

Historical Sea Level: an app for the study of global sea level changes

Mihir Odhavji
mihir.odhavji@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisbon, Portugal

December 2021

Abstract

There is a lot of data about mean sea level variation from studies conducted around the globe. This data is dispersed, lacks organization along with standardization, and in most cases, it is not available online. In some instances, when it is available, it is often in unpractical ways and different formats. Analyzing it would be inefficient and very time-consuming. In addition to all of that, to successfully process spatial-temporal data, the user has to be equipped with particular skills and tools used for geographic data like PostGIS, PostgreSQL and GeoAlchemy. The presented solution is to develop a web application that solves some of the issues faced by researchers. The web application allows the user to add data, be it through forms in a browser or automated with the help of an API. The application also assists with data querying, processing and visualization by making tables, showing maps and drawing graphs. Comparing data points from different areas and publications is also made possible. The implemented web application permits the query and storage of spatial-temporal data about mean sea level variation in a simplified, easily accessible and user-friendly manner. It will also allow the realization of more global studies. **Keywords:** Mean Sea Level Variation, Web Application, Geographical Database, Python

1. Introduction

Climate change, which can be defined as a modification of climate patterns, has caused worldwide effects owing to changes in weather and climate extremes, a rise in global surface temperature, glacier melting and shrinking along with worldwide mean sea level increase[39]. On the social sphere, these changes are a contributing factor to worsening inequality, economic losses due to a higher number of natural disasters, crop death and failure related to drought or out of season rains, which in turn inflate the food prices[13]. A rise in sea level is especially impacting due to its effects on coastal zones, where big and important cities are mostly located[31]. Nonetheless, sea level integrates the effects of diverse complex geological and climatological phenomena[12]. As an example, locally obtained sea level changes measurements can result from vertical land motion owing to either glacial isostatic adjustments or tectonics (for example, local glacier retreat generates a decreased ice burden - or weight - over landmasses, which in turn generates land uplift), which do not indicate an increase in absolute mean sea level[17]. Understanding the climate response at global, regional and local scales from various lines of evidence is crucial to estimate climate-related risks and adaptation[39]. Even though sea

level change represents one of the longest records using human instruments, going back 300 years in Europe, these are not enough to describe the non-anthropogenic related sea level increase of approximately 120m, which has occurred since the last glacial maximum (~19 000 years)[30, 17]. Therefore, the study of sea level changes is based on geological proxies that show a past position of sea level. Sea-level index indicators include instrumental evidence (for example from tidal gauges), ancient shorelines (such as raised beaches), biological indicators (such as bivalve shells), intertidal deposits (such as coastal marshes), and archaeological evidence (for example the position of ancient harbor and port facilities)[16].

With an increasing number of studies conducted and articles published, the amount of data available about sea level variation is also growing. However, data was saved on physical paper, which is easily corruptible, difficult and slow to share at a global level, can amount to huge stacks very quickly leading places to run out of space. Nonetheless, with computers becoming more mainstream, some of the problems were solved with excel sheets. However, excel comes with its problems like not being able to create maps and difficulty with collaboration. Every time someone wants to participate, a link has

to be shared and they have to be added to the excel sheet and that is not scalable. The file could also be made public to solve the scalability issues, but that is not secure enough. Another argument against excel is the fact that it is not a relational database. If everything is on the same sheet, it can become messy very quickly, if the data is in different sheets then, firstly it can be easy to lose track of the parameters, secondly, it can not have relations between tables, and thirdly queries cannot be made. A further problem is the fact that even assuming such a database exists, it is not online. By having it public the database can increase in size in a manner that a private one could never. Also, more people visualizing the Database means more people looking for errors and easily correcting them if needed. The problem is that there isn't a single database that combines all the data from all the studies conducted.

The objective is to develop a web tool that aggregates the maximum number of data points taken from local researchers about the variation of the average sea levels and helps the study of changes in mean sea level by using data from the holes dug. A tool that lets researchers collaborate without creating unnecessary friction (such as sharing links and files). An instrument that shows and creates dynamic maps and graphs, allows for some data query and exploration along with letting users compare the variation of the average sea level from different parts of the world. The tool should also facilitate the data entries made by the researchers, help them collaborate more smoothly and do some of the data standardization required for a better comprehension of the data available.

The expected results from this thesis are the creation of a web application that uses a relational database, a georeferenced database and allows for the creation of dynamic maps and graphs. The relational database saves observations that are represented by their geographic coordinates in a georeferenced database, the age, the area, the title of the paper on which they were published and the sea level. Using these data points, the application creates maps and graphs where it is possible to compare different places for the variation of the mean sea level.

2. Background

2.1. Building and Interpreting Sea level Curves

Sea level index points represent the position of past sea levels and mainly comprise biological, sedimentological, morphological, or archaeological evidence of past sea levels. Each type of sea level indicator is associated with a precision (assumed as an error interval) in its relationship with the mean sea level. For example, specific foraminiferal as-

semblages (foraminifera are single cell organisms with particular external shells) are nowadays found in high marsh environments, living only between the highest astronomical tide and the mean high water-level [10]. So, to a great degree, by finding these shells at lower depths when studying sediment cores, researchers find evidence of shallow water conditions, within a specific height range relative to the existing mean sea level. The next step is to estimate the age of the sediment and/or shells with the appropriate age estimation methods, such as radiocarbon dating, and a local sea level index point can be established.

Several ages estimation methods are often used in the establishment of sea level index points, largely depending on the indicator and time frame under analysis. Each method has its precision, mainly related to the statistical uncertainty obtained from the physical or chemical analysis used in the age determination [37, 20]. Due to this reason, age estimation of either sediment, biological, or rock samples, is always accompanied by an error.

Sea level curves are normally represented by scatter plots showing the relative sea level plotted against age, also named sea level index points, at a local scale. The consideration of precision/errors associated with the selected sea level indicator, and region (due to different tidal ranges, for example), combined with age estimation uncertainties, which depend on the technique used, are extremely important when comparing sea level index points from different locations or ages, and are commonly represented by either polygons or error bars [37, 18].

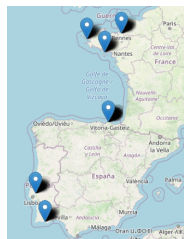
Sea level curves can differ considerably in different locations, as they represent the local relative sea level (height of the sea surface relative to the earth surface), mostly due to also local vertical land movement [10]. This land movement is related to several regional processes, including present and past changes and land ice mass, with major contributions from surface loading response to past glacial-isostatic adjustments, plate tectonics, as well as local processes, such as sediment compaction and past tidal range changes [10]. In fact, Holocene sea level databases are frequently used to constrain parameters in Glacial isostatic adjustment models, with important applications in the current understanding of sea level rise and its geographical variability [10, 17]. Quantification of relative and absolute sea level changes, the latter also named global mean sea level rise, is in continuous development due to the increasing amount of sea level data, and vertical land movements measured using geophysics and geodesy equipment [32]. Taking all these complex inter-related processes, the development of standardized Holocene sea level and vertical land-movement databases is pivotal to es-

tablish the global mean sea level rise and to have a better understanding of its changes and relationship with anthropogenic forcing.

2.2. Data Representation Libraries

The web application that is being planned to be developed has a server side and a client side. For the client side, HTML[38] is being considered, as it is easy to understand, it has a lot of documentation, forums and tutorials online, it can be embedded with other languages such as JavaScript[6] and with some basic knowledge, is fairly easy to create forms and tables for data collection and display. Other than forms and tables, the application should show personalized maps and plot graphs. However is not possible with HTML and JavaScript alone to show personalized maps and plot graphs, so alternatives were contemplated.

One of the requirements for the application is to show maps from a specific area or publication. These maps shown should have a pin for each of the observations on that specific area or publication. The coordinates of every observation are sent to the browser via a REST/AJAX[9, 22] request. Considering all the prerequisites, two alternatives were shortlisted: Google Maps[11] and Leaflet[36]. In figure 1, an example of a map shown to the user is provided, along with the JSON[15] received by the browser. The library, in this case, should use the data obtained to pin the points on the map.



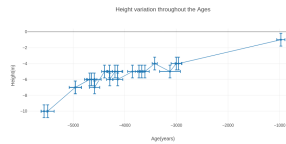
(a) How coordinates are shown on a map.

```
{
  "CoordinatesLat": 48.682179, "CoordinatesLong": -1.728949
},
{
  "CoordinatesLat": 48.272317, "CoordinatesLong": -4.665832
},
{
  "CoordinatesLat": 47.532456, "CoordinatesLong": -3.118968
},
{
  "CoordinatesLat": 43.449233, "CoordinatesLong": -2.815113
},
{
  "CoordinatesLat": 38.946335, "CoordinatesLong": -8.923438
},
{
  "CoordinatesLat": 37.075348, "CoordinatesLong": -8.129479
},
{
  "CoordinatesLat": 48.272317, "CoordinatesLong": -4.665832
}
```

(b) Example of a JSON received by the browser.

Figure 1: Example of a map with coordinates of observations points and the code to do it.

Another requirement for the application was to plot graphs for the sea level variation for a specific area and have the option to compare two or more areas. With these conditions in mind, two alternatives were finalized: Chart.js[4] and Plotly[26]. In figure 2, an example of a graph shown to the user is provided, along with the JSON[15] received by the browser. The library, in this case, utilizes the data received to draw the graphs and the respective vertical and horizontal error bars.



(a) How graphs are shown.

```
{
  "Height": { "value": -4, "error": 0.5},
  "Age": { "min": 1500, "max": 1750},
  "Name": "Htchay, Portugal"
},
{
  "Height": { "value": -6, "error": 1.5},
  "Age": { "min": 1800, "max": 1900},
  "Name": "Htchay, Portugal"
}
```

(b) Example of a JSON received by the browser.

Figure 2: Example of a graph plotted with error bars and the code to do it.

2.3. Geographical Databases

Being a project closely related to geology and geography, it was decided to look for tools that combine these fields with servers and Databases. ArcGIS[8], GeoServer[21], MySQL[23] and PostgreSQL[35] were found. ArcGIS is a platform to generate, organize, distribute and inspect spatial data. It can be run locally or on cloud services. GeoServer is a combination of a Web server with a Web Gis software and a database. It is open-source and allows users to share, manage and organize geospatial data. MySQL is a relational database management system. MySQL spatial extensions permit the users with the creation, storage and inquiry of spatial data through spatial data types and functions[24]. PostgreSQL is an open source object-relational database system that is more closely related to SQL, however, to use geographical libraries an extension named PostGIS[27] has to be added to the database. A further advantage of PostGIS is its capacity for doing spatial queries. Using PostGIS you can create a variety of geographical data types like Points, Line, Circles, Polygons, etc.[2]. These data types can interact with each other to make queries such as the meeting point of two lines or the intersection of a line with a shape[28]. This functionality is not only limited to geometry entities, it can be done with geographical bodies as well, namely lakes, rivers mountains, etc.

2.4. ORM

The browser can create queries for a database that is connected to the server. A system is essential to translate the server's request to the database. That is where ORM enters. Object-relational mapping (ORM)[1] is a technology that lets the user add, query, change and delete data from a database employing an object-oriented paradigm. The ORM library is an entirely normal library written in the same language as the rest of the code. The big advantage of using ORM is that it encapsulates the code required to manipulate the data, so SQL is not used. Direct interaction is done with the objects that use the same language as you. SQLAlchemy allows the use of various database management systems with just a configuration change. For the server, Python is used. Hav-

ing that in mind, options for ORM were searched, and two of them were shortlisted: SQLAlchemy[33] and Django[5]. The main difference between using SQL and SQLAlchemy is that, while using the former generic table rows are received, but using SQLAlchemy a list with objects is returned, which can be immediately used for further development.

3. Implementation

3.1. Requirements

Before starting the implementation, it is important to outline the requirements of the program that will be developed :

- The application should be remotely accessible.
- The application should store the data generated by the researcher about proxies for the variation of sea levels (proxy data).
- The proxy data about sea level variation should be organized in areas.
- The application should store the publications where the data was published.
- The application should store information about the indicator used to estimate the age of the sample.
- Each data point should include the following information: coordinates, estimated age and height.
- The application should present the data in maps (global with all the data and particular to a publication).
- The application should create plots with the representation of the data for a specific area.
- The plots should contain error bars for the age and height estimations.
- The user should be allowed to compare different areas in the same plot.
- The user should be able to download the data in tabular format.
- The application should contain local information about the vertical land movement.
- The application should provide forms for the introduction of data to the database.

3.2. Data Model

In order to solve the problems explained, it was decided to build a relational database. The final result is illustrated in figure 3.

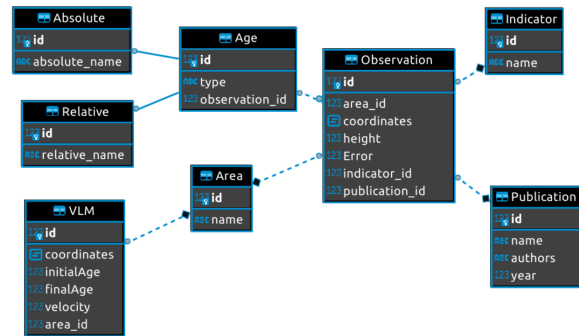


Figure 3: Unified Modelling Language of the data structures.

As seen in figure 3, there are many tables that save different data proxies. Every table has a primary key and its own ID.

The tables are:

- **Area** - a table that saves the area name and an ID number that is unique to each area.
- **Indicator** - a table that saves the indicator name and an ID number that is unique to each Indicator.
- **Publication** - a table that saves the publication name, its authors and the year on which it was published. It also has a unique ID number for each row.
- **Vertical Land Movement or VLM** - a table that saves the coordinates of the point, the initial and final ages, the velocity in mm/yr, the ID of the area that is related to and a unique ID for each of its own entries.
- **Observation** - a table that saves the coordinates and height of the sample, the error (in meters) associated with the Indicator, the ID of the area and publication and a unique ID for each observation.
- **Age** - a table that saves the ID of the observation that it is related to, the type of age (Absolute or Relative) and a unique ID for each age. The Absolute and Relative ages table inherit the ID from its parent table age[34].

3.3. Architecture

The three main components of the program developed are:

- The browser is used as the client of the server. It is where the HTML[38] presents forms to introduce data along with showing tables, graphs and maps. The browser has the succeeding modules:

- **Add Data** - A module for adding data to the database.
 - **View Map** - A module for viewing maps with the coordinates sent by the server.
 - **Plot Graph** - A module for plotting graphs with the information received from the server.
 - **Download CSV** - It is a module that transforms data listed on the web page to a CSV type file[3]
- The server receives requests sent by the client(browser) and responds to them. To successfully respond to them, it needs to convert the requests to queries for the database. To do that translation, it uses ORM[1]. After receiving the result from the database, it sends the output to the browser to show it to the user. The server is comprised of the following modules :
 - **Add Data** - A module for adding data to the database.
 - **List Data** - A module to list data from the database
 - **Vertical Land Movement** - A module that stores the data related to vertical land movement.
 - **View Map** - A module that plots the observations' coordinates on a flat 2D map.
 - **Plot Graph** - A module that draws the graphs for sea level variation throughout the ages.
 - The database receives queries from the server and it responds to them. It has to correctly save geographical data such as the point coordinates.

3.4. Support System

To successfully implement this web application, it was important to plan in which language to code the three main modules of the program. The browser is mainly coded in HTML and JavaScript[6], the server in Python[29], using the Flask framework[25], and the Database in PostgreSQL[35].

Starting from the browser, HTML is used to create the web pages, to receive input from the users and to show them the output. JavaScript is utilized to create requests that are sent to the Flask server. The requests are read by the HTML from the user, built by JavaScript and sent to the server with help of REST and AJAX[9, 22]. Afterward, the JavaScript receives the response and displays it using to the user using HTML. To produce the maps, JavaScript's Leaflet[36] library is used. Leaflet is

used because of how coding friendly and well documented it is. For plotting the graphs it is used the Plotly[26] library. Plotly's easiness and lack of friction in processes such as adding data to the plot was the key factor, so it was decided to go ahead with Plotly. Additional advantages of using Plotly are the fact that the user can personalize the zoom settings as well as download the graph plotted to the computer directly, without having to take screenshots.

Relative to the server, it was coded in python using the Flask framework, mainly due to its simplicity and velocity. The server has the responsibility to receive requests sent by the browser and give them to the ORM with to query the database. After trying Django[5] and SQLAlchemy[33], SQLAlchemy was elected to be the bridge between server and database because the author of this thesis has some experience with it from a previous project. As the database has geographical elements, the GEOAlchemy[7] library is used to add support for spatial data types. Using SQLAlchemy and GeoAlchemy libraries, the first is used to translate the queries received by the server to SQL[14] and the latter to assist the first regarding spatial queries. With GeoAlchemy it is also possible to add certain geographical data types such as 'POINT' to represent the latitude and longitude of an observation.

At last, the Database selected is PostgreSQL. It is an advanced form of SQL that allows the use of the normal SQL functions like query, primary keys, triggers, etc plus the use of Geospatial tools like GeoAlchemy and its Geometry library to enable the creation of POINT data type. PostGIS is also installed as an extender for PostgreSQL. Other options like MySQL[23] were rejected because GeoAlchemy only supports PostGIS, while GeoServer[21] and ArcGIS[8] were very complex to start programming.

3.5. End Points REST

For a secure and trustworthy connection between browser and server, it was decided to implement it using REST endpoints.

The REST Endpoints Implemented are the following :

1. **"/API/Area/"**, **'GET'** - Used to get the list of all areas.
2. **"/API/Pub/"**, **'GET'** - Utilized to get the list of all publications.
3. **"/API/Obs/"**, **'GET'** - It has the objective to get the list of all observations.
4. **"/API/Indicator/"**, **'GET'** - Implemented to get the list of all indicators.

5. `"/API/VLM/"`, 'GET' - Exercised to get the list of all vertical land movements.
6. `"/API/GetName/'`, 'POST' - It has the goal to get a name of area or publication.
7. `"/API/GetObservations/"`, 'POST' - Employed to get a list of observations based on ID.
8. `"/API/AddArea/"`, 'POST' - Request used to create a new area.
9. `"/API/AddPub/"`, 'POST' - Request utilized to create a new publication.
10. `"/API/AddObs/"`, 'POST' - Request sent to create a new observation.
11. `"/API/AddInd/"`, 'POST' - Request used to create a new indicator.
12. `"/API/AddVLM/"`, 'POST' - Request used to create a new vertical land movement.
13. `"/API/GetChart/"`, 'POST' - Request created to plot a graph.

The returned values are then utilized to create the output shown to the user on the browser such as tables, maps and Graphs.

3.6. Software Organization

With the browser and database already discussed, the focus of this section is on the server. The server is made of smaller modules. These modules are:

- **Add Data** - It is a module to add data to the database. It can be an area, publication, observation, indicator or vertical land movement. To help with the standardization while adding data, an algorithm was implemented that converts ages from the Before Present Scale to the AD/BC scale. It does so by taking 1950 and subtracting the age introduced in the Before Present Scale. It utilizes the following REST endpoints already described in section 3.5:

- `"/API/AddArea/"`, 'POST'
- `"/API/AddPub/"`, 'POST'
- `"/API/AddObs/"`, 'POST'
- `"/API/AddInd/"`, 'POST'
- `"/API/AddVLM/"`, 'POST'

- **List Data** - It is a module to list data from the database. It can be an area, publication, observation, indicator or vertical land movement. There is also the possibility to list observations for a particular area or publication. The module makes use of the upcoming REST endpoints already explained in section 3.5:

- `"/API/Area/"`, 'GET'
- `"/API/Pub/"`, 'GET'
- `"/API/Obs/"`, 'GET'
- `"/API/Indicator/"`, 'GET'
- `"/API/VLM/"`, 'GET'
- `"/API/GetName/'`, 'POST'
- `"/API/GetObservations/"`, 'POST'

- **View Map** - It is a module that plots the observations' coordinates on a flat 2D map. You can see all the points on the Database or filter for a particular area or publication. It uses the ensuing REST endpoints described in section 3.5:

- `"/API/Obs/"`, 'GET'
- `"/API/GetObservations/"`, 'POST'

- **Plot Graph** - It is a module that draws the graphs for sea level variation throughout the ages. You can see it for each area individually, see the graphs for all the areas simultaneously and able or disable certain areas. The module also draws the horizontal and vertical error bars for the graphs. It makes use of the following REST endpoint described in section 3.5:

- `"/API/GetChart/"`, 'POST'

- **Vertical Land Movement** - It is a module that stores data related to the vertical land movement and later can be used for rectifying the data introduced on the database by applying a correcting factor related to tectonic movement and geographic location. It does not use any of the REST endpoints explained in section 3.5.

4. Results

This section presents how the requirements were fulfilled in the project.

4.1. The application should be remotely accessible

The creation of a web application satisfies this requirement. The historical sea level application runs on a browser, it is connected to a server that communicates to a database using an ORM. An API is also provided to send requests to the server and received information. It is also possible to develop other web applications that connect to the server and show data differently.

4.2. The application should store the data generated by the researcher about proxies for the variation of sea levels (proxy data)

The creation of a database with the model given in figure 3, utilized to store the different data types related to an observation meets this requirement. The

data is not only stored on a georeferenced database but there is also the possibility of doing spatial queries. This can be a foundation, where future works could be built.

4.3. The proxy data about sea level variation should be organized in areas

According to the UML provided in figure 3, a table with the name Area is created, that saves the name of the area and a unique ID to identify it. This table is related to the Observation table as well as the Vertical Land Movement table. Therefore, the proxy data is organized in areas. The papers, from where the data points are taken, organize them in areas as well. Those areas are the ones that later are plotted in the graphs.

4.4. The application should store the publications where the data was published

As stated by the UML given in figure 3, a table with the name Publication is created, that saves the name of the publication, the name of its' authors, the year it was published and possesses a unique ID to identify it.

4.5. The application should store information about the indicator used to estimate the age of the sample

According to the UML given in figure 3, a table with the name Indicator is created, that saves the name of the indicator along with a unique ID to identify it. This table is related to the Observation table as well.

4.6. Each data point should include the following information: coordinates, estimated age and height

As stated by the UML provided in figure 3, a table with the name Observation is created, that saves the coordinates and the height they were found. It is related to the Age table, which saves the Age type(Absolute or Relative) as well as the value.

4.7. The application should present the data in maps (global with all the data and particular to a publication)

The application can show a list of all the publications in the database. By selecting one, it is possible to see a map with the coordinates points of observations taken from that publication. The application also has a tab called Map to see the location of all observations on the database.

4.8. The application should create plots with the representation of the data for a specific area

The application gives the possibility to see a list of all areas in the database. By selecting one, it is feasible to spot a graph with the data point about that area. An example is given in figure 4.

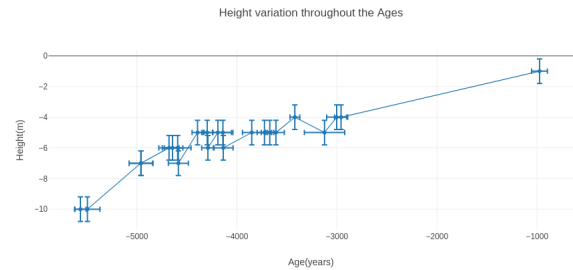


Figure 4: Graph plotted with data from only one area.

4.9. The plots should contain error bars for the age and height estimations

As seen in figure 4, the plots have error bars. The vertical error bars are related to the height estimations, while the horizontal bars are related to the age estimation. Compared with the graph from [19], it can be concluded that the graphs plotted by the application are similar to the graphs drawn on the research papers.

4.10. The user should be allowed to compare different areas in the same plot

The application has the option to see the graphs of all areas in the same plot. On that plot, there is a choice to hide and unhide certain areas.

4.11. The user should be able to download the data in tabular format

To download the data from the database to your device, at the end of every list on the web application a download button is placed. By clicking on it, the data from the table that the user is consulting gets written to a CSV file[3] and downloaded.

As an example, if the table with the list of observations for a particular area was to be transformed and downloaded, the resulting CSV file would have the following columns : ID, Coordinates, Height, Age, Indicator and Error. The data in the downloaded CSV file can be processed and used with other tools or applications.

4.12. The application should contain local information about the vertical land movement

As claimed by the UML provided in figure 3, a table with the name Vertical Land Movement is created, that saves the coordinates of the point, the initial and final Ages, the velocity and has a unique ID to identify it. This table is related to the area table as well. Thus, the application contains local information about vertical land movement. Besides the fact that data about vertical land movement is only stored and not processed, its' existence in the database allows for the further development and implementation of a simpler model. This tool will be useful for future work about the relation between

vertical land movement and mean sea level variation because it is possible to realize studies at a small and local scale along with at a larger and global scale.

4.13. The application should provide forms for the introduction of data to the database

The application provides forms for the user to introduce data into the database. The form seen in figure 5 is for adding observation to the database. Similar forms are available to introduce the different data types: Area, Publication, Indicator and Vertical Land Movement.

The input numbers in the figure 5 are:

1 - To add the Latitude coordinate. It has to be between -90 and 90 degrees.

2 - To put the Longitude coordinate. It has to be between -180 and 180 degrees.

3 - To introduce the altitude value.

4 - To write the error value(in meters).

5 - To choose the area related to the observation.

6 - To select the indicator used to estimate the observation's age.

7 - To pick the publication on which the observation was published.

8 - For choosing the age type - absolute or relative. If absolute is selected, the text box on number eleven is hidden, and only the text box on number ten is used to input the age value.

9 - If the Before Present option is selected, it automatically converts the Age values to the AD/BC scale.

10 - To input the Upper Limit in case the Age type is relative, in case the age type is absolute, this is the only text box used to input the value.

11 - To input the Lower Limit in case the Age type is relative, if it is absolute, then this text box is hidden.

12 - Button used to submit data into the database.

Add a new Observation

Figure 5: Form utilized to introduce an observation into the database.

To ease the navigation between pages of the web application, a menu was created that is visible on all the pages. The menu has seven options :

- **Home** - Brings the user back to the homepage.
- **Area** - To add or list all areas.
- **Publication** - To add or list all publications.
- **Observation** - To add or list all observations.
- **Indicator** - To add or list all indicators.
- **Map** - To see a map with all the different coordinates of the observations stored in the database.
- **Graphs** - To see a graph with the mean sea level variation of all the areas stored in the database.

While hovering above some of the options, a drop down menu appears with two options: add or list all.

5. Conclusions

Until now, there was not a platform that aggregated spatial information and allowed at a global scale the processing of data related to mean sea level variation and vertical land movement. With this thesis, a web application was created that aggregates data from numerous publications and makes them available in an organized and researcher friendly manner. The application can be accessed by anyone from a browser. A REST API was also developed, so requests can also be automated. The user can also add data to the database in a very simple and frictionless way. The data added is standardized, especially the age value can be converted from

Before Present to the AD/BC scale. Data about the vertical land movement is also stored on the database and can be used for further work. The developed application lists the data in a clean and organized manner. That data can be viewed online or downloaded to a CSV file for other works. The developed application also has the possibility of viewing maps and plotting graphs. Additionally, there is the possibility of data comparison. The plots about the sea level variation throughout the ages can have graphs from more than one area enabling, this way, the comparison between two or more areas. This project, for the first time, allows the creation of interactive graphs (with features like zoom in/out and selecting which areas to hide or show) about the mean sea level variation. This new way of data presentation not only permits, easy comparison between distinct areas but also for the availability of this content on the web, to increase the visibility of such problems. The graphs plotted are a tool used by specialists, who try to understand what kind of events occurred in those areas. However, these types of graphs help identify the time intervals on which the events might have happened. The current version only stores the values related to the vertical land movement. It is also the first time, that it was possible to aggregate on one platform these types of data at a global level. This will allow scientists and data modelers, the development of new theories and/or models to explain the differences between the various areas. Using the spatial queries abilities of PostGIS, combined with the data available on the database about the vertical land movement, mainly the geographic coordinates, a model can be developed that automatically suggests corrections based on the location of the observations.

References

- [1] Object-relational mapping(ORM). https://en.wikipedia.org/wiki/Object%E2%80%93relational_mapping. (accessed: 30.10.2021).
- [2] PostGIS: Data Management and Queries. <https://postgis.net/docs/manual-1.4/ch04.html>.
- [3] Y. Shafranovich. Comma-separated values. <https://datatracker.ietf.org/doc/html/rfc4180>, 2005. (accessed: 30.10.2021).
- [4] Chart.js contributors. Chart.js. <https://www.chartjs.org/>, 2021. (accessed: 30.10.2021).
- [5] Django Software Foundation. Django. <https://www.djangoproject.com/>, 2021. (accessed: 30.10.2021).
- [6] ECMA International. JavaScript. <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>, 2021. (accessed: 30.10.2021).
- [7] Eric Lemoine. GeoALchemy. <https://geoalchemy-2.readthedocs.io/en/latest/>, 2012. (accessed: 30.10.2021).
- [8] ESRI. ArcGIS. <https://www.esri.com/en-us/arcgis/products/arcgis-desktop/resources>, 2021. (accessed: 30.10.2021).
- [9] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, 2000.
- [10] A. García-Artola, P. Stéphan, A. Cearreta, R. E. Kopp, N. S. Khan, and B. P. Horton. Holocene sea-level database from the atlantic coast of europe. *Quaternary Science Reviews*, 196:177–192, 2018. <https://doi.org/10.1016/j.quascirev.2018.07.031>.
- [11] Google. Google Maps. <https://developers.google.com/maps>, 2021. (accessed: 30.10.2021).
- [12] V. Gornitz. Sea level change, post-glacial. *Encyclopedia of Paleoclimatology and Ancient Environments*, pages 887–893, 2009. https://doi.org/10.1007/978-1-4020-4411-3_207.
- [13] S. Hallegatte. *Shock waves: managing the impacts of climate change on poverty*. World Bank Publications, 2016. <https://openknowledge.worldbank.org/handle/10986/22787>.
- [14] ISO/IEC. SQL. <https://www.iso.org/standard/63555.html/>, 2016. (accessed: 30.10.2021).
- [15] JSON.org. JSON. <https://www.json.org>, 2002. (accessed: 30.10.2021).
- [16] M. S. Kearney. Sea level indicators. *Encyclopedia of Paleoclimatology and Ancient Environments*, pages 899–902, 2009. https://doi.org/10.1007/978-1-4020-4411-3_209.
- [17] N. S. Khan, B. P. Horton, S. Engelhart, A. Rovere, M. Vacchi, E. L. Ashe, T. E. Törnqvist, A. Dutton, M. P. Hijma, and I. Shennan. Inception of a global atlas of sea levels since the last glacial maximum. *Quaternary Science Reviews*, 220:359–371, 2019. <https://doi.org/10.1016/j.quascirev.2019.07.016>.

- [18] K. Lambeck, C. D. Woodroffe, F. Antonioli, M. Anzidei, W. R. Gehrels, J. Laborel, and A. J. Wright. Paleoenvironmental records, geophysical modeling, and reconstruction of sea-level trends and variability on centennial and longer timescales. *Understanding Sea-Level Rise and Variability*, pages 61–121, 2010. <https://doi.org/10.1002/9781444323276.ch4>.
- [19] E. Leorri, A. Cearreta, and G. Milne. Field observations and modelling of holocene sea-level changes in the southern bay of biscay: implication for understanding current rates of relative sea-level change and vertical land motion along the atlantic coast of sw europe. *Quaternary Science Reviews*, 42:59–73, 2012. <https://doi.org/10.1016/j.quascirev.2012.03.014>.
- [20] W. Li, X. Li, X. Mei, F. Zhang, J. Xu, C. Liu, C. Wei, and Q. Liu. A review of current and emerging approaches for quaternary marine sediment dating. *Science of The Total Environment*, 780:146522, 2021. <https://doi.org/10.1016/j.scitotenv.2021.146522>.
- [21] Open Source Geospatial Foundation. GeoServer. <http://geoserver.org/>, 2021. (accessed: 30.10.2021).
- [22] OpenJS Foundation and jQuery contributors. AJAX. <https://api.jquery.com/>, 2021. (accessed: 30.10.2021).
- [23] Oracle Corporation and/or its affiliates. MySQL. <https://www.mysql.com/>, 2021. (accessed: 30.10.2021).
- [24] Oracle Corporation and/or its affiliates. MySQL Spatial data types. <https://dev.mysql.com/doc/refman/8.0/en/spatial-types.html>, 2021. (accessed: 30.10.2021).
- [25] Pallets. Flask. <https://flask.palletsprojects.com/en/2.0.x/>, 2010. (accessed: 30.10.2021).
- [26] Plotly. Plotly.js. <https://plotly.com/javascript/>, 2021. (accessed: 30.10.2021).
- [27] PostGIS Project Steering Committee. PostGIS. <https://postgis.net/>, 2021. (accessed: 30.10.2021).
- [28] PostGIS Project Steering Committee. PostGIS: Building Applications. <https://postgis.net/docs/manual-1.4/ch05.html>, 2021. (accessed: 30.10.2021).
- [29] Python Software Foundation. Python. <https://www.python.org/>, 2021. (accessed: 30.10.2021).
- [30] M. Siddall, E. J. Rohling, A. Almogi-Labin, C. Hemleben, D. Meischner, I. Schmelzer, and D. Smeed. Sea-level fluctuations during the last glacial cycle. *Nature*, 423(6942):853–858, 2003. <https://doi.org/10.1038/nature01690>.
- [31] C. Small and R. J. Nicholls. A global analysis of human settlement in coastal zones. *Journal of coastal research*, pages 584–599, 2003. <https://www.jstor.org/stable/4299200>.
- [32] G. Spada. Glacial isostatic adjustment and contemporary sea level rise: An overview. *Surveys in Geophysics*, 38(1):153–185, 2017. <https://doi.org/10.1007/s10712-016-9379-x>.
- [33] SQLAlchemy authors and contributors. SQLAlchemy. <https://www.sqlalchemy.org/>.
- [34] SQLAlchemy authors and contributors. SQL Inheritance Hierarchies. <https://docs.sqlalchemy.org/en/14/orm/inheritance.html>, 2021. (accessed: 30.10.2021).
- [35] The PostgreSQL Global Development Group. PostgreSQL. <https://www.postgresql.org/>, 2021. (accessed: 30.10.2021).
- [36] Vladimir Agafonkin. Leaflet. <https://leafletjs.com/>, 2021. (accessed: 30.10.2021).
- [37] M. Walker. *Quaternary dating methods*. John Wiley and Sons, 2005.
- [38] WHATWG. HTML. <https://html.spec.whatwg.org/>, 2021. (accessed: 30.10.2021).
- [39] P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, et al. Climate change 2021: The physical science basis. *Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, 2021. https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_SPM.pdf.