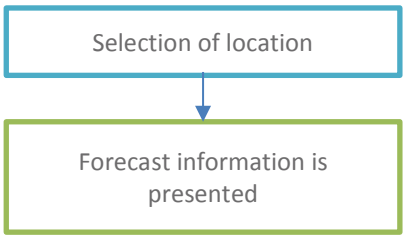
b.3. Use cases

The table below summarize the main use cases which will be presented in detail.

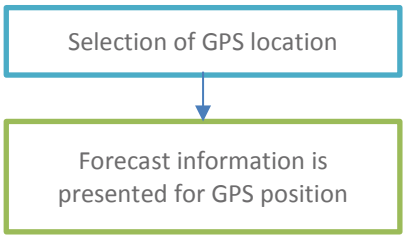
A/A	Use case	Users
1	Show forecast for predefined location	Public / Registered users
2	Show forecast for GPS location	Public / Registered users
3	Add location to Favorite locations	Registered users
4	Get precaution actions information	Public / Registered users
5	Show Maps of Forecasting variables	Registered users
6	User registration	Public

Table 3. List of mobilies' application use cases

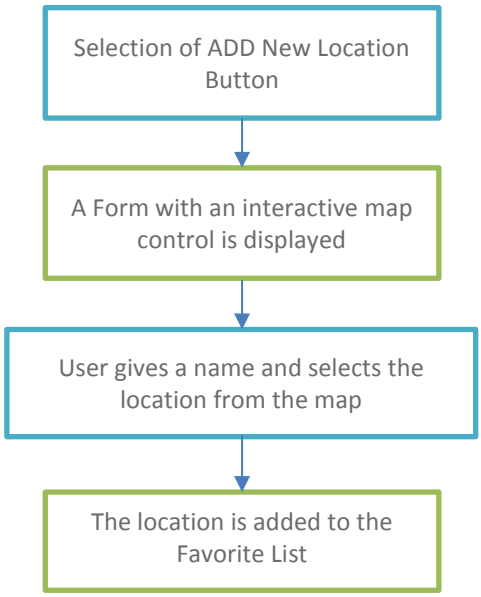
A detailed description of the above use cases is provided below.

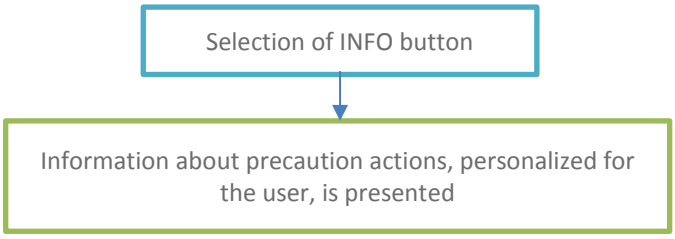
Use Case #	**1**	
Name	**Show forecast for predefined location**	
Users	Public Registered users	
Scope	Provide information from the meteorological forecast model	
Abstract	The user selects one of the available predefined locations to view, in a simple, easy to understand way, the results of the meteorological model.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects a location from the location list	
		The system refreshes the page with the results of the forecasting models. Information is presented as indexes which values can be easily interpreted
ACTIVITY DIAGRAM		

Use Case #	**2**
Name	**Show forecast for GPS location**
Users	Public Registered users
Scope	Provide information from the meteorological forecast model at user's location
Abstract	The user selects his/her current location to view, in a simple, easy to understand way, the results of the meteorological model.
Prerequisites	-

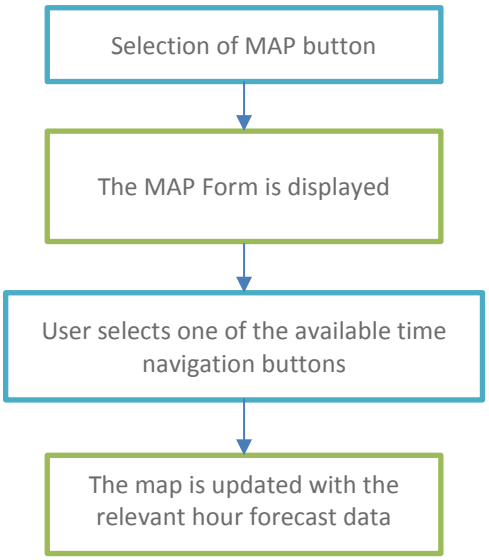
Description of primary flow	**User action**	**System action**
	The user selects Current GPS location from the main menu	
		The system presents a page with forecast information for the GPS location
ACTIVITY DIAGRAM		

Use Case #	**3**	
Name	**Add location to Favorite locations**	
Users	Registered users	
Scope	Save in favorites location a new location (like home, work, family etc.)	
Abstract	The user navigates in an interactive map to select a location to add to the Favorites List. Locations available in Favorite List can be used to quickly retrieve information for specific locations like home, office etc.	
Prerequisites	The user should be registered	
Description of primary flow	**User action**	**System action**
	The user selects the ADD New Location button	
		A map with a pin is displayed
	The user selects a location for the pin and gives a name for the location	
		The system adds this location to his/her Favorite List

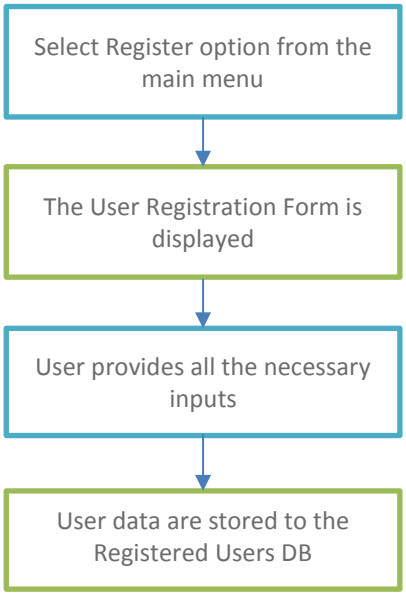
ACTIVITY DIAGRAM	

Use Case #	**4**	
Name	**Get precaution actions information**	
Users	Public Registered users	
Scope	Get information about precaution actions	
Abstract	The user requests information about precaution actions. This information is related to the event characteristics as well as to the user characteristics. For personalized information user registration is required.	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	The user selects the INFO button available for each forecast day	
		The system displays a text with information related to precaution action / measures.
ACTIVITY DIAGRAM		

Use Case #	**5**
Name	**Show Maps of Forecasting variables**
Users	Registered users

Scope	Show maps of forecast parameters (index, temperature)	
Abstract	Users are able to see maps of forecast parameters. The system provides an easy to use interface to navigate in time (next hour, previous hour, play, pause) to examine the temporal / spatial variation of the phenomenon	
Prerequisites	-	
Description of primary flow	**User action**	**System action**
	User selects the MAP button	
		A map of the current time is displayed
	The user selects one of the available navigation options: Previous hour Next hour Play / Pause	
		The map is updated
ACTIVITY DIAGRAM		

Use Case #	**6**	
Name	**User registration**	
Users	Public	
Scope	Register user to system for personalized information	
Abstract	Users can be registered to the system in order to get personalized information. During the registration process some personal data should be provided.	
Prerequisites	-	
Description of primary flow	**User action**	**User action**
	The user selects the Register option	

		A registration form is displayed
	The user provides the necessary information	
		The system registers user to the Registered Users Database
ACTIVITY DIAGRAM		

forecAsting System for urban heaT Island effect

The project Implementation of a forecAsting System for urban heat Island effect for the development of urban adaptation strategies- LIFE ASTI has received funding from the LIFE Programme of the European Union".