

Genealogy Web App

PROJECT WORKBOOK – TEAM GNLG₂

MAHDI CHAYMAE – COSTA CUNHA IVO – SAMIA LAHCENE – MACKPAYEN PRINCE – SAMIA OUSSAMA

Summary

1. Overview	1
1.1 Objective of the project	1
1.2 Organization	1
2. Requirements	2
2.1 Specifications	2
2.2 Environment	3
2.3 Use cases	4
2.4 Requirement analysis	4
3. Design	8
3.1 Design rationale	8
3.2 Static view	8
3.3 Dynamic view	9
3.4 Development planning	9
4. Engineering	10
4.1 Software organization	10
4.2 Build process	10
4.3 Tests	10
5. Deployment	10
5.1 Deliverables	10
5.2 Installation	10
5.3 Configuration	10
5.4 Operations	10
6. Annexes	11
6.1 Use case diagrams	11

Genealogy Web App – Project Workbook – Team GNLG2

Figure 1- Team flow chart.....	1
Figure 3 - Gantt Diagram part1.....	9
Figure 4 - Gantt diagram part2	10
Figure 4- Current system (Use Case).....	11

1. Overview

1.1 Objective of the project

The crowning of each university or technical formation is always done through an end of studies project or internship. Our training as part of this formation of a professional nature ends with disciplinary project spanning over two months.

In this context our training allowed us to take advantage of many opportunities to conduct group projects to refine our professional objectives.

Disciplinary group projects lead students to develop several transferable skills such as communication, cooperative and teamwork skills like planning, management, leadership and peer support. And so, we hope that this project will allow our team to develop or refine these skills.

The project itself consist of the modernization of a genealogy website called "expoactes" that allows genealogists to work together on old hard to read or partially erased birth, marriage and dead certificates in order to first digitalize them and secondly deduce family relations through them.

The project also consists of new functions to be added to the old website.

1.2 Organization

The development team members are composed of 1 project manager and 4 developers.

MAHDI Chaymae will assume the role of project manager, will communicate with the client but will also participate in the redaction of documentation, in the conception, and in the development as needed.

The 4 developers will be BOUSADIA Lahcene, MACKPAYEN Prince Divin, SAMIA Oussama, and COSTA CUNHA Ivo. Those developers will have the same tasks which will be the redaction of documentation, the conception and the development which will include of course the mandatory tests.

BOUSADIA Lahcene is nominated to take responsibility for Project Tracking.

The following organogram resume this point.

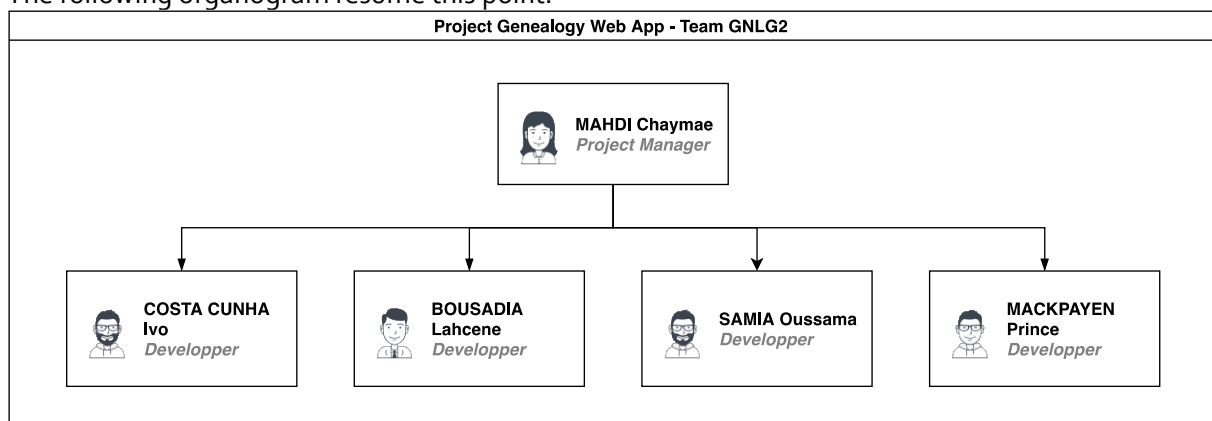


Figure 1- Team flow chart

1.2.1 Contacts

Name	Mail
Ivo COSTA CUNHA	Ivo.costa-cunha@etu.unice.fr
Oussama SAMIA	Oussama.samia@etu.unice.fr
Lahcene BOUSADIA	Lahcene.bousadia@etu.unice.fr
Chaymae MAHDI	Chaymae.mahdi@etu.unice.fr
Prince Divin MACKPAYEN	divinmackpayen@gmail.com

2. Requirements

2.1 Specifications

2.1.1 Specified objective of the project

The client specifications are the following:

1. "expoactes" is a system that allows to add and navigate in civil status acts. It is dated, in old PHP and with a user interface also dated. Upgrade "expoactes" in object-oriented PHP and update the user interface to a modern one. Why not also add visualizations, which leads to a second part of the project.
2. Using GEOCOM files (an export format that also allows the sharing of genealogy data), the client wants to obtain the following functionalities:
 - a. Obtain visualizations and navigations in the genealogy tree: on a map, a "traditional" tree, a "circular" tree (in fact they are graphs, an ancestor potentially being an ancestor for multiples "reasons").
 - b. Allowing the search of common ancestor by crossing 2 genealogies.
 - c. Have a social aspect by being notified of certain searches concerning the department, city name, patronymic, genealogist name "(les / sont des et / ou)".
 - d. A "standalone" version and a WordPress integration are desired.

The back end of the application must be written in PHP7 using the object-oriented paradigm since it's a client requirement.

Current website: <http://www.racinesardechoises.fr/expoactes/index.php?vue=T&xtyp=N>

2.1.2 Functional requirements

Functional requirements are those that must meet the requirements of the future system by terms of functionality. The identified needs are as following:

- The system must allow the user to visualize the different genealogical trees and navigate inside them either using a map visualization, a traditional tree or a circular tree.
- The user must be allowed to search a common ancestor by crossing 2 genealogies.
- The user must be allowed to find a potential cousinhood by crossing cities and names.
- The system must notify users about searches concerning department, city name, patronymic, or a genealogist name. "(les / sont des et / ou)"

2.1.3 Non-functional requirements

Non-functional requirements describe all the constraints to which the system is subjected for its realization and its proper functioning. These are the following:

- The system must be reliable.
- The UI must be ergonomic, user friendly and flexible.
- The system must be fast regarding data treatment.
- The code must be visible, commented and understandable in order to allow its evolution and extendibility.

2.2 Environment

In order to develop this web application our team will use the following software and technologies.

2.2.1 Software

- **GitHub:** It is the obvious choice to host our code in the cloud. It also allows versioning and the management of conflicts during a merge.
<https://github.com/Master1-MIAGE-UCA/GNLG2>
- **Balsamic:** It will allow us to create UI models for documentation and as a guideline to our developers.
<https://balsamiq.com/wireframes/>
- **Visual Studio Code:** This simple code editor shines in web and node development. Its functionalities extended by modules make it very adaptable to any project.
<https://code.visualstudio.com>
- **PhpStorm:** It's one of the best IDEs for PHP and that alone makes it an interesting choice for our project.
<https://www.jetbrains.com/phpstorm/>
- **Draw.io:** Easy to use, accessible in any navigator, web app that allows the creation of UML diagrams.
<https://app.diagrams.net>
- **Trello:** It will allow us to organize the project tasks and user stories, associate task with developers and follow what's done and what isn't at deadlines.
<https://trello.com/>
- **ArgoUML:** Another UML diagram editor.
<https://sourceforge.net/projects/argouml/>

2.2.2 Technologies

- **HTML/CSS/JavaScript:** The markup web language and its associated language CSS used to alter easily markups. It will affect the front-end of our application. JavaScript is the web scripting language today used extensively for a multitude of applications.
- **PHP7+:** Historical server-language and sometimes also used for the front-end that still updated. It will be used for back-end in our application.
<https://www.php.net>
- **VueJS:** It's a framework of JavaScript that has simple to use UI design libraries that will be used for the front-end of our application and that will allow us to develop a modern UI with the

use of the Material UI library. It also can be used to compile apps that can be used as standalone apps via Electron as desired by the client.

<https://vuejs.org>

- **Bootstrap5:** A UI JavaScript library that can be used as standalone with vanilla JavaScript.
<https://getbootstrap.com>
- **Laravel:** Laravel is a PHP framework with expressive and elegant syntax. It allows you to quickly develop the different functionalities of a project and improves PHP syntax for readability.
<https://laravel.com>

2.3 Use cases

2.3.1 Actors

Administrator: The site administrator has administrative right so he can allow visitors to make accounts that allow to have extensive access to the web site / app functionalities.

Registered visitor: A person with an account that can upload geocom files.

Visitor: A person that visit the site without account.

The in-depth study of the specifications allowed us to identify several use cases. It allowed us to structure the needs of the users and the corresponding objectives of the system, We represent, in the figure below, all the basic use cases in order to have a global view of how our application works, as well as any relationships that can take place.

See [figure 2](#) in annexes for the use case diagram.

2.3.1 Scenarios

2.4 Requirement analysis

2.4.1 Processing GEOCOM files

A GEDCOM file is a genealogical act saved in the format of **genealogical data communications** (GEDCOM) format. It contains family history acts and genealogical event data, as well as metadata linking the acts. GEDCOM files often contain information about births, deaths, marriages, children, and physical attributes of family members.

2.4.2 familytree365/Laravel-geocom

It is a package to analyse GEDCOM databases and import them as Laravel models. It allows, among other things, to use them and convert them back into other file formats.

2.4.3 System “expoactes” objective

“expoactes” is a software module making it possible to publish very simple examinations of acts and tables of civil status and parish registers.

2.4.4 The features of “expoactes”

“expoactes” documentation: <https://expocartes.monrezo.be>

2.4.4.1 Features of visitors

“expoactes” allows registered visitors to:

- browse the tables by municipality, then alphabetically by surname.

- search for acts by surname of interested parties or other appearing.
- access (if you authorize it) the details of the acts.
- management of the number of acts that can be seen by visitors in a given period.

For the management of acts, we have access to an administrative part of the site protected by login and password.

2.4.4.2 Management of the acts

The management of acts includes:

- loading of acts in NIMEGUE (Version 2 and 3) or CSV formats.
- the suppression of redundant acts and the reversal of marriage or promises acts.
- the re-export in NIMEGUE format of the documents you have submitted.
- management of access codes for visitors and depositors.
- control of database deposits and modifications.

2.4.4.3 Access levels

The main administrator can manage the access rights of visitors and depositors according to a 9-level hierarchy. If necessary, a points mechanism makes it possible to limit the number of consultations of acts during a given period.

The 9-level hierarchy:

1. Access to the list of communes and parishes
2. Access to the list of patronymics (by commune/parish and act type)
3. Access to the tables (names and dates)
4. Access to details of acts
5. Authorized to load data with NIMEGUE only
6. Authorized to load data of CSV type
7. ** Non used
8. Authorized to manage all data
9. Administrator

The general policy is that any registered visitor has access up to level 4.

2.4.4.4 Creation and edition of user profiles

A user profile is composed of:

- name
- surname
- mail address
- password (automatically created if wanted)
- login code
- access level
- status
 - N -> Normal
 - B -> Banned
 - + 2 others used during auto register

2.4.4.5 Import and export of acts by registered users

“expoactes” allows upload and export of acts by user if they are given the “holder” status of an act, otherwise only the administrators in the only one who is authorised to do it.

2.4.4.6 Other functions

“expoactes” also have function, for the administrator, to send circular emails, and if during the inscription the visitor choose to being send by mail his credentials to login later he can choose.

The use case which is in the annex represents the different interactions possible between the system and the actors as well as between the actors.

It will help us to understand the functions and the roles of each actor of this system.

After identifying and describing the different actors of the system, we now have to describe the different possible scenarios :

1.a.To log In :

The various system actors must log in or have an account before accessing the system's functionalities.

1.b.Give access to visitors to connect :

The administrator must be able to give access to simple visitors to consult their information even if they have an account to connect, they must wait for the right of access according to the administrator of the application.

1.c. send circular mails :

Admin must be able to send circular emails ,of another term he can send email to all group members. and he must have feedback if an email address no longer exists.

2.a.Change the statut of a user :

The administrator must be able to change the status of registered users so that he can for example change the access rights of a user from level 3 to a level still high of 4.

2.b.Block IP :

When an IP is banned by the admin we will have two possible scenarios either by :

2.b.a : Remove from IP list :

The administrator must be able to delete IPs from a list

2.b.b :Keep IP :

either keep it in the list but set its status to unblocked

2.c.Consult family tress :

The administrator and the registered user must be able to consult the geological trees.

3.a.Research at tree level :

The administrator and registered user must be able to search at tree level

the administrator and the registered user can also do research by crossing two genealogical trees.

3.b.Import gedcom files :

The administrator and registered user can import new different gedcom files. either of births or deaths or even of marriage.

4.a.Get visualisations and navigations in family tress :

The administrator and the registered user must be able to visualize the family trees either in card form, or on traditional tree or on circular shaft.

4.b. Find potential cousinship by crossing surnames and cities :

Scenario as-is

The user seeks to find potentials cousinships by crossing surnames and cities precondition

The user select « cousinship search »

The use select a city

The user select a name

The system search using the city and name as filter

The system find a match

The system show in the UI the results of the research

Extension 1

3.a The user enters an incorrect city name

3.b The systems shows an error to the user when the user finish typing

Follow up on 3

Extension 2

4.a The user enters an incorrect name

4.b The systems shows an error to the user when the user finish typing

Follow up on 4

Variant 1

6.a The system doesn't find a match

6.b The system displays in the UI that there is no match.

Follow up on 1

4.c. Register and create an account on the application forum :

Scenario as-in

The user seeks to register himself in « expoactes »

The user clicks on « retour sur le forum »

The user clicks on « connexion »

The user clicks on « Register »

The user inform asked information

The user clicks on « Register »

The administrator of « expoactes » validates the user account

The system informs the user that his candidacy is accepted and is notified with a mail that also contains his credentials

The user has access to « expoactes »

Extension 1 - Negative scenario

- 2.a The user doesn't is lost in the site and doesn't know that « retour sur le forum » allows him to register himself
- 2.b The exit quits the website

Extension 2 - Negative scenario

- 3.a The user doesn't know that « connexion » allows him to register himself
- 3.b The user exit the website

Extension 3 - Negative scenario

- 7.a The administrator of « expoactes » refuse the user
- 7.b The user is informed by the system by mail that his candidacy is refused.
- 7.c The user has not access to « expoactes »

Interactions between actors:

In the use case diagram we presented a relation between the administrator and the registered user by a generalization relation because we thought that in the case where the administrator is absent the registered user can replace him and make its functionalities while waiting for it to arrive

3. Design

3.1 Design rationale

3.1.1 Website host

The current website uses a MySQL database and is written completely in PHP.

Since the project has as requirements the modernisation of the PHP code and to not change the database a host that allows both must be used. Since our team wants to use VueJS for the front-end since it provides tools for creating a modern interface the host must also be able to that code. OBS was proposed and seems to do both.

3.2 Static view

The objective of this project is to modernize the existing web page (<http://www.racinesardechoises.fr/expoactes/index.php?vue=T&xtyp=N>) with a modern design and add new functionalities while keeping the old ones functional.

Using a GEOC file the new application must possess the following functionalities:

- *Get visualizations and be able to navigate in genealogy trees.*
- *Be able to search by common ancestor or by crossing 2 genealogies.*
- *Find a possible cousinhood by crossing cities and names.*
- *The new application must have a social aspect which is the notification of the user when **certain searches** concerning **his/her** department, city, name and researcher name happen.*

In this part, the list of internal technical components to be produced will be given.

3.2.1 Manipulation of the expoact system database

It must be able allowed to analyse the database of the expoact system and import it in order to use it to visualize the data of people.

3.2.2 Analysis of GEOCOM files

In order to be able to analyse GEOCOM files, our team must first know that all the information in the GEOCOM file is described by keywords (eg: GIVN for first names, PLAC for location...) and linked together by references (eg: @ I.... @ For individual, @ F... @ for family) which allow genealogy software to reconstruct your tree from the file. So this information can help us to make a program which is able to parse and read a GEOCOM file.

Exporting or importing a GEDCOM file is a mandatory function of genealogy software.

We must be able to process an entire family tree, but also an ascending or descending branch from a selected person.

A GEDCOM file is divided into:

- A header section (HEAD).
- Various recordings of various kinds:
 - "person" record (INDI = individual)
 - "family" record (FAM = family)
 - "note" record (NOTE = note)
 - "source" record (SOUR = source)
 - "archive repository" record (REPO = repository)
- multimedia object" record (OBJE = object)
- end of file marker (TRLR = trailer)

3.3 Dynamic view

3.4 Development planning

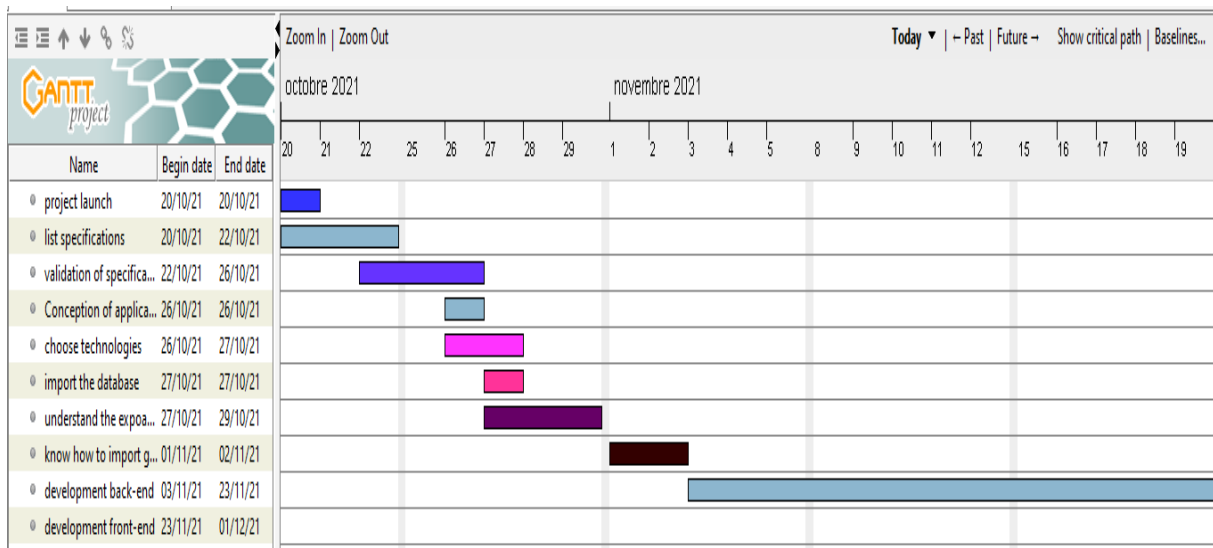


Figure 2 - Gantt Diagram part1

Genealogy Web App – Project Workbook – Team GNLG2

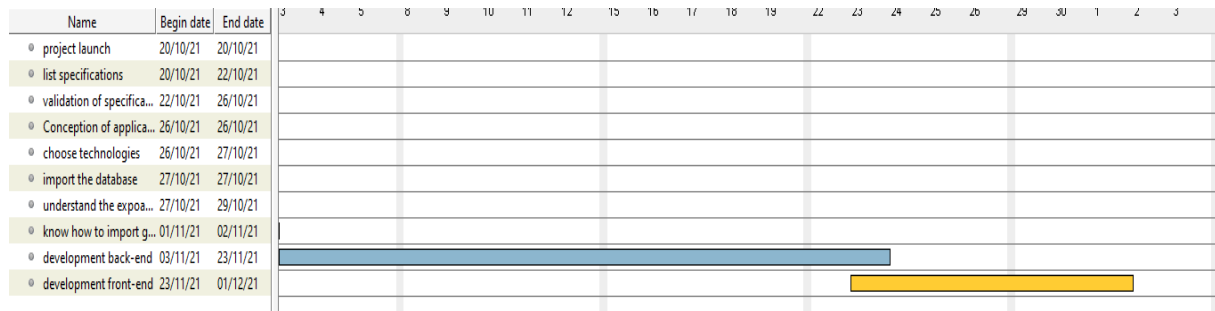


Figure 3 - Gantt diagram part2

4. Engineering

4.1 Software organization

4.2 Build process

4.3 Tests

5. Deployment

5.1 Deliverables

5.2 Installation

5.3 Configuration

5.4 Operations

6. Annexes

6.1 Use case diagrams

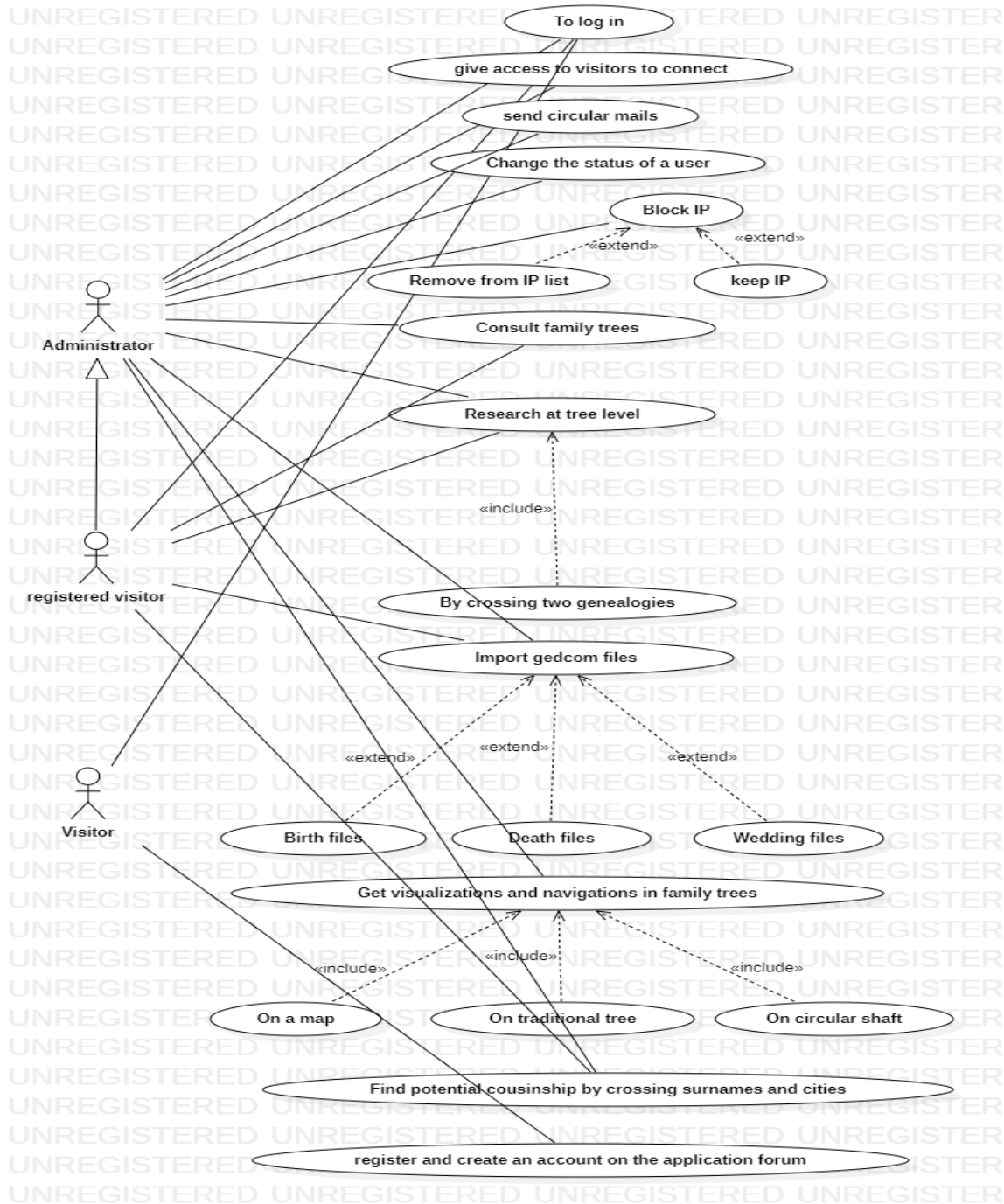


Figure 4- Current system (Use Case)