Genealogy Web App

PROJECT WORKBOOK – TEAM GNLG2

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

2020-2021

Summary

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 formationof a professional nature ends with disciplinary project spanning over two months.
In this context out 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.

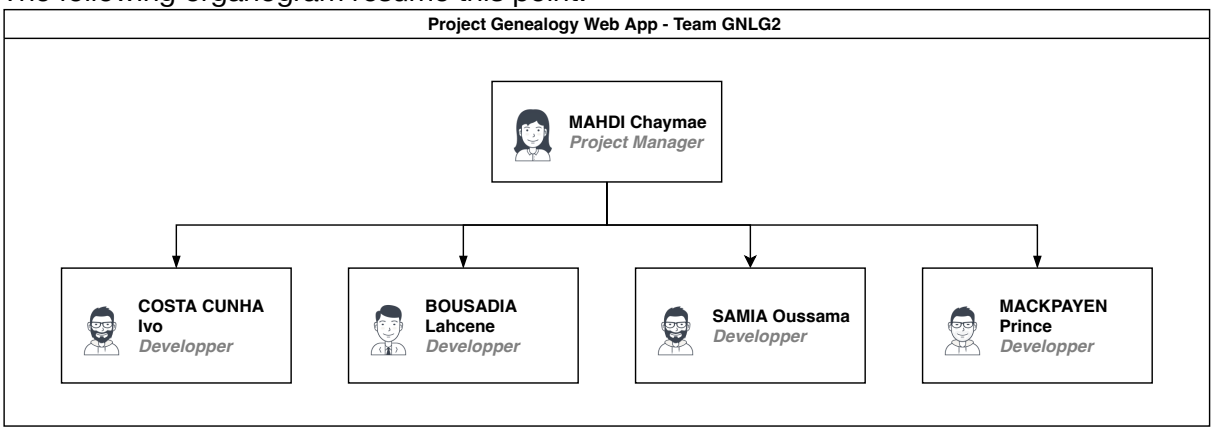


Figure 1- Team flow chart

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 also not add visualizations, which is 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

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

2.1.2 Analysis of GEOCOM files

The GEOCOM format was created to allow to share genealogical data in 1980 by “L’Eglise de Jesus-Christ des saints des derniers jours”. The objective was to give a solution to the exchange of data between 2 different genealogy software.

The GEDCOM file contains information about each person, each family and each event of a tree.

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)

See annexes (5&6).

2.1.3 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)”**

2.1.4 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 able to minimize the processing time in maximum 5s .
- 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 allows 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 our application and that will allow us to develop a modern UI with the use of the Material UI library. Il also can be used to compile apps that can be uses 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.

See annexe (7) for the use case diagram.

2.3.1 Scenarios

After identifying and describing the different actors of the system, we can now describe the different possible scenarios.

2.3.1.1 Log In

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

2.3.1.2 Give access to visitors

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.

2.3.1.3 Send circular mails

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

2.3.1.4 Change the status 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.

2.3.1.5 Block IP

When an IP is banned by the admin it will be added to a list of banned IPs.

2.3.1.6 Remove from IP list

The administrator must be able to delete IPs from a list

2.3.1.7 Keep IPs

The administrator must be able to keep an IP in the list but set its status to unbanned.

2.3.1.8 Consult family trees

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

2.3.1.9 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.

2.3.1.10 Import GEDCOM files

The administrator and registered user can upload new different GEDCOM files. Either of births, deaths or marriage.

For the user to upload new GEOCOM files the administrator must give to the registered user a special permission, since in a normal case only the administrator can upload those. The registered user also become responsible for this particular GEDCOM file that he or she uploaded.

2.3.1.11 Get visualisations and navigations in family trees

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.

2.4 Requirement analysis

2.4.1 “expoactes” functionalities resume

The current system is a GEOCOM database manager that gives access to the administrator and to registered users to GEOCOM data that is translated from acts of birth, death and marriage from the years 1800 until today. The objective is to digitalise those acts that are in paper form, most of the time hard to read and subject to be destroyed by time, into MySQL data objects using GEOCOM file format.

To achieve this objective the “expoactes” system, a web software coded in old PHP, has different functionalities that are described next to this.

To access the data a visitor must be register himself or herself and the administrator must validate this inscription.

We can reformulate this by certain scenarios existing in the system:

- 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 informs 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 »

See (2) and (3) for inscription steps.

Problems
The visitor must follow the links in the red circles to register himself. We think those steps are confusing and lead to a lot of negative scenarios where the user leaves the website because he is unable to register himself or even know that he must register himself first to access the data. We aim to improve the UI in our modernised application in a way for it to be intuitive.

The following alternatives and negative scenarios show these problems.

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 exits 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 »

2.4.2 Problems faced by the team

2.4.2.1 Deadline problems

The client asks way too many functionalities for the given time. After evaluations the time that each feature would take the team decided to not do the following ones:

- Standalone App
- WordPress integration
- Renovate current code in modern PHP

Standalone App
We determined that that task would take alone at least 1,5 months to our team and so it's not possible to do it in the time given by the client to the project.

WordPress integration
While the team determined that this task would take scientifically less time than the standalone app it still prolongs the project past deadline and so we decided to remove it.

Renovate current code in modern PHP
The team decided that it would be faster to rewrite new code than to reuse the old one due to the volume of it and complexity of it (214 hard to understand files). We will make sure that the new application will provide all the same functionalities, however.

2.4.2.2 Database restricted access

The team figured out that the new app require access in writing to the actual used MySQL database, however we only have the reading access. We need the client to give us writing permissions in order to complete the new application.

2.4.3 Cousinhood search

The cousinhood search should follow this scenario.

Find potential cousinship by crossing surnames and cities

Scenario as-is
1. The user seeks to find potentials cousinship by crossing surnames and cities
2. The use selects "cousinhood search"
3. The user selects a city
4. The user selects a name
5. The system search using the city and name as filter
6. The system find a match
7. 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 show 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 show 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

2.4.4 Processing GEOCOM files

A GEDCOM file is a genealogical act saved in the format of **ge**nealogical **d**ata **com**munications (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.

2.4.5 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.

2.4.6 The features of "expoactes"

"expoactes" documentation: https://expocartes.monrezo.be

2.4.6.1 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.

See (1) for more details.

2.4.6.2 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.

2.4.7 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.

2.4.7.2 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

2.4.7.3 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.

2.4.7.4 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.

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.

3.1.2 Creation and integration of an HTML template in a Laravel project

For the creation of our project, we decided to create a Laravel project in order to properly structure our code files and to integrate an HTML template into this project.
We decided to work on a Laravel project because it is an open-source web framework written in PHP respecting the model-view-controller principle which allows us to properly organize the structure of our project, in addition to that it's fully developed in object-oriented programming.
Laravel contains several object-oriented libraries which will facilitate the management of GEDCOM files, and which contains documentation that is clear and easy to understand.

3.1.3 Creating diagrams using ArgoUML

We modelled the diagrams of our project using the ArgoUML tool because it is free, multilingual software that supports code generation and reverse engineering, and especially that the generation of code from class diagrams is supported in PHP languages.

3.1.4 Creation of application mock-ups using balsamic software

Balsamic software is a software editor that has helped us to make UI design, that allows us to collaborate, create mock-ups, version control and run user tests.

The creation of the models of the application will serve us to see the real structure of the different pages of our applications and its functionalities and we decided to make these models to concretize the ideas of the project. which can then serve other secondary objectives: to explore an idea, to test it, or to convince stakeholders or its hierarchy of the concept.

3.2 Static view

After studying the concept of the project well and analysing the difficulties that we will find during the realization of this project, we found that the first aspect of the project contains several unclear and ambiguous points which still require clarification according to the client. We decided to start with the second aspect of the project which is the processing of GEDCOM files.

For that, we decided to create an application that allows the management and processing of GEDCOM files, which we must be able to:

- Obtain visualizations and navigations in the family tree: on a map, in a “traditional” tree and in a “circular” tree.
- Allow the search for a common ancestor by crossing 2 genealogies.
- Find potential cousinship by crossing surnames and cities.
- The web app must have a social aspect be informed of certain searches concerning department / city name / by patronymic / name of the researcher (genealogist).

In this part we will define the different tasks to be carried out for the second aspect of the project which are as follows:

- Research the handling of GEDCOM files
- Make app mock-ups
- Make a class diagram for this application.
- Creation of a Laravel project and integration of an HTML template in there.
- Creation of a BD where we will store the GEDCOM files selected by the user
- Learn how to read GEDCOM files and understand how to retrieve information from annotations used like @Fam ...
- To learn how to navigate in the different family trees.
- Make and implement algorithms for the visualization of family trees in the form of a map
- Make and implement algorithms for the visualization of family trees in the form of a circular tree
- Make and implement algorithms for the visualization of family trees in the form of a traditional tree
- Make known and implement how to cross two genealogies
- Research a common ancestor by crossing two genealogies
- Find potential cousinship by crossing surnames and cities
- Be notified of certain searches concerning department / city name / by patronymic / researcher's name.

3.3 Dynamic view

In this part, we will show the different models of our application to show the different interactions between components.

See annexes (8) (9) (10) (11) (12) (13) (14) (15) (16) (17).

3.4 Development planning

3.4.1 Task IDs

For each task type we assign an ID. They will also correspond to project tracking ones.

- GM: General management
- GM.MET: Team meetings
- WR: Workbook Redaction
- PT: Project tracking
- PC: Project conception
- PC.APP: App conception
- DEV: Development
- DEV.DB: Database tasks
- L: Learning
- L.GF: Learning GEDCOM files
- L.DB: Learning DB structure

3.4.2 Team IDs

Each member of the team has an ID made from her or his name initials.

- ICC: Ivo COSTA CUNHA
- CM: Chaymae MAHDI
- PDM: Prince Divin MACKPAYEN
- LB: Lahcene BOUSADIA
- OS: Oussama SAMI

3.4.3 Task assignment and estimated times

Id	Task description	Time estimate	Task performed	Remains to be done	Actors
WR	Writing the workbook	5h*j	Done	-	CM + ICC
PT	Make the follow-up file (Project tracking)	1h*j	Done	-	CM + ICC
L.GF	Research the handling of GEDCOM files	5j*h	Done	-	PDM
CP	Make a class diagram for this web application	1h*j	In progress	-	ICC
CP	Creation of a Laravel project and integration of an HTML template in there	5h*j	Done	-	OS
DEV.DB	Creation of a DB where we will store the GEDCOM files selected by the user	1h*j	In progress	-	OS
L.GF	Learning how to read GEDCOM files and understand how to retrieve information from annotations	4h*j	In progress	-	PDM
PC.APP	Make app mock-ups	2h*j	Done	-	CM

DEV	Development of application functionalities	5h*j	Not yet	- Make and implement algorithms for the navigations and visualizations of family trees. - Implement search by crossing two genealogies. - Research a common ancestor by crossing two genealogies. - Find potential cousinship by crossing surnames and cities. - Be notified of certain searches.	CM ICC OS PDM LB
GM.MET	Team meetings	3h*j	In progress	-	CM ICC OS PDM LB

4. Engineering

4.1 Software organization

4.1.1 Classes

Since we aim to use object-oriented programming paradigm a class diagram must be done. Our initial diagram class is the following (18).

4.2 Build process

4.3 Tests

5. Deployment

5.1 Deliverables

5.2 Installation

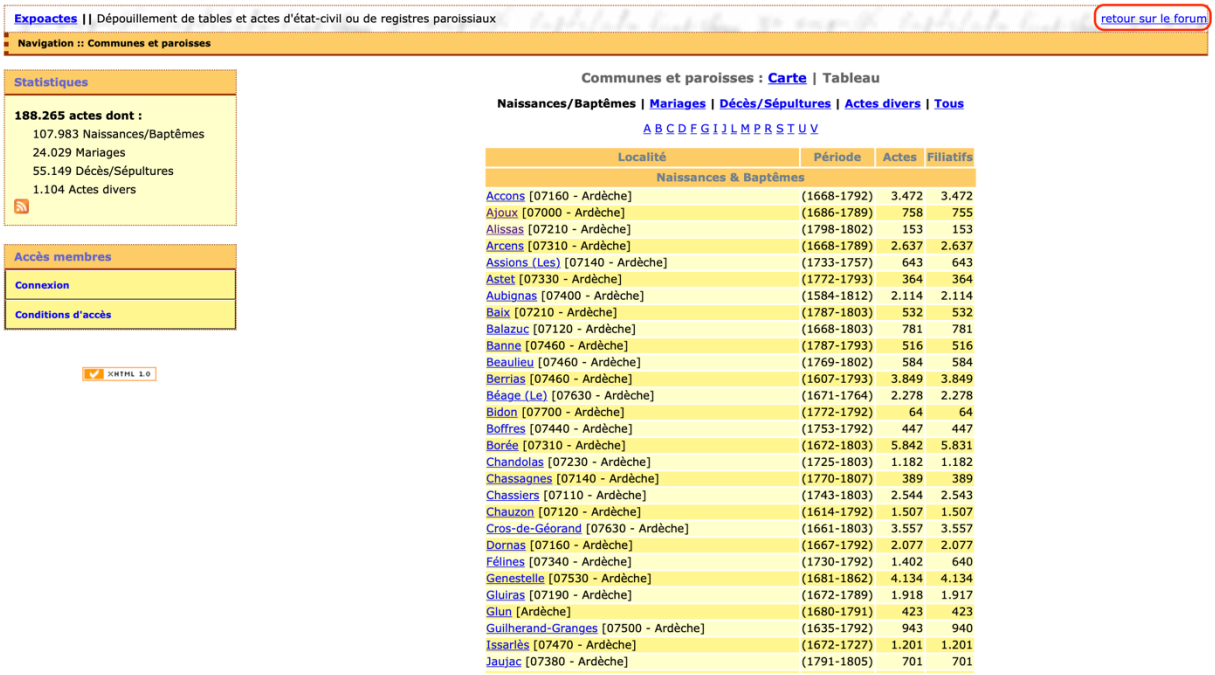
5.3 Configuration

5.4 Operations

6. Annexes

6.1 Current application screen captures

6.1.1 Current application welcome page



Expoactes || Dépouillement de tables et actes d'état-civil ou de registres paroissiaux

retour sur le forum

Navigation :: Communes et paroisses

Statistiques

188.265 actes dont :
107.983 Naissances/Baptêmes
24.029 Mariages
55.149 Décès/Sépultures
1.104 Actes divers

Accès membres

Connexion

Conditions d'accès

XHTML 1.0

Communes et paroisses : Carte | Tableau

Naissances/Baptêmes | Mariages | Décès/Sépultures | Actes divers | Tous

A B C D F G I J L M P R S T U V

Localité	Période	Actes	Filiatifs
Naissances & Baptêmes			
Accons [07160 - Ardèche]	(1668-1792)	3.472	3.472
Ajoux [07000 - Ardèche]	(1686-1789)	758	755
Alissas [07210 - Ardèche]	(1798-1802)	153	153
Arcens [07310 - Ardèche]	(1668-1789)	2.637	2.637
Assions (Les) [07140 - Ardèche]	(1733-1757)	643	643
Astet [07330 - Ardèche]	(1772-1793)	364	364
Aubignas [07400 - Ardèche]	(1584-1812)	2.114	2.114
Baix [07210 - Ardèche]	(1787-1803)	532	532
Balazuc [07120 - Ardèche]	(1668-1803)	781	781
Banne [07460 - Ardèche]	(1787-1793)	516	516
Beaulieu [07460 - Ardèche]	(1769-1802)	584	584
Berrias [07460 - Ardèche]	(1607-1793)	3.849	3.849
Béage (Le) [07630 - Ardèche]	(1671-1764)	2.278	2.278
Bidon [07700 - Ardèche]	(1772-1792)	64	64
Boffres [07440 - Ardèche]	(1753-1792)	447	447
Borée [07310 - Ardèche]	(1672-1803)	5.842	5.831
Chandolas [07230 - Ardèche]	(1725-1803)	1.182	1.182
Chassagnes [07140 - Ardèche]	(1770-1807)	389	389
Chassiers [07110 - Ardèche]	(1743-1803)	2.544	2.543
Chauzon [07120 - Ardèche]	(1614-1792)	1.507	1.507
Cros-de-Géorand [07630 - Ardèche]	(1661-1803)	3.557	3.557
Dornas [07160 - Ardèche]	(1667-1792)	2.077	2.077
Félines [07340 - Ardèche]	(1730-1792)	1.402	640
Genestelle [07530 - Ardèche]	(1681-1862)	4.134	4.134
Gluiras [07190 - Ardèche]	(1672-1789)	1.918	1.917
Glun [Ardèche]	(1680-1791)	423	423
Guilherand-Granges [07500 - Ardèche]	(1635-1792)	943	940
Issarlès [07470 - Ardèche]	(1672-1727)	1.201	1.201
Jaujac [07380 - Ardèche]	(1791-1805)	701	701

Figure 2 - Current application welcome page

6.1.2 Web site register procedure

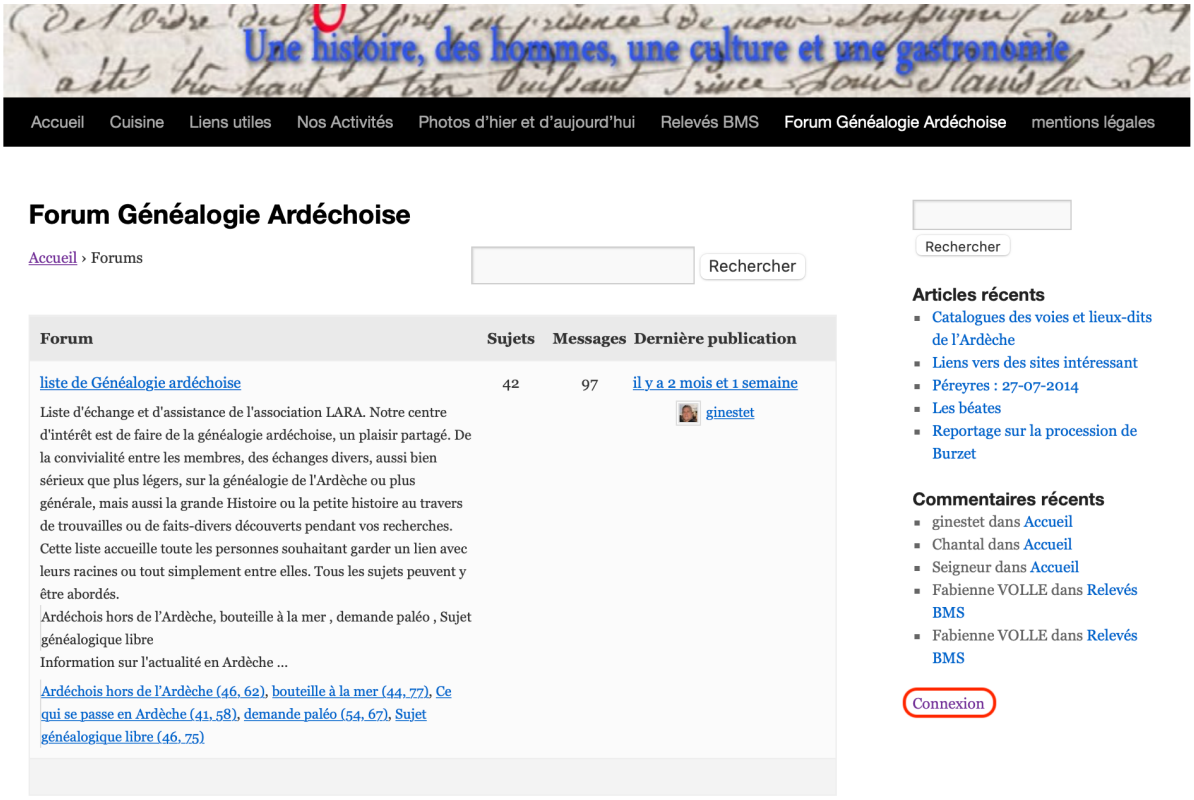


Figure 3 - Current application registering procedure (phase 1)

Log In

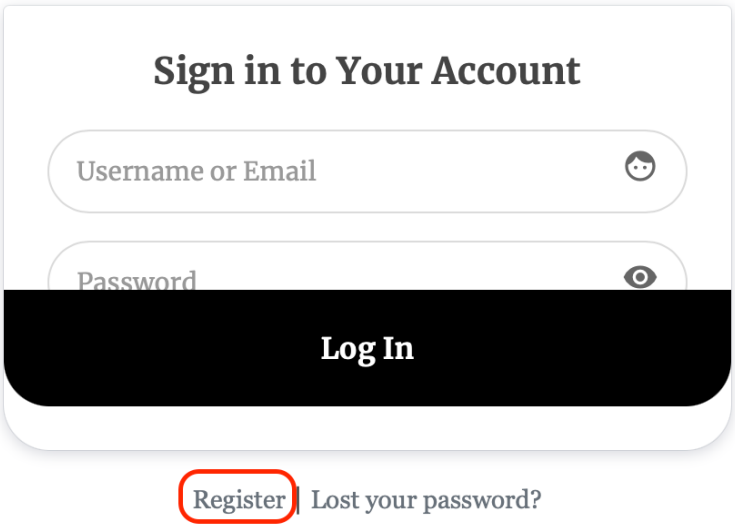


Figure 4 - Current application registering procedure (phase 2)

6.2 Other screen captures

6.2.1 GEDCOM data structure

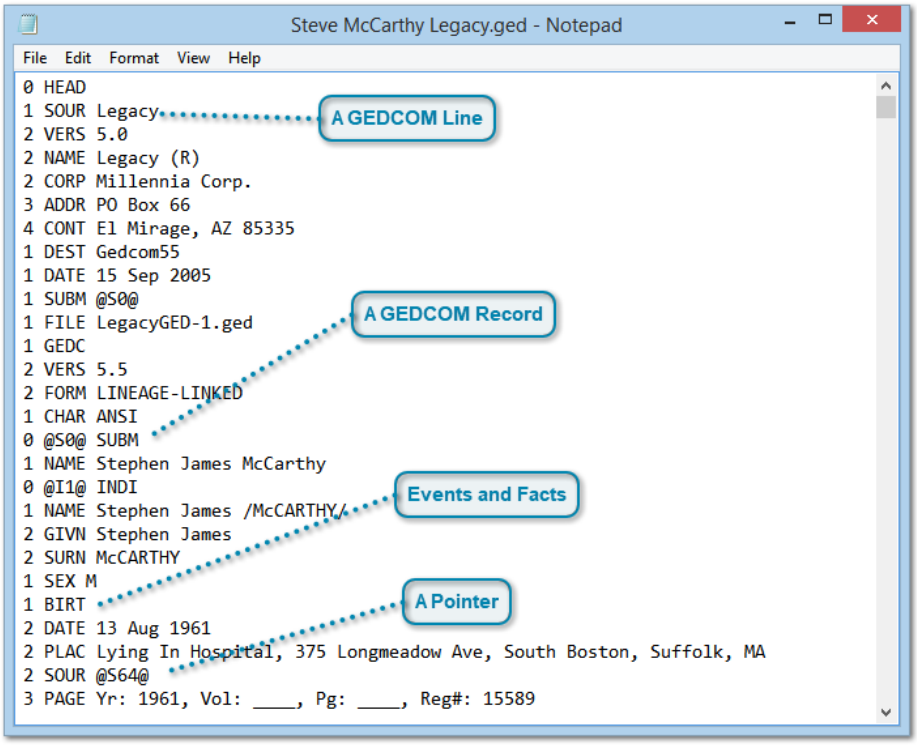


Figure 5 - GEDCOM data structure

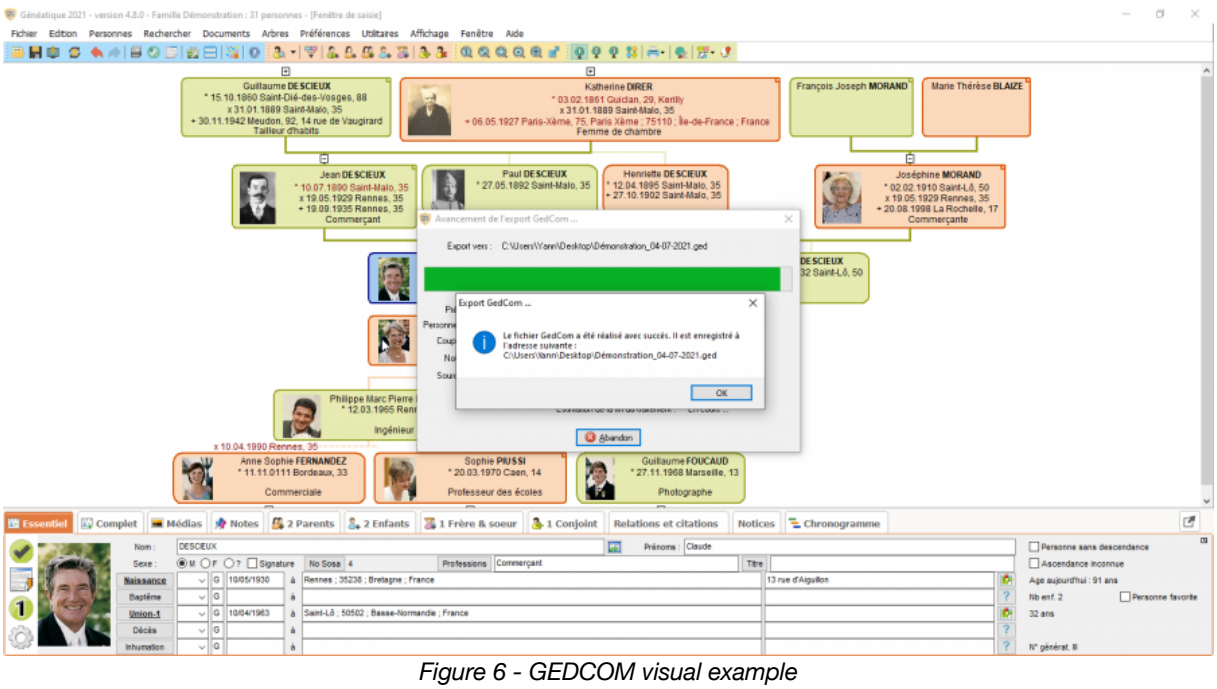


Figure 6 - GEDCOM visual example

6.3 Use case diagrams

6.3.1 Current system

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.

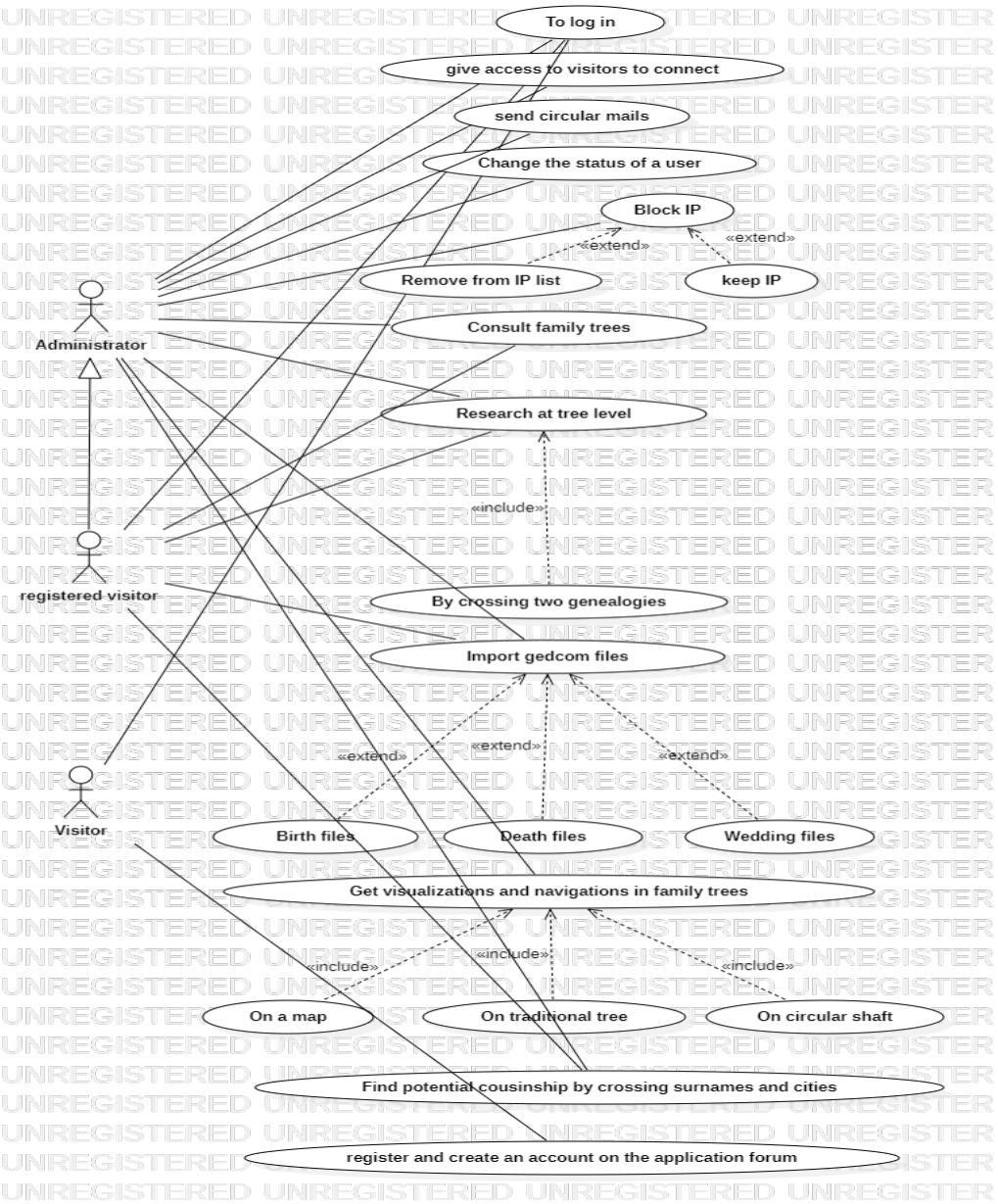


Figure 7- Current system (Use Case)

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 use of some of his exclusive functionalities while waiting for him or her to get back.

6.4 UI Models

6.4.1 Authentication page

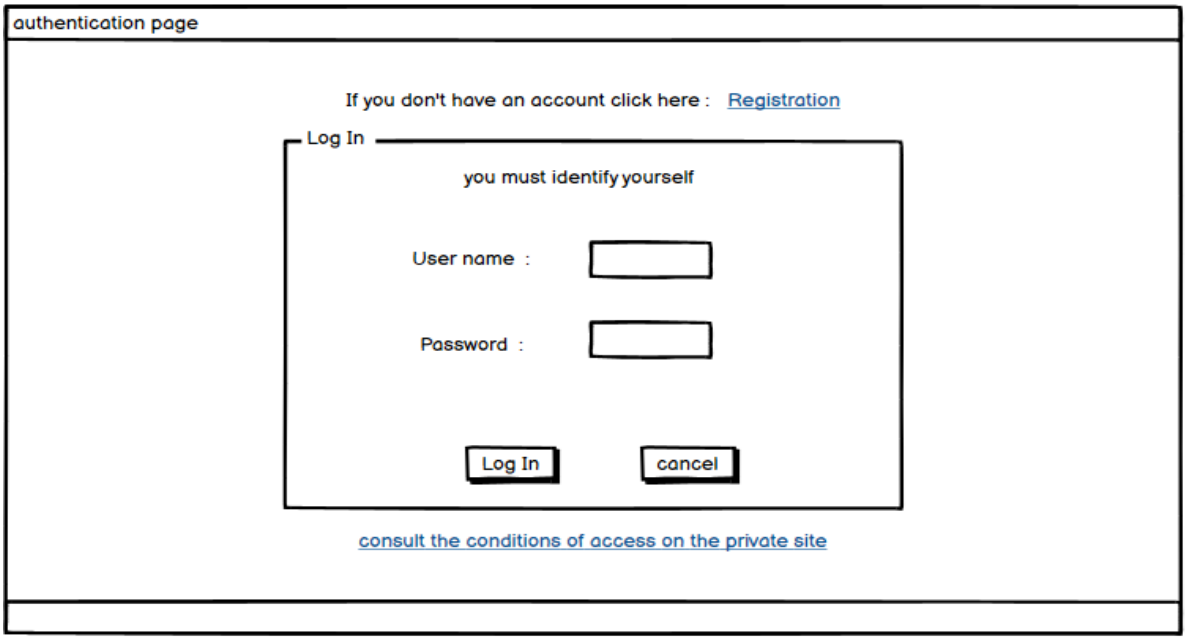


Figure 8 - Authentication page UI model

6.4.2 Registering page

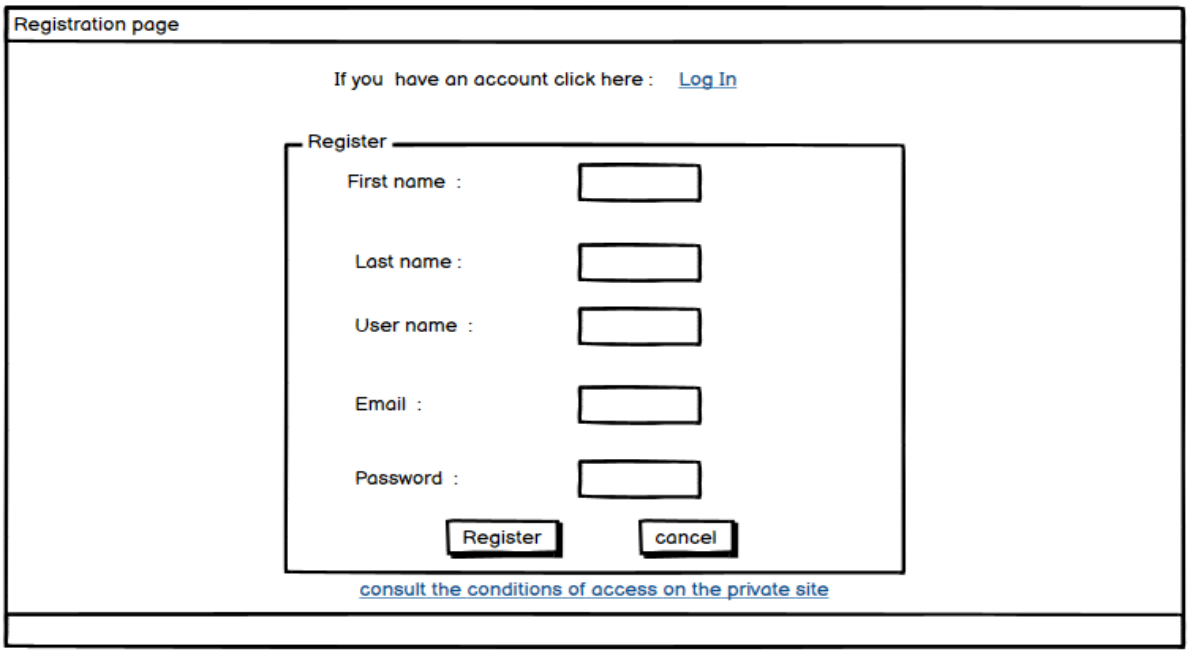


Figure 9 - Registration page UI model

6.4.3 Main page

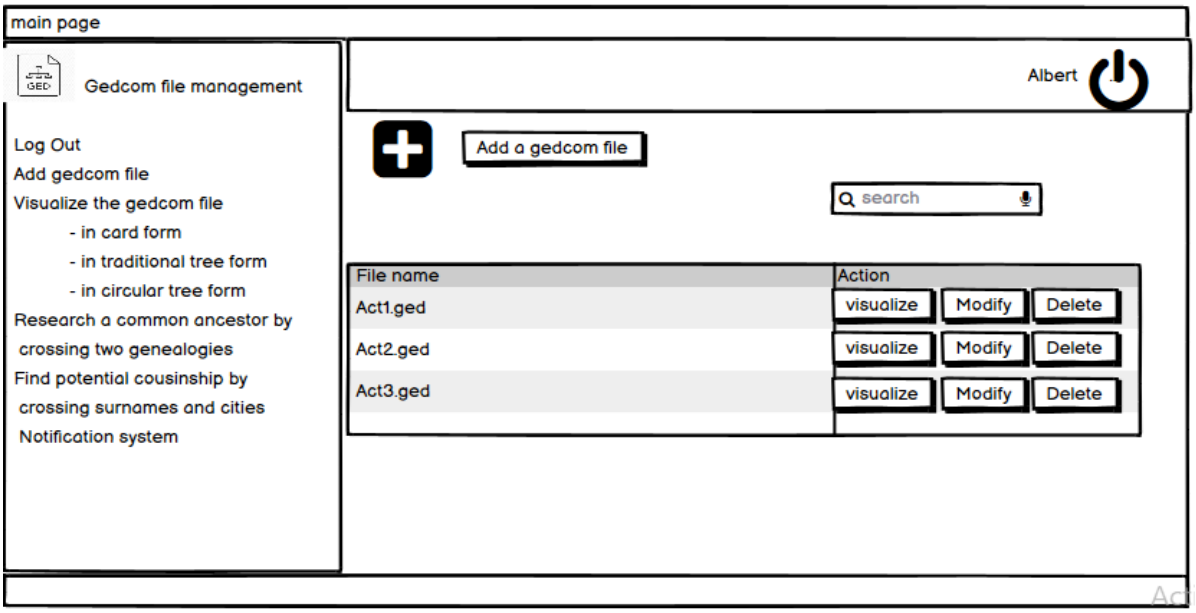


Figure 10 - Main page UI model

6.4.4 Add page

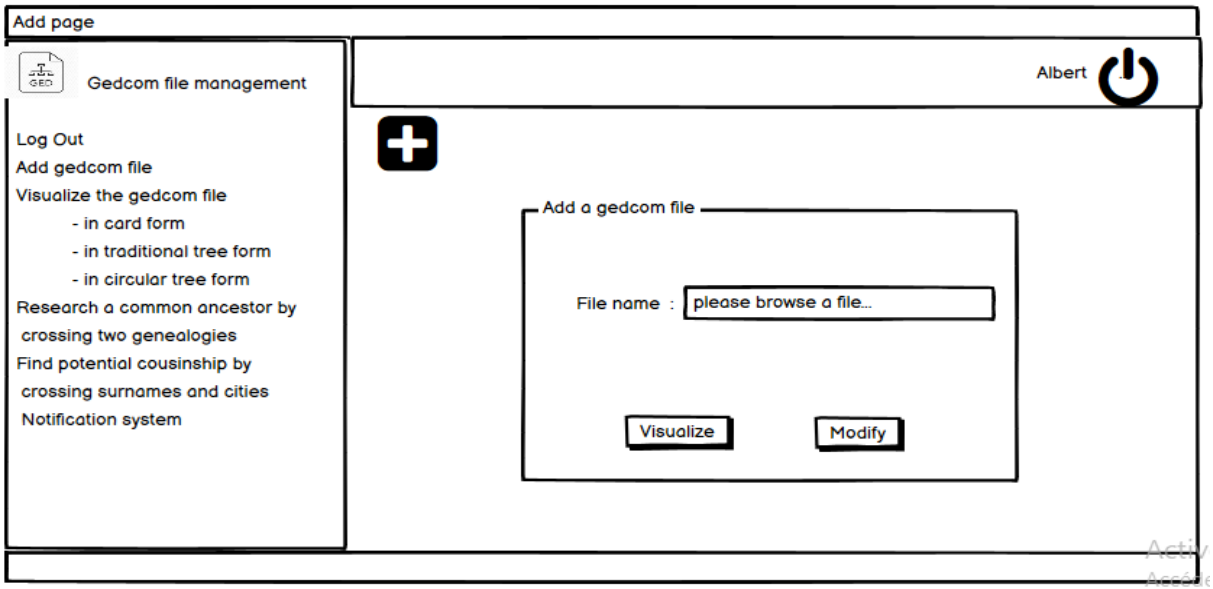


Figure 11 - Add page UI model

6.4.5 Default viewing page

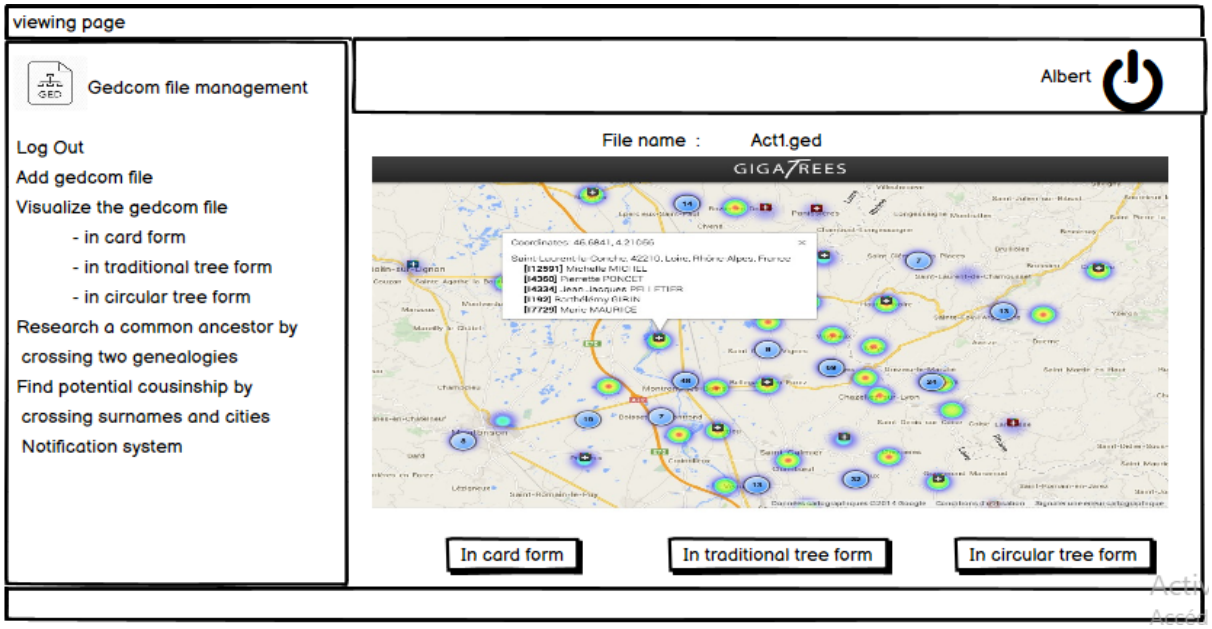


Figure 12 - Default viewing page UI model

6.4.6 Viewing page (traditional form)

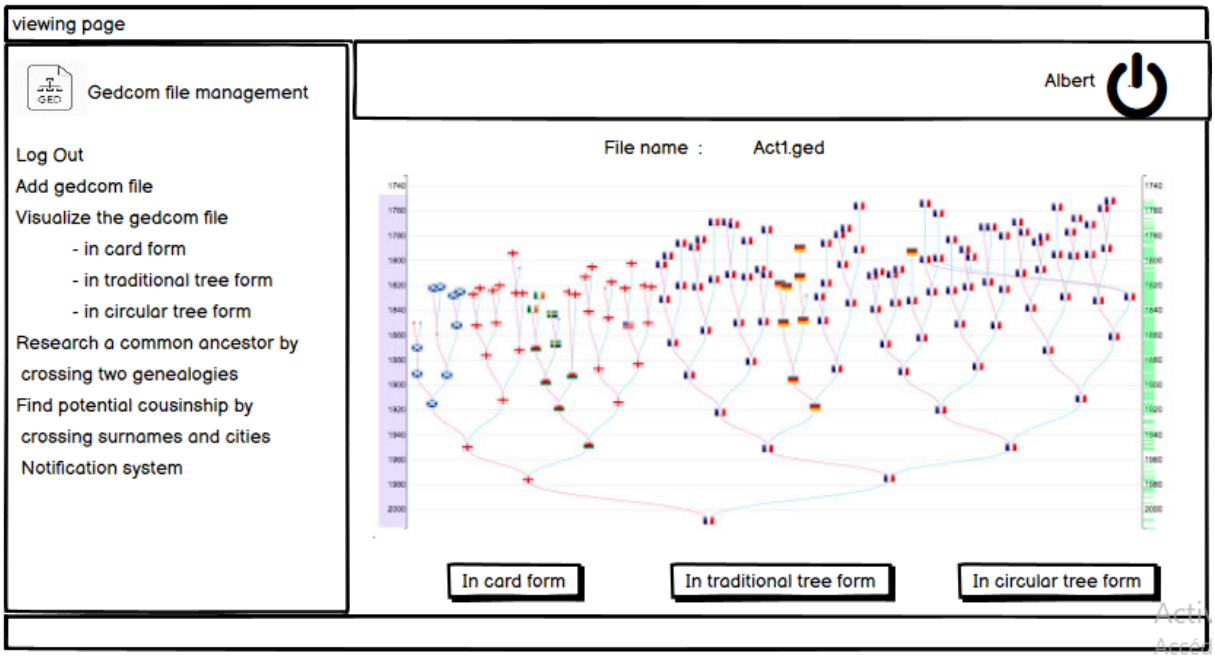


Figure 13 - Viewing page (traditional form) UI model

6.4.7 Viewing page (circular form)

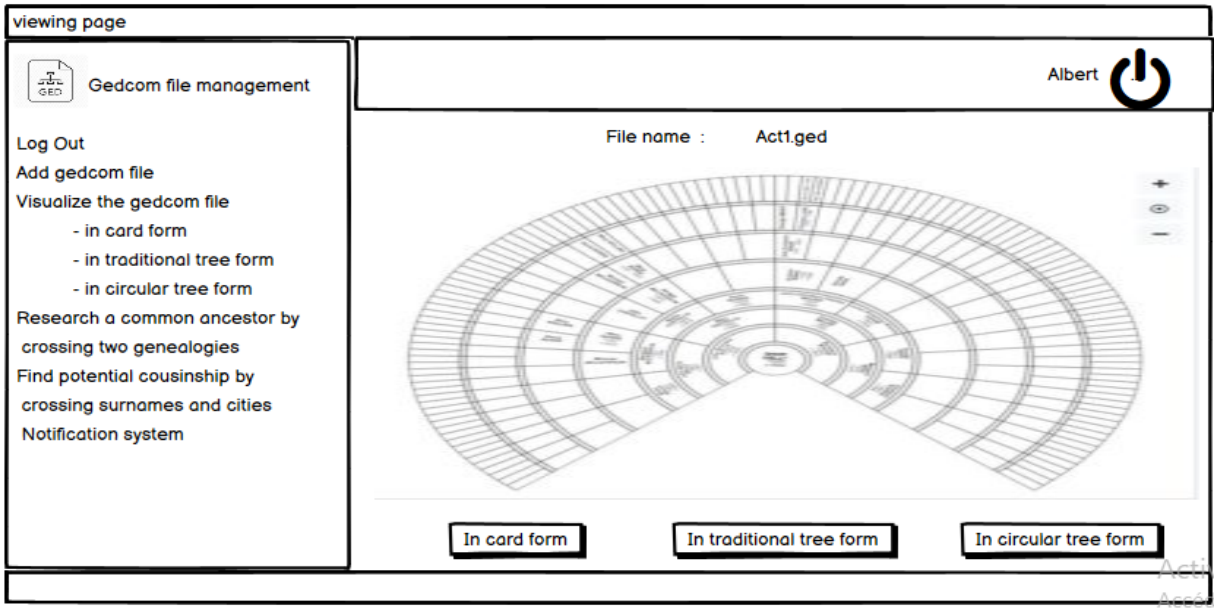


Figure 14 - Viewing page (circular form) UI model

6.4.8 Research by crossing 2 genealogies

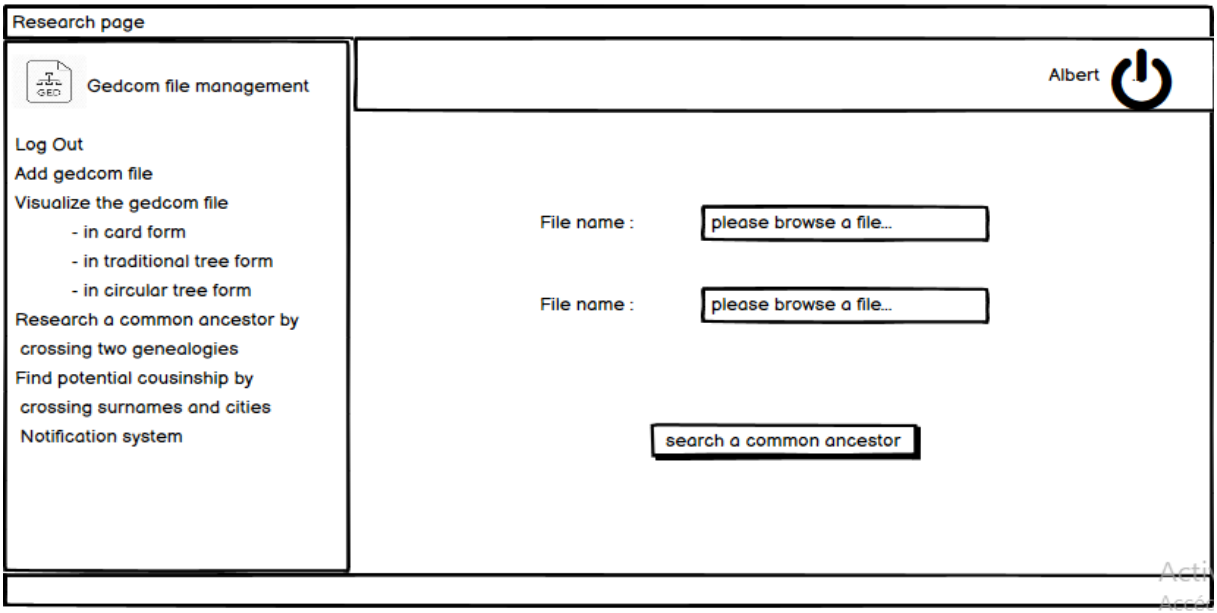


Figure 15 - Research by crossing 2 genealogies UI model

6.4.9 Research potential cousinhood

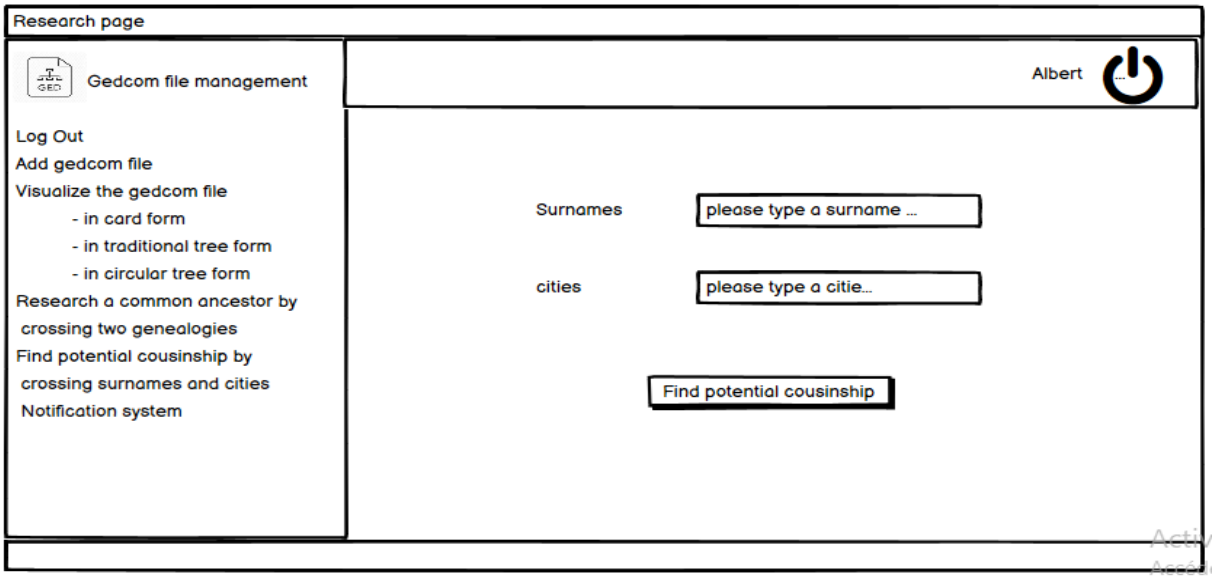


Figure 16 - Research potential cousinhood UI model

6.4.10 Notification page

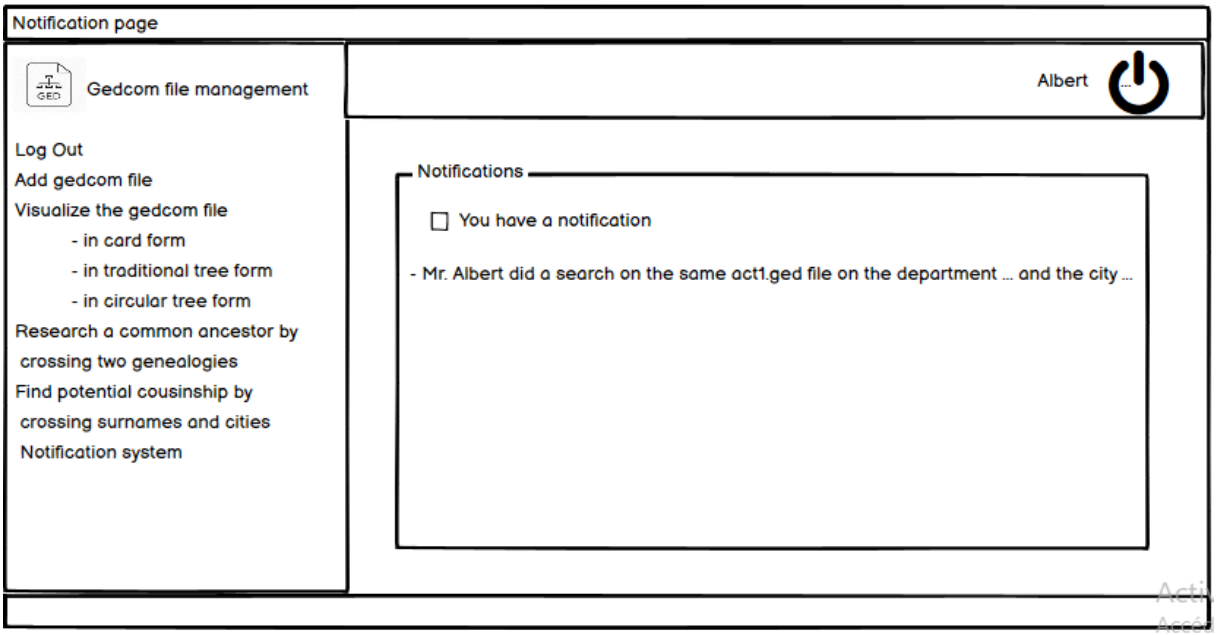


Figure 17 - Notification page UI model

6.5 Class diagrams

6.5.1 Initial class diagram

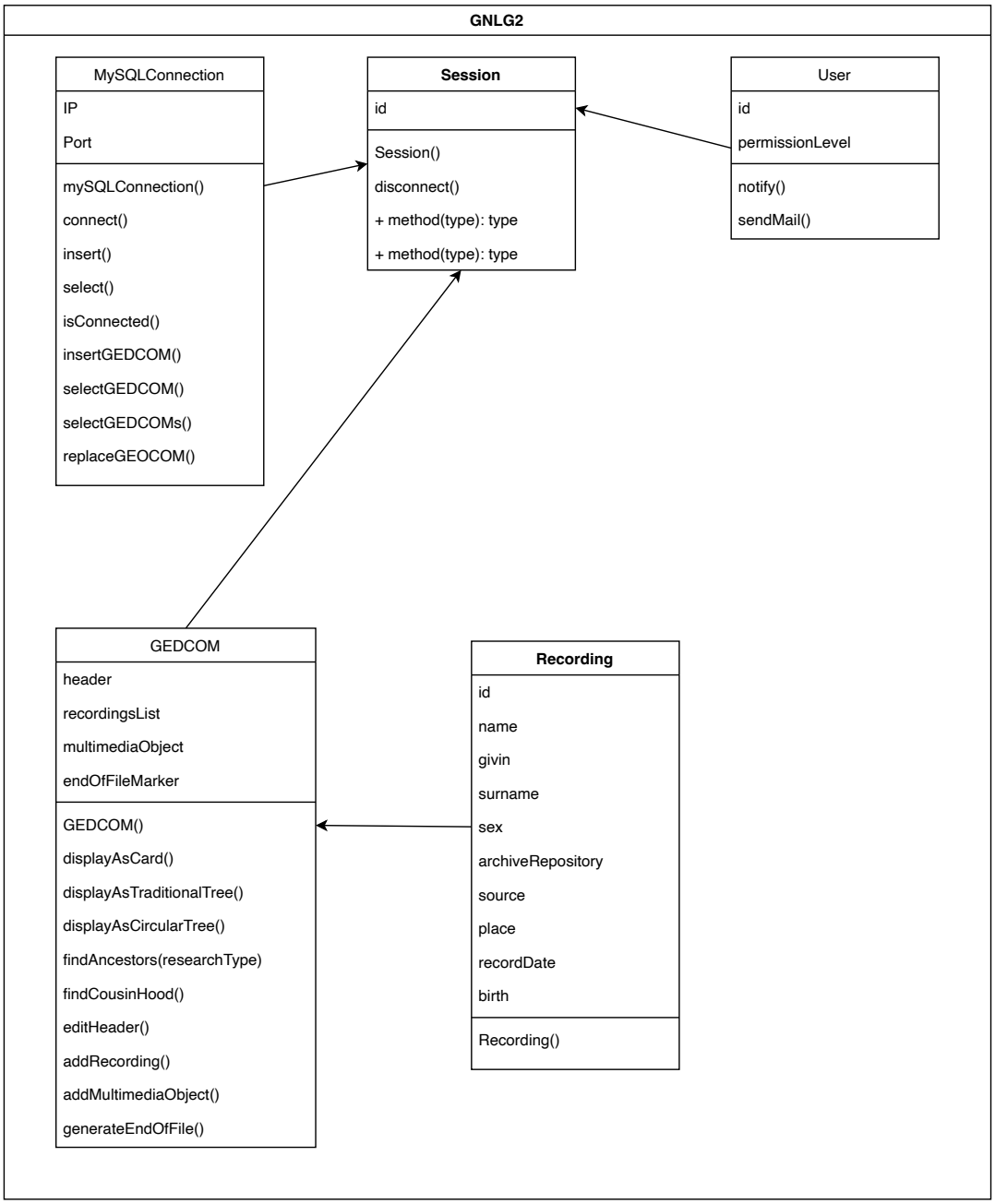


Figure 18 - Initial class diagram