Cairo University
Faculty of Computers and Artificial Intelligence
2020/2021

Graduation Project Documentation

Natural Disasters Statistics and Earthquake Shockwaves Prediction

Prepared by:

Abdelaziz Mohsen Abdelaziz	20170155
Marwan Ahmed Abdelmoneim	20170275
Mohamed Khaled Khedr	20170228
Seifeldin Mohamed Gamal	20150131
Youssef Hassan ElSayed	20170340

UNDER SUPERVISION OF:

ASSOCIATE PROFESSOR NEAMAT EL-TAZI

T.A. HODA TAWAKL

July 2021

Table of Contents

Chapter 1: Introduction

1. Background 2
2. Scope 2
3. Development Methodology 3
   3.1. Waterfall 3
   3.2. Phases of Waterfall Model 4
   3.3. Advantages and Disadvantages 5
4. Overall Description 5
   4.1. Project Perspective 5
   4.2. Project Objectives 6
   4.3. User Classes and Characteristics 6
   4.4. Operating Environment 6
   4.5. Design and Implementation Constraints 6

Chapter 2: System Analysis 8

1. External Interface Requirements 8
   1.1. User Interfaces 8
      1.1.1. Landing Page 8
      1.1.2. Interactive Map and Statistics 8
      1.1.3. Prediction View 9
      1.1.4. Analysis Process View 9
   1.2. Software Interfaces 10
   1.3. Communications Interfaces 10
2. System Features 10
   2.1. Interactive Natural Disasters Map 10
   2.2. View Natural Disasters Statistical Information and Graphs 10
   2.3. Predict Earthquake Resulting Shockwaves' Aspects 11
   2.4. View Analysis Process 11
3. Nonfunctional Requirements 11
   3.1. Availability 11
   3.2. Reliability 11
   3.3. Recoverability 11
   3.4. Scalability 12
   3.5. Maintainability 12
   3.6. Usability 12
   3.7. Security 12
4. Other Requirements 12
   4.1. Data management 13
   4.2. User Interface 13
5. Design Strategy 13
   5.1. Business Goals 13
   5.2. Users and Stakeholders 13

5.3. General Tasks 13
5.4. Technological Constraints 14
5.5. Branding Goals 14
6. User Analysis 14
6.1. User Groups 14
6.2. User Profile 14
6.3. Task Profile 14
6.4. Task Prioritization 15
6.5. Environmental Profile 16
6.6. Personas 16
7. Data Gathering 16
7.1. National Oceanic and Atmospheric Administration (NOAA) 17
7.2. United States Geological Survey (USGS) Earthquake Hazards 17
8. Data Analysis 17
8.1. Process Overview 17
8.2. Data Snapshots 17
8.3. Data Preprocessing 18
8.4. Insights and Conclusions 18
8.5. Graphs to be plotted and shown 19
9. Machine Learning and Prediction Model 20
9.1. What do we want to predict? 20
9.2. Which attributes should be used? 20
9.3. How will these attributes be used in the model? 20
9.4. Which machine learning models can be used? 20
9.5. Why do we choose regression models? 21
9.6. How will we apply the regression model? 21

Chapter 3: System Design 22

1. Use-Cases 23
1.1. Register 23
1.2. Login 23
1.3. View Map 24
1.4. View Statistical Information 24
1.5. Predict Shockwave 25
1.6. View Analysis Process 25
1.7. Information and Tips 26
2. Use-Case Diagram 27
3. Sequence Diagram 27
4. Class Diagram 28
5. Business Process Model and Notation (BPMN) 30

Chapter 4: Implementation 31

1. Application Backend 31
1.1. Backend Server 32

1.2. Database **32**
1.3. RESTful API **32**
2. Application Frontend **32**
2.1. Static HTML5, CSS3 and Basic Javascript **33**
2.2. React.js **33**
3. Authentication **33**
4. Application Snapshots **33**
4.1. User Signup/Login **34**
4.2. Home Page **35**
4.3. Statistics Page **36**
4.4. Prediction Page **37**
4.5. Analysis Page **37**
5. Testing **37**
5.1. Unit Testing **38**
5.2. System Testing **38**

Chapter 5: Results and Conclusion 40

1. Evaluation of Data Analysis Process **40**
1.1. USGS Dataset **40**
1.2. NOAA Dataset **41**
2. Visualizations **41**
3. Conclusion **47**

Appendix A: Work Distribution **48**

Appendix B: Time **49**

Appendix C: Technology Used **49**

Appendix D: References **50**

Chapter 1 : Introduction

1.1 Background

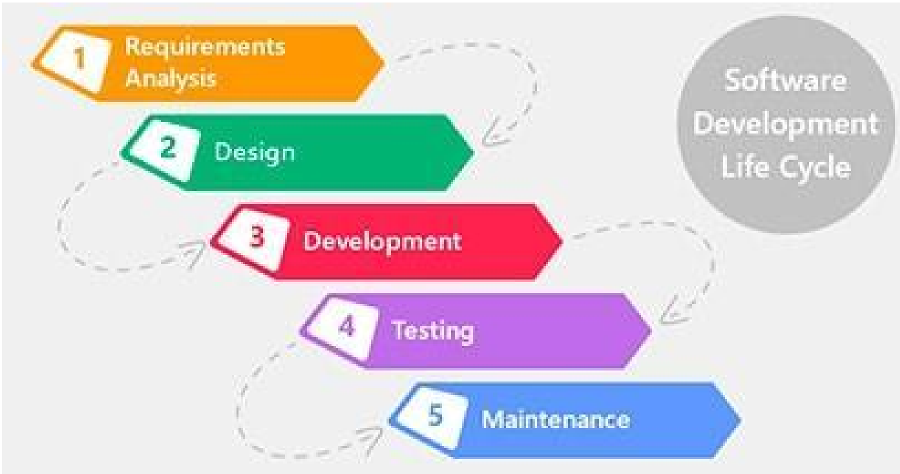
We all know that from time immemorial, earthquakes have had a great impact on human lives and resulted in many victims. In the past, scientists were optimistic that a practical method for predicting earthquakes would soon be found but over the years by continuing failure led many to question whether it was even possible. Predicting earthquakes has long been thought to be near impossible. Earthquake prediction is the science of seismology concerned with the specification of the time, location, and the magnitude of future earthquakes within stated limits and, particularly the determination of parameters for the next strong earthquake to occur in a region. Earthquake prediction is sometimes distinguished from earthquake forecasting which is concerned with probabilistic assessment of general earthquake seismic hazard, including the frequency and magnitude of damaging earthquakes in a given area over years or decades.

1.2 Scope

In every part of the world there are accidents that take place by what are called natural disasters like earthquakes ,volcanic eruptions ,tsunami floods, and many other natural disasters that happen and cause a lot of losses and damages whether it is economic losses or even human losses. So we want to intervene and help people understand those disasters and their results on our world. But we here are focusing on a specific natural disaster which is Earthquakes and we try here to use a huge dataset of the history of the earthquakes that happened in the world and analyze it to get useful information (statistics) and use it in a model to predict what will be resulting from these earthquakes and where there outcomes might take place all over the world. So we try to predict the earthquakes' shockwaves which means we try to predict the radius of occurrence of these resulting shockwaves.

1.3 Development Methodology

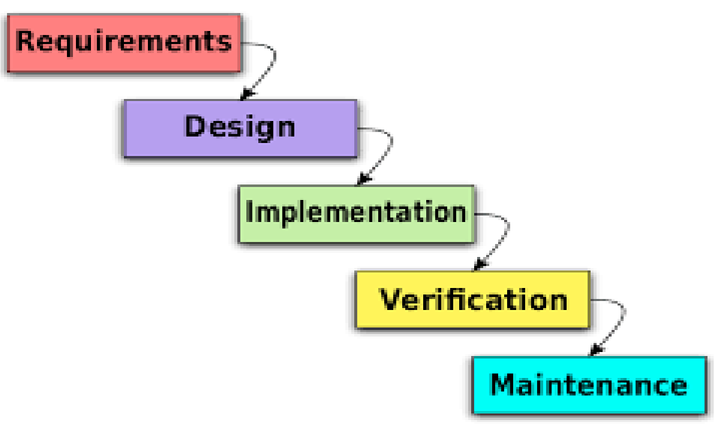
What is the Software Development Life Cycle (SDLC)?



Software Development Life Cycle consists of detailed steps and activities which describe how to design, develop, maintain, replace, alter, enhance, test or even launch a software. The activities can be broken down into a very detailed level but at the same time they can be grouped into five (5) core categories: Plan, Design, Develop, Test and Deploy. There are various software development life cycle models defined which are designed for different types of project. Each model follows a series of unique steps that best fit to its project type to ensure the success process of software development. Waterfall will be used in our case.

1.3.1 Waterfall

Waterfall is the earliest, best-known and most commonly used methodology. It is a sequential life cycle that is simple to understand and use. Each phase must be finished before another start which means no overlapping is allowed. The output of each phase serves as the input for the next stage.



1.3.2 Phases of Waterfall Model

1. Requirements

Requirement Phase mainly focuses on communicating with business users to gather and analyze requirements. Project managers try their best to understand and analyze the business, capture all the details of the user's needs, define the scope and arrange resources in the Business Case Documentation.

2. Design

With the Business Case Documentation in hand prepared in the Requirement Phase, Business Analysts evaluate and start on the logical design of the software by making use of the information and requirements that are collected by the Project Managers. Based on the high-level design which has fulfilled all the user requirements, System Analysts transform the high-level design to the physical design which puts hardware and software technology into consideration. System architecture is defined at the Design Phase as well.

3. Implementation

Implementation Phase is where the actual code is written. Programmers develop the software according to the instructions recorded on the documents prepared in Requirement and Design phases. Their output is the Functional Specification which files all the details of the functions that are implemented.

4. Testing

With the inputs from the Implementation Phase, testers in Testing Phase will draft the Test Plans based on the Functional Specification. Programmers prepare the Test Plan in a checklist to examine if every function is executable as expected. Business Analysts prepare the Test Plan for the users which focuses on meeting the user requirements. Finally, Quality Control (QC) experts gather all the documentations from all previous phases and do an overall test on every aspect on a deeper level that are documented including system architecture, technology used, etc.

5. Deployment

After receiving a “PASS” from the Testing Phase, the product is said to be ready to release. Software or Application will either be deployed to production servers or released for users to install on their own machine.

6. Maintenance

It is inevitable there are some defects or issues that will come up. In addition, the world keeps changing every day and as a result, enhancements are necessary from time to time. Maintenance Phase is for catering such situations and delivering changes to the users again. Within the Maintenance Phase, a subset of SDLC Waterfall Model is involved.

1.3.3 Advantages and Disadvantages

Advantages

- Easier to manage as there is a clear schedule for each stage that gives clear milestones.
- Easier to control with limited external factors as no overlapping development phase.
- Provide extensive documentation.

Disadvantages

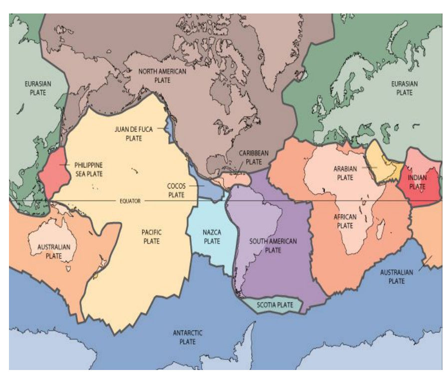
- Cannot have scope change or requirement change.
- Cannot preview the product until the deployment phase.
- Not flexible to handle unexpected risks.

1.4 Overall Description

1.4.1 Project Perspective

There will be two separate scripts, one for the data analysis stage and the other for the prediction model, those two scripts will be alongside a fully functioning web application, the application contain 2 levels:

The first level is about getting statistics after analyzing the dataset using a specific algorithm for the analytic process of data and then after getting the information from the analysis we get some important statistics and then we will show this information on a map like google map by label coloring to show magnitude and depth of each disaster.



The second level will be using a specific machine learning model to predict what will happen after an earthquake regarding the resulting shockwaves in the future, the user will choose an earthquake and the model will use its learning experience to predict it’s next shockwave magnitude.

1.4.2 Project Objectives

- Collect data about natural disasters for those who are interested in this field and analyze this data to answer some questions about disasters and find relationships between those disaster attributes.
- Predict the future whereabouts of some of those disasters (e.g., earthquakes) and its consequences.
- Create a research website that shows what we achieve in analyzing and predicting this data so it could be easy for anyone who is interested in natural disasters but cannot deal with databases.
- Provide an interactive map and graphs -to show the answers of our questions about disasters- in our website so the user can know what he wants very easily.

1.4.3 User Classes and Characteristics

Anyone who is interested in recent natural disasters or wants to gather statistical information about them. Also, anyone who is willing to use our prediction model for any type of usage.

1.4.4 Operating Environment

This project should operate on any working device which has internet access as all the components will be connected to the web application.

1.4.5 Design and Implementation Constraints

The data set with the most records need to be extracted manually, so having the latest data from those sets will require constant manual updating. On the other hand, we will be using an API for the latest earthquakes, it can provide us up to a month's worth of data, so we need to figure out a middle ground between those two data sources. The prediction of natural disasters is not that simple, specially the way we are trying to predict the happening of an earthquake or a volcanic eruption is not that common, so the number of papers published talking about this way is not that big. According to our research until now it is quite hard to predict disasters, especially earthquakes as we have to search for a pattern of events that happens every time before this disaster occurs.

Chapter 2: System Analysis

2.1 External Interface Requirements

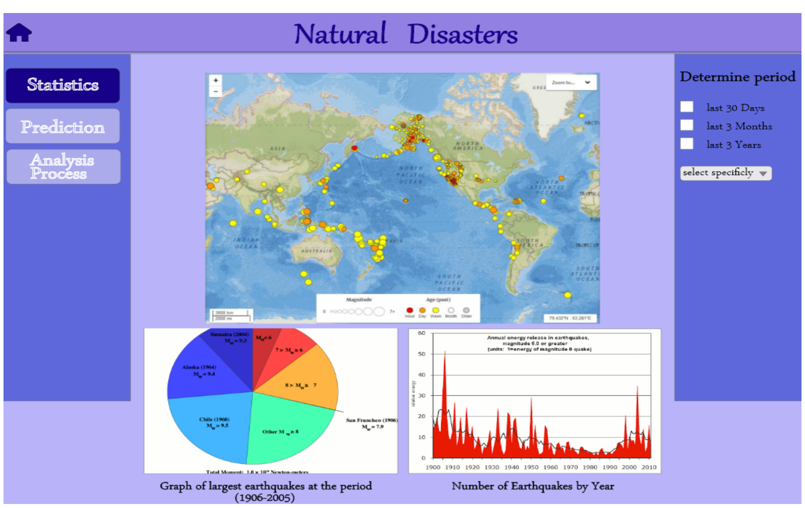
2.1.1 User Interfaces

2.1.1.1 Landing Page

This screen is just a welcome screen. User can choose one of the two choices and it will take him where he wants.

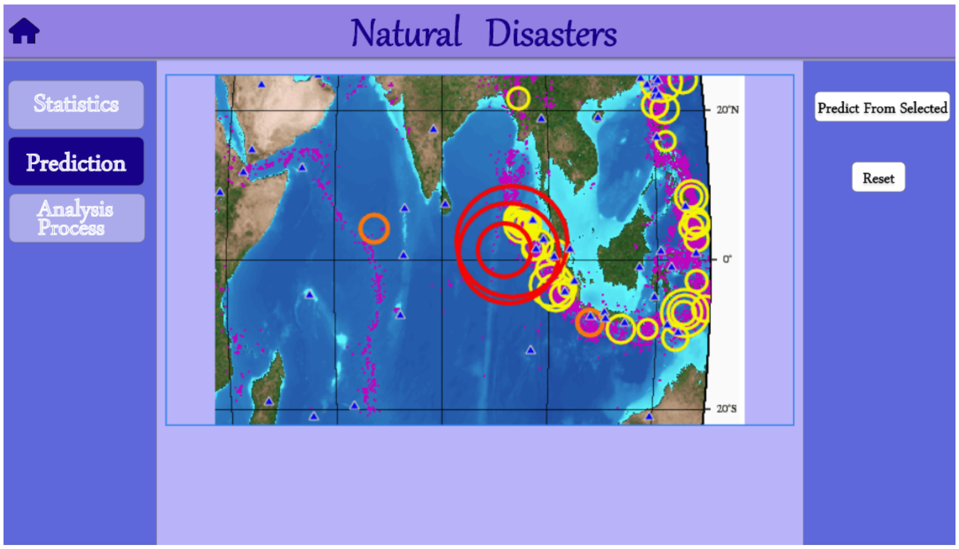
2.1.1.2 Interactive Map and Statistics

This screen is to show the results of our analysis for the data we have. It contains two things, the first one is an interactive map where the user can choose the disaster and a period of time and it will show him the place where every disaster takes place at the selected period. Also, he will find many graphs he can scroll down the page to see them all.



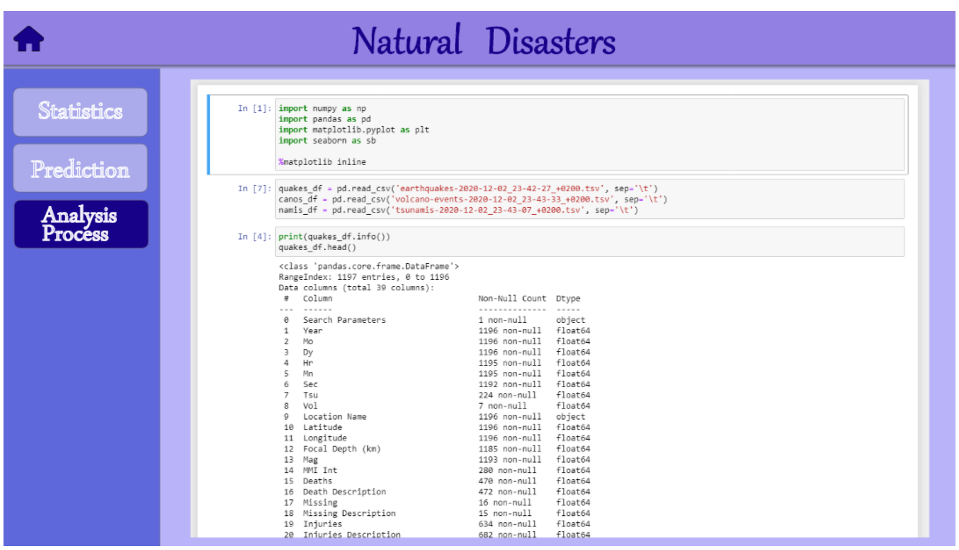
2.1.1.3 Prediction View

This screen where we will show the user our prediction results according to his choice. He can choose the earthquake then our model will predict the shock waves' magnitude.



2.1.1.4 Analysis Process View

This screen will show our user the steps of our analysis and how we reach the results he saw.



2.1.2 Software Interfaces

There will be a connection between the web application and a designated database with all the required information for basic functionalities.

2.1.3 Communications Interfaces

Any modern browser will prove effective in accessing the web application.

2.2 System Features

2.2.1 Interactive Natural Disasters Map

2.2.1.1 Description and Priority

- Allows users to view and interact with a map to view the various natural disasters.
- **Priority:** High.

2.2.1.2 Stimulus/Response Sequences

1 - User interacts with the map like any normal map.
2 - An element is selected from the map.
3 - A pop-up message is generated with the element's specific information.

2.2.1.3 Functional Requirements

REQ-1: View Map.

2.2.2 View Natural Disasters Statistical Information and Graphs

2.2.2.1 Description and Priority

- Allows users to view pre-generated graphs and statistics.
- **Priority:** High.

2.2.2.2 Stimulus/Response Sequences

1 - User chooses to view the map and statistical information.
2 - User views the provided graphs.

2.2.2.3 Functional Requirements

REQ-1: View Statistical Information.
More details in section 2.8.

2.2.3 Predict Earthquake Resulting Shockwaves' Aspects

2.2.3.1 Description and Priority

- Allows users to predict future shockwaves from a selected earthquake.
- **Priority:** High.

2.2.3.2 Stimulus/Response Sequences

1 - Users select from the recently occurred earthquakes.
2 - User inspects the resulting output.

2.2.3.3 Functional Requirements

REQ-1: Predict Shockwave.
More details in section 2.9.

2.2.4 View Analysis Process

2.2.4.1 Description and Priority

- Allows users to view the steps behind all the provided data and insights.
- **Priority:** Medium

2.2.4.2 Stimulus/Response Sequences

1 - User clicks the Analysis process button.
2 - Analysis view page is now the current page.

2.2.4.3 Functional Requirements

REQ-1: View Analysis Process Notebook.

2.3 Nonfunctional Requirements

2.3.1 Availability

- Hours of operation: any time the application is available.
- Locations of operation: any location to use the application.

2.3.2 Reliability

- Mean Time between Failures: failure should take only 1 second no more than it if there will be failure.
- Mean Time to Recovery: recovery should take 2 seconds maximum to solve the error.

2.3.3 Recoverability

- Recovery process: when an error happens we direct the user to the home page and then fix the error.

2.3.4 Scalability

- We have the ability to enhance the system by adding new functionality at minimal effort.

2.3.5 Maintainability

- Repair or replace faulty or worn-out components without having to replace the rest of the working parts.
- Prevent unexpected breakdowns.
- Maximize the product's life span.
- Meet new requirements.
- Make future maintenance easier.
- Cope with any changed environment.

2.3.6 Usability

- Ease of enrollment and use.
- Look and feel standards, interface is simple to use and has colorful design that can attract users to use the application.

2.3.7 Security

- The application is very secure that it’s difficult for anyone to edit or try to hack the system, many tests will be applied to make sure this point is achieved.
- There is maintenance every now and then to check security issues.
- We will have a security layer which will help us to achieve this requirement.

2.4 Other Requirements

2.4.1 Data management

- handle geo-referenced information such as maps, vector data

2.4.2 User Interface

The User Interface needs to provide a situation map for visualization of all geo-referenced data, preferably by using a 2D global metaphor, familiar to most users from Google Maps.

2.5 Design Strategy

2.5.1 Business Goals

- Provide pre-made plots and concluded statistical information about disasters for different users.
- Provide a prediction of the magnitude of an upcoming earthquake.

2.5.2 Users and Stakeholders

- Geology geeks

Anyone who is interested in enriching his knowledge about natural disasters.

- Researchers

Anyone who wants to research about disasters and wants some graphs, we will provide them graphs about what countries have the most disasters, relations between each disaster, etc.

- Governments

Decision maker is what we mean by the government as our prediction would be a second step for them because when scientists could predict earthquakes, they could use our model to know the magnitude of the upcoming quakes.

2.5.3 General Tasks

- Show graphs about disasters.
- Interact with the map to see the history of disasters.
- Show the result of our prediction.

2.5.4 Technological Constraints

- Should browse from PC
- Should have an internet connection

2.5.5 Branding Goals

- Easy to use.
- Prediction would be as accurate as possible.

2.6 User Analysis

2.6.1 User Groups

- Geology geeks.
- Researchers.

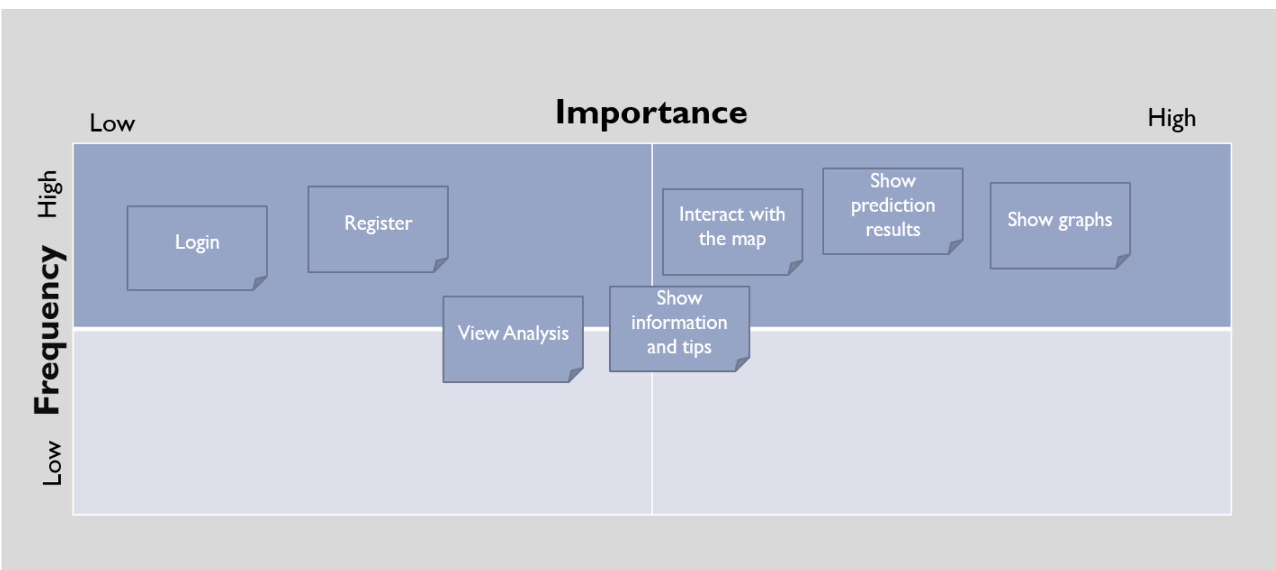
2.6.2 User Profile

Type:	**Geology Geeks**	**Researchers**	**Admin**
Age:	15 to 50 years old	20 to 50 years old	25 to 35 years old
Actor:	Male or Female	Male or Female	Male or Female
Gender	High School education	Bachelor degree	Technical education
Level of Education:	English (Not fluent)	English (Not fluent)	English
Language	Medium	Medium	High
Web Expertise:	15 to 50 years old	20 to 50 years old	25 to 35 years old
Expectations:	-Easy to use. -High speed in usage.	-Easy to use. -High speed in usage. -Many graphs.	-Comprehensive functionality.

2.6.3 Task Profile

Type:	Geology geeks	Researchers	Admin
Register:	✔	✔	
Login:	✔	✔	
view Map:			
Show Statistical Information:	✔	✔	
Predict Shockwaves:	✔	✔	
Mark Safe:	✔	✔	
Information and Tips:	✔	✔	
Keep data up to date:			✔
Maintenance & Support:			✔

2.6.4 Task Prioritization



2.6.5 Environmental Profile

Characteristic	Geology geeks	Researchers	Admin
Location	Indoor or Outdoor	Indoor or Outdoor	Indoor
Workspace	Home	Home	Office
Hardware	PC or Laptop	PC or Laptop	PC or Laptop
Software	Any browser	Any browser	Chrome

2.6.6 Personas

A)

Name: Ahmed.

Age: 17 years old.

Job: High school student.

Things he would like: A website that keeps track of every disaster so he could know what happens around the world.

B)

Name: Mohamed.

Age: 21 years old.

Job: College student.

Status: His doctor asked him to do research about natural disasters.

Things he would like: He didn’t want to check a lot of websites just to get updated graphs about natural disasters.

2.7 Data Gathering

2.7.1 National Oceanic and Atmospheric Administration (NOAA)

- NCEI (National Centers for Environmental Information) archives and assimilates tsunami, earthquake, and volcano data to support research, planning, response, and mitigation. Long-term data can be used to establish the history of natural hazard occurrences and help mitigate against future events.

- These datasets will be downloaded manually as the organization's API is on for authorized users.

- To be downloaded datasets shall contain data describing: Earthquakes, Tsunamis and Volcanic Eruptions.

2.7.2 United States Geological Survey (USGS) Earthquake Hazards

- Provides a useful API that allows us to view the latest occurring earthquakes in 3 different time ranges: Last day, Last 7 days or Last 30 days.

- All of these datasets will be accessed programmatically using their API.

2.8 Data Analysis

2.8.1 Process Overview

The datasets that we acquired need to be treated and filtered before even thinking about using it or getting correct insights from it so it will go through the data wrangling process. This process has specific steps to be followed to insure accurate insights, those steps are:

- Identification of variables and data types.
- Analyzing the basic metrics.
- Non-Graphical Univariate Analysis.
- Graphical Univariate Analysis.
- Bivariate Analysis.
- Variable transformations.
- Missing value treatment.
- Outlier treatment.
- Correlation Analysis.
- Dimensionality Reduction.

2.8.2 Data Snapshots

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 274 entries, 0 to 273
Data columns (total 46 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Search Parameters                1 non-null      object
 1   Year                             273 non-null    float64
 2   Mo                               272 non-null    float64
 3   Dy                               272 non-null    float64
 4   Hr                               240 non-null    float64
 5   Mn                               240 non-null    float64
 6   Sec                              226 non-null    float64
 7   Tsunami Event Validity           273 non-null    float64
 8   Tsunami Cause Code               273 non-null    float64
 9   Earthquake Magnitude             220 non-null    float64
 10  Vol                              11 non-null     float64
 11  More Info                        0 non-null      float64
 12  Deposits                         273 non-null    float64
 13  Country                          273 non-null    object
 14  Location Name                    273 non-null    object
 15  Latitude                         259 non-null    float64
 16  Longitude                        259 non-null    float64
 17  Maximum Water Height (m)         248 non-null    float64
 18  Number of Runups                 273 non-null    float64
 19  Tsunami Magnitude (Abe)          1 non-null      float64
 20  Tsunami Magnitude (Iida)         5 non-null      float64
 21  Tsunami Intensity                6 non-null      float64
 22  Deaths                           29 non-null     float64
 23  Death Description                29 non-null     float64
 24  Missing                          3 non-null      float64
 25  Missing Description              3 non-null      float64
 26  Injuries                         18 non-null     float64
 27  Injuries Description             21 non-null     float64
 28  Damage ($Mil)                    13 non-null     float64
 29  Damage Description               66 non-null     float64
 30  Houses Destroyed                 14 non-null     float64
 31  Houses Destroyed Description     22 non-null     float64
 32  Houses Damaged                   5 non-null      float64
 33  Houses Damaged Description       14 non-null     float64
 34  Total Deaths                     79 non-null     float64
 35  Total Death Description          79 non-null     float64
 36  Total Missing                    5 non-null      float64
 37  Total Missing Description        5 non-null      float64
 38  Total Injuries                   79 non-null     float64
 39  Total Injuries Description       89 non-null     float64
 40  Total Damage ($Mil)              32 non-null     float64
 41  Total Damage Description         135 non-null    float64
 42  Total Houses Destroyed           48 non-null     float64
 43  Total Houses Destroyed Description  64 non-null  float64
 44  Total Houses Damaged             42 non-null     float64
 45  Total Houses Damaged Description 65 non-null     float64
dtypes: float64(43), object(3)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 39 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Search Parameters                   1 non-null      object
 1   Year                                1196 non-null   float64
 2   Mo                                  1196 non-null   float64
 3   Dy                                  1196 non-null   float64
 4   Hr                                  1195 non-null   float64
 5   Mn                                  1195 non-null   float64
 6   Sec                                 1192 non-null   float64
 7   Tsu                                 224 non-null    float64
 8   Vol                                 7 non-null      float64
 9   Location Name                       1196 non-null   object
 10  Latitude                            1196 non-null   float64
 11  Longitude                           1196 non-null   float64
 12  Focal Depth (km)                    1185 non-null   float64
 13  Mag                                 1193 non-null   float64
 14  MMI Int                             280 non-null    float64
 15  Deaths                              470 non-null    float64
 16  Death Description                   472 non-null    float64
 17  Missing                             16 non-null     float64
 18  Missing Description                 15 non-null     float64
 19  Injuries                            634 non-null    float64
 20  Injuries Description                682 non-null    float64
 21  Damage ($Mil)                       139 non-null    float64
 22  Damage Description                  979 non-null    float64
 23  Houses Destroyed                    306 non-null    float64
 24  Houses Destroyed Description        448 non-null    float64
 25  Houses Damaged                      340 non-null    float64
 26  Houses Damaged Description          569 non-null    float64
 27  Total Deaths                        480 non-null    float64
 28  Total Death Description             478 non-null    float64
 29  Total Missing                       17 non-null     float64
 30  Total Missing Description           19 non-null     float64
 31  Total Injuries                      644 non-null    float64
 32  Total Injuries Description          692 non-null    float64
 33  Total Damage ($Mil)                 145 non-null    float64
 34  Total Damage Description            985 non-null    float64
 35  Total Houses Destroyed              309 non-null    float64
 36  Total Houses Destroyed Description  450 non-null    float64
 37  Total Houses Damaged                329 non-null    float64
 38  Total Houses Damaged Description    554 non-null    float64
dtypes: float64(37), object(2)
```

2.8.3 Data Preprocessing

The first step is to make sure that the datasets have no cleanliness problems, these problems are as follows: datasets seem to have datatype problems and null values that need either be removed or be calculated using average values. There arises another problem the datasets have, which is it has too many unneeded columns that are not even to be considered in our analysis.

Data tidiness is then to be considered, Every data set can be given a new attribute to set its type then all 3 data sets can be merged as one, that can be done only if they have all attributes common, merging date time attributes into on timestamp attribute would provide a much better representation of datetime instead of 6 separate attributes.

Now as data is cleaned and tidy, exploratory analysis can begin to find more accurate insights and conclusions.

2.8.4 Insights and Conclusions

The most import thing in this step is to ask appropriate questions that need answers and we chose a set of question that would help us further understand natural disasters:

- What is the year with the most disasters?
- Which place has the most natural disasters?
- What are the relationships, if any exists, between different types of natural disasters?
- Which type of natural disasters caused the most casualties?

Those questions might look simple yet they will require us to dig deep in the freshly cleaned dataset to find something of note.
Univariate visualisations will provide us with most of the basic statistical information required and give hints if there exists any outliers.
Bivariate one between each attribute will take place to find relationships between attributes, this will set us to further conclusions and show which way to go next to find better answers.
Multivariate exploration will verify our conclusion further by plotting more than 2 attributes at a time.
The conclusions produced in this process will form a separate dataset and plots to be displayed in our system, this system will be used as an explanatory analysis for users interested in natural disasters and their statistical information.

2.8.5 Graphs to be plotted and shown

After doing every analysis step that we mentioned we will answer questions from our cleaned dataset using graphs so we could help our users to find answers directly without much search. Those graphs would be:

- Number of disasters every year
- Countries with the most disasters
- Relation between a quake's magnitude and number of deaths
- Relation between a quake's magnitude and number of houses destroyed
- Relation between focal depth and magnitude
- What countries have the safest building structure which can face earthquakes

2.9 Machine Learning and Prediction Model

2.9.1 What do we want to predict?

- The area that the aftershocks will take place at.
- The number of aftershocks.
- Magnitude and Focal Depth of an earthquake in a specific longitude and latitude.

2.9.2 Which attributes should be used?

- Year
- Mo
- Dy
- Hr
- Mn
- Sec
- Latitude
- Longitude
- Focal Depth
- Mag

2.9.3 How will these attributes be used in the model?

First we are going to use the first 6 attributes to form a timestamp attribute. Then we will cluster every main earthquake with its aftershocks using the timestamp and distance between every earthquake in the same area and that happened in the same timestamp range. Then we use latitude, longitude, magnitude and focal depth in our model to predict the number of aftershocks that will happen due to a major earthquake. For our last prediction we will use the latitude, longitude and timestamp to predict the upcoming quake's magnitude and depth.

2.9.4 Which machine learning models can be used?

We are going to use more than one model for prediction and compare between their results to find the best accuracy level between them.

- Linear Regression
- Random Forest

There are many more models to be tested, these are only the first to be tried.

2.9.5 Why do we choose regression models?

As known, regression is supervised learning where we have input variables and output variables and we use an algorithm to map the input into output based on the examples in the training set. After studying our dataset we found that supervised learning would be helpful for our prediction than unsupervised learning as we have inputs and will calculate outputs to form a training set where the model can map its input to output and provide us the result of our prediction. Also, as we are going to predict numerical value we thought that regression techniques would be the best option.

2.9.6 How will we apply the regression model?

In linear regression, we have dependent variable y and independent variables or predictors x = $(x_1, \ldots, x_r)$ where r is the number of variables. The relationship between x and y can be represented in this equation y = $\beta_0 + \beta_1 x_1 + \cdots + \beta_r x_r + \varepsilon$. β_0 ,β_1, …., β_r are the regression coefficients and ε is the random error.

In our first two predictions, the dependent variable would be the number of aftershocks and the area that those aftershocks will take place. While the independent variables will be timestamp, latitude, longitude, magnitude and focal depth.

In the third prediction, the dependent variable would be the magnitude of an upcoming earthquake. While the independent variables will be the timestamp, latitude and longitude.

Chapter 3: System Design

3.1 Use-Cases

3.1.1 Register

Use-case:	Register	Use-case ID:	UC-01	Priority:	Medium
Description:	Allows users to Register into the system.				
Actor:	User				
Trigger:	Users want to use the system.				
Preconditions:	None.				
Postcondition:	1- User gains access to the system. 2- Show Login Page.				
Normal Course:	1- User fills the represented information fields. 2- Fields get verified. 3- Successfully register user in the system.				
Alternative Course:	None.				
Assumptions:	None.				

3.1.2 Login

Use-case:	Login	Use-case ID:	UC-02	Priority:	Medium
Description:	Allows users to log in to the system.				
Actor:	User				
Trigger:	Users want to use the system.				
Preconditions:	Users already registered in the system and have UserID and Password.				
Postcondition:	1- User login successfully to the system. 2- Show Welcome Page.				
Normal Course:	1- User enters the ID and Password. 2- ID and Password getting verified from the Users Table. 3- Successfully login into the system.				
Alternative Course:	None.				
Assumptions:	Users already registered in the system and have the correct ID and Password.				

3.1.3 View Map

Use-case:	View Map	Use-case ID:	UC-03	Priority:	High
Description:	Allows users to view and interact with a map to view the various natural disasters				
Actor:	User				
Trigger:	User wants to view specific natural disaster data.				
Preconditions:	User signed in.				
Postcondition:	1- View specific element information.				
Normal Course:	1- User interacts with the map like any normal map. 2- An element is selected from the map. 3- A pop-up message is generated with the element's specific information.				
Alternative Course:	None.				
Assumptions:	None.				

3.1.4 View Statistical Information

Use-case:	View Statistical Information	Use-case ID:	UC-04	Priority:	High
Description:	Allows users to view pre-generated graphs and statistics.				
Actor:	User				
Trigger:	User wants to view natural disaster statistics				
Preconditions:	User signed in.				
Postcondition:	1- View specific element information.				
Normal Course:	1- Display pre-made plots and concluded statistical information regarding the past natural disasters.				
Alternative Course:	None.				
Assumptions:	None.				

3.1.5 Predict Shockwave

Use-case:	Predict Shockwave	Use-case ID:	UC-05	Priority:	High
Description:	Allows users to predict future shockwaves from a select earthquake.				
Actor:	User				
Trigger:	User wants to know the magnitude of future shockwaves.				
Preconditions:	User signed in.				
Postcondition:	1- Show magnitude of potential shockwaves of the selected earthquake on map				
Normal Course:	1- User selects from the recently occurred earthquakes. 2- User inspects the resulting output.				
Alternative Course:	None.				
Assumptions:	None.				

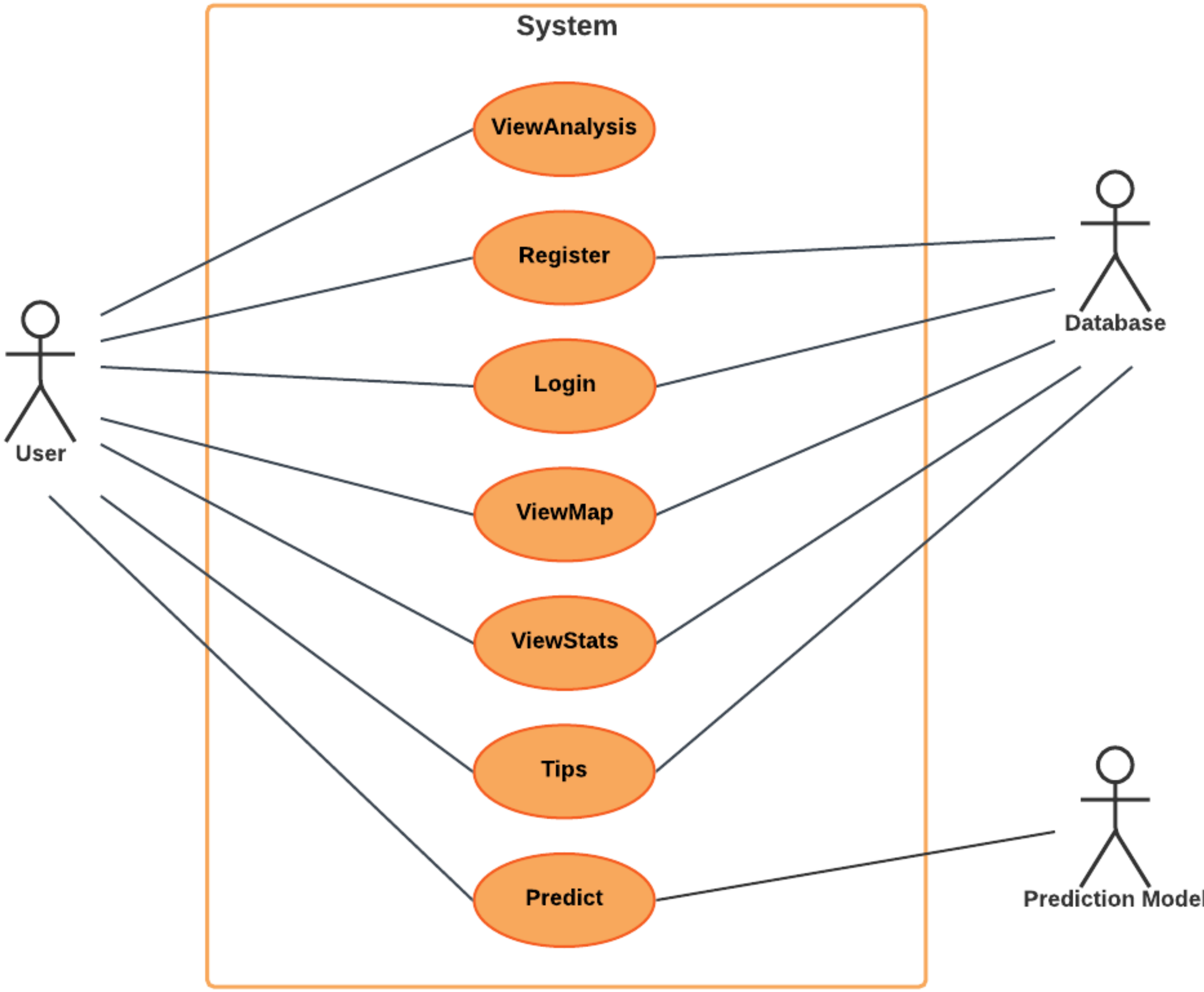
3.1.6 View Analysis Process

Use-case:	View Analysis Process	**Use-case ID:**	UC-06	**Priority:**	Medium
Description:	Allows users to view the steps behind all the provided data and insights.				
Actor:	User				
Trigger:	None.				
Preconditions:	User signed in.				
Postcondition:	None.				
Normal Course:	1 - User clicks the Analysis process button. 2 - Analysis view page is now the current page.				
Alternative Course:	None.				
Assumptions:	None.				

3.1.7 Information and Tips

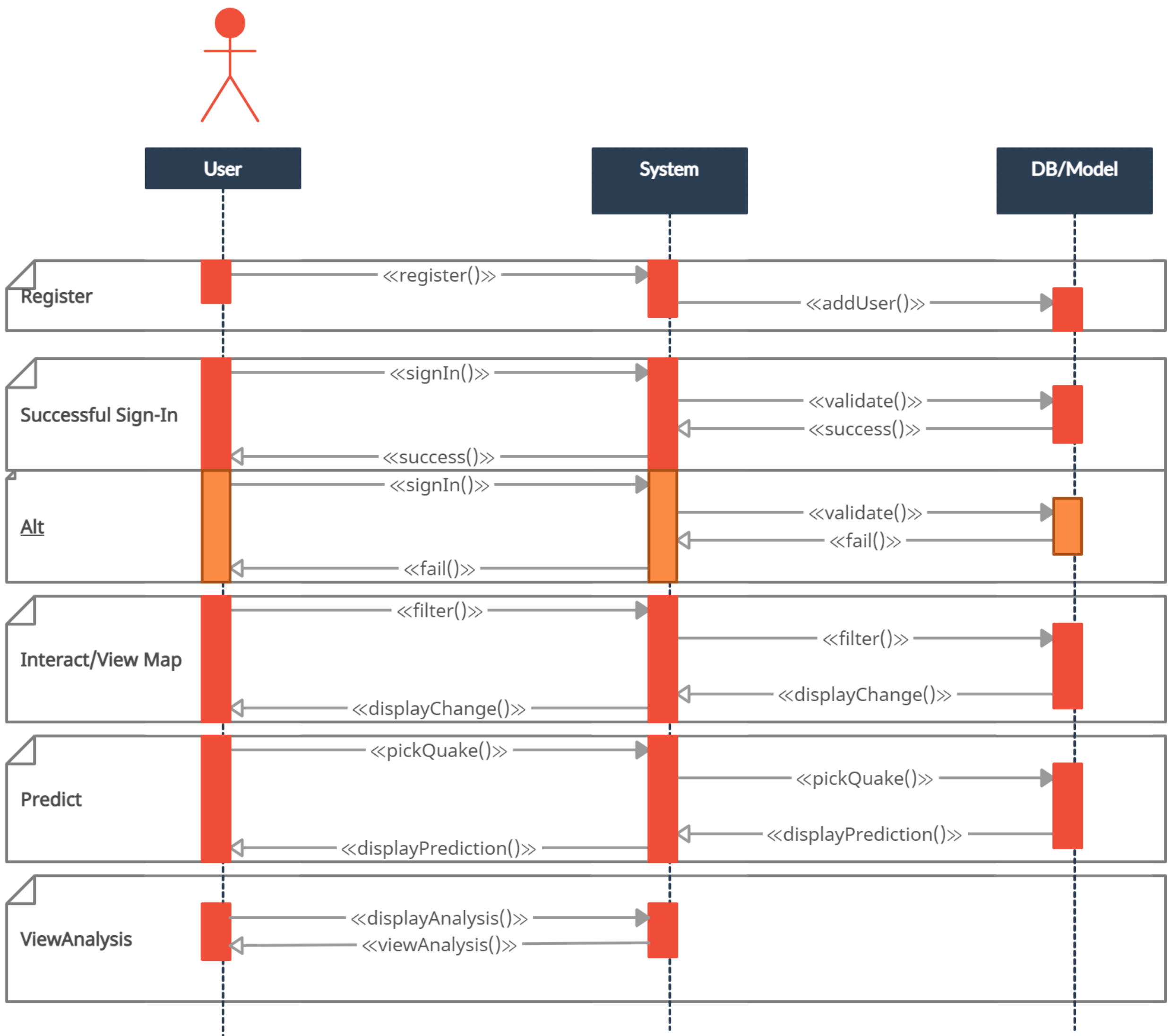
Use-case:	Tips	**Use-case ID:**	UC-07	**Priority:**	Low
Description:	Display to users general natural disasters information and tips to follow in case of any emergency related to those disasters.				
Actor:	User				
Trigger:	User want to access general information and emergency tips for natural disasters.				
Preconditions:	User signed in.				
Postcondition:	None.				
Normal Course:	1- User chooses general information and emergency tips for natural disasters.				
Alternative Course:	None.				
Assumptions:	None.				

3.2 Use-Case Diagram



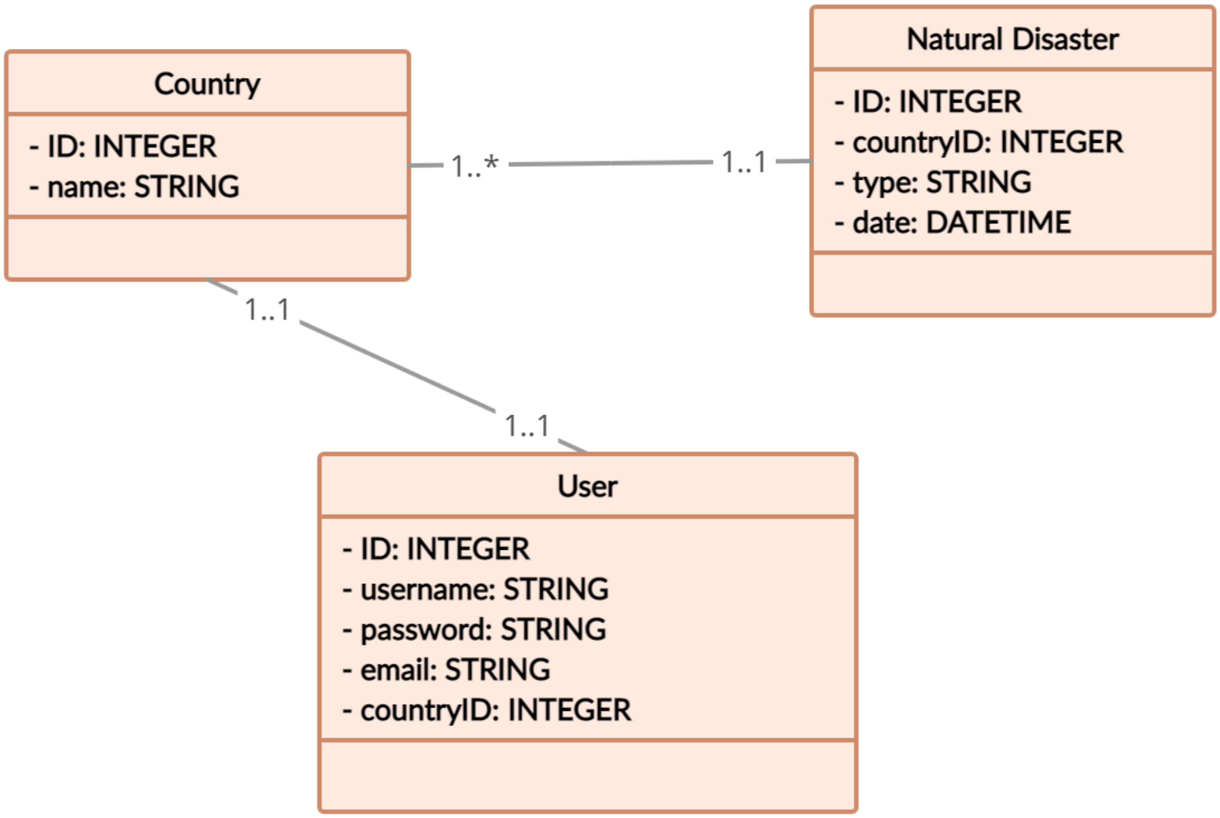
A user can access all the functions of out system as they are designed to be so, ViewAnalysis function is just a saved view of our analysis so there is no need to access the database for it, the rest of the functions except for Predict need access to our database to save, retrieve and modify data according to each requirement, Predict function will access the prediction model script as its backend part.

3.3 Sequence Diagram



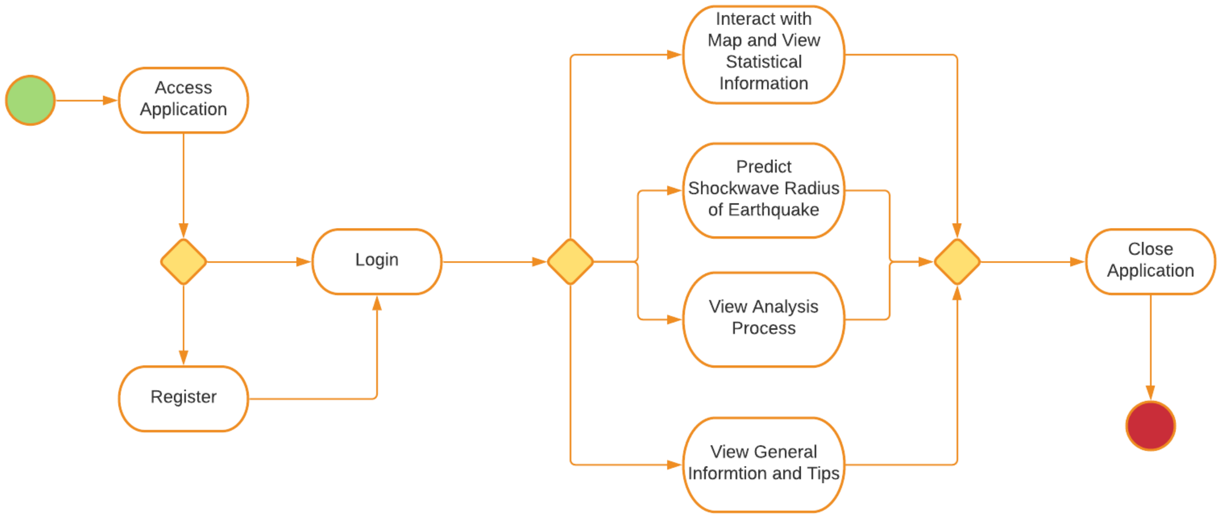
Each of the functions has scenario sequences, those are presented in this diagram.

3.4 Class Diagram



User data shall be stored and a relationship between each user and natural disaster will be defined by their mutual country.

3.5 Business Process Model and Notation (BPMN)



User goes through our system according to the flow of this diagram every condition or decision made is considered in its making.

Chapter 4: Implementation

4.1. Application Backend

4.1.1. Backend Server

At first we were going to use one of python's frameworks to build the backend server but Node.js was just too good to be missed out in this application. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js comes with its own package manager to manage a whole lot of useful packets that help in building a proper server. npm or Node package manager, is a package manager for the JavaScript programming language maintained by npm, Inc. npm is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

4.1.2. Database

Then comes the data and where to store it, we chose a document based no-sql database called MongoDB. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License. Being able to deal with a no-sql database like that was something new and somewhat challenging as it was the first time handling such technology. To handle MongoDB within Node.js, the Mongoose package was used to make use of the already known structure of schemas in such a no-sql database.

4.1.3. RESTful API

Combining both of these powerful technologies gave us an opportunity to create a RESTful API. To manage the API's routes we used Express.js which made it somewhat easier to handle middlewares and database connections.
There are 2 route types in the server, one works for user authentication and the other is CRUD in nature. The final route URLs are to be used in the frontend-side server to perform the needed actions on our data.

4.2. Application Frontend

4.2.1. Static HTML5, CSS3 and Basic Javascript

Designing the structure of our web application needed to be handled first with basic HTML tags to make a simple skeleton for the application, then features started snowballing from that point, more forms were added, additional buttons and even sectioning tags. Afterwards, basic CSS came in to show-off the full potential of our built structure, editing each tag's properties according to what we saw fit to our design and vision for our application. Then, Javascript was used to handle functionality issues such as creating a collapsible navigation bar and managing the map library to be used in displaying our database. Leaflet.js is an open source JavaScript library used to build web mapping applications. In Leaflet.js we used MapBox which is an American provider of custom online maps for websites and applications.

4.2.2. React.js

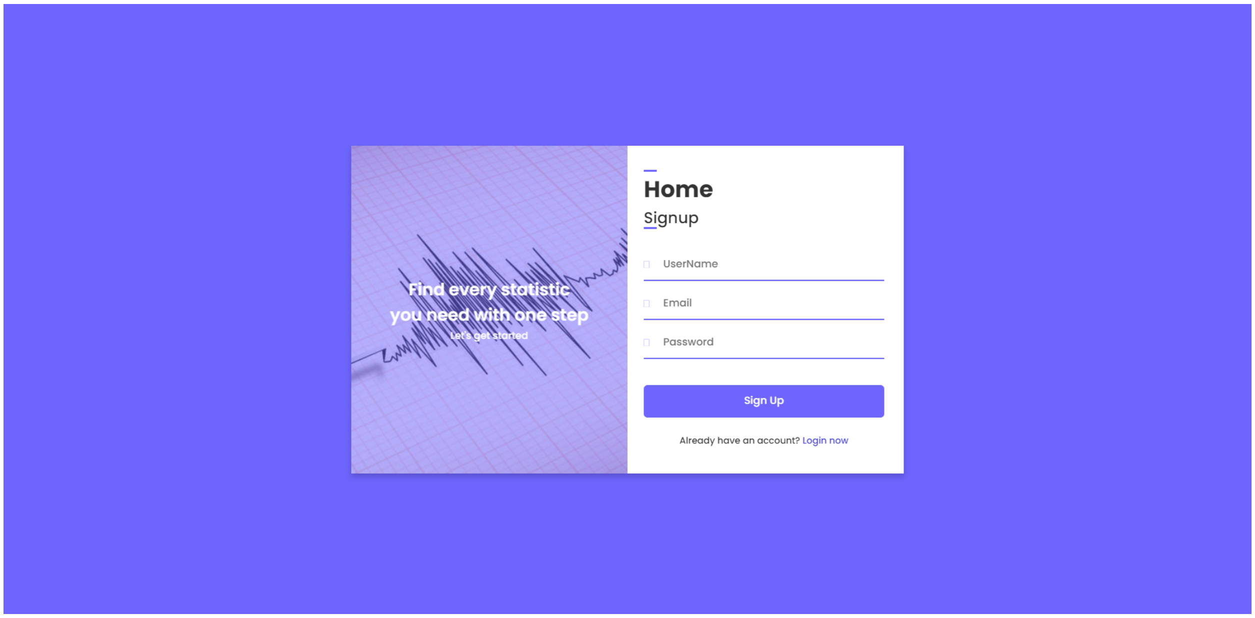
At first, only static and basic technologies were going to be used in our front-end side of our application, then, we decided to take it to the next level and implement React.js as our main interface builder. React.js is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React.js can be used as a base in the development of single-page or mobile applications. With React.js, our application can now be a single page application with faster response time, the old basic HTML pages were converted to Views and each reoccurring section of those pages was decomposed into a Component to be dynamically used. Then comes the React DOM Router, which is used to handle changes in the URL as changes in which Components and Views are displayed on the Page. In fact, this was somewhat challenging as it felt like re-building the whole frontend-side from scratches all over again.

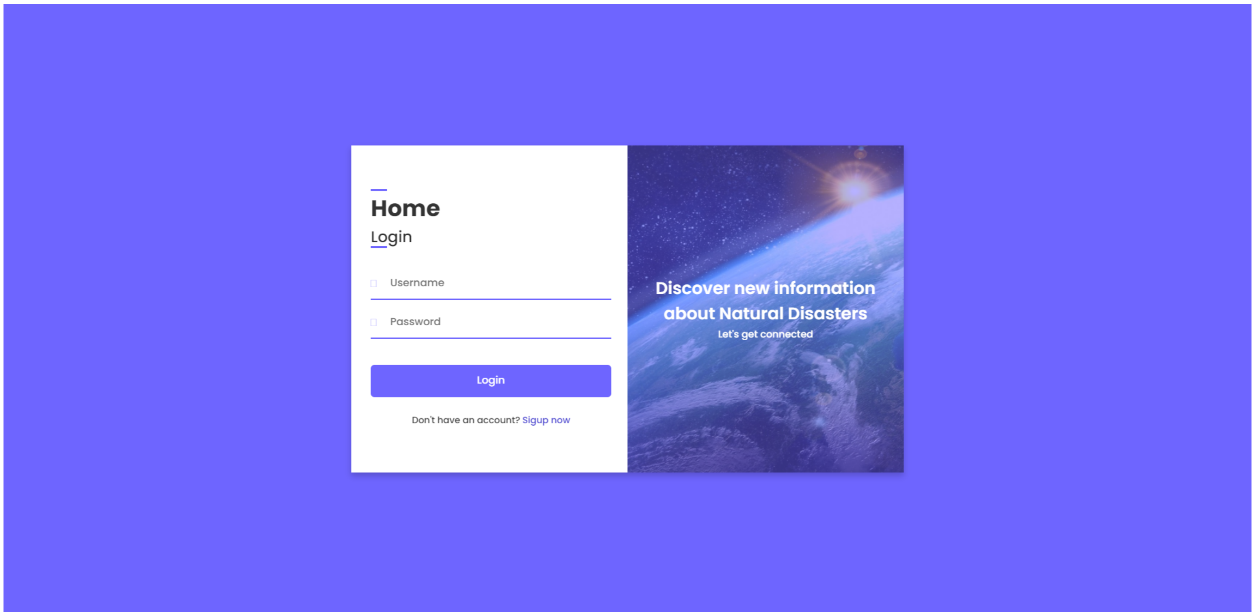
4.3. Authentication

To keep users data more secure, authentication techniques were applied on the connection between backend and frontend sides. Password hashing before being added to the database and Cookies are used to pass the required general data alongside an authentication token, specifically a JsonWebToken, the token is generated using a secret string and user's ID. This token is then kept in the browser for a period of time imitating a session. Tokens are used again to make sure whether a user has access to our application or not via a custom middleware in the backend side.

4.4. Application Snapshots

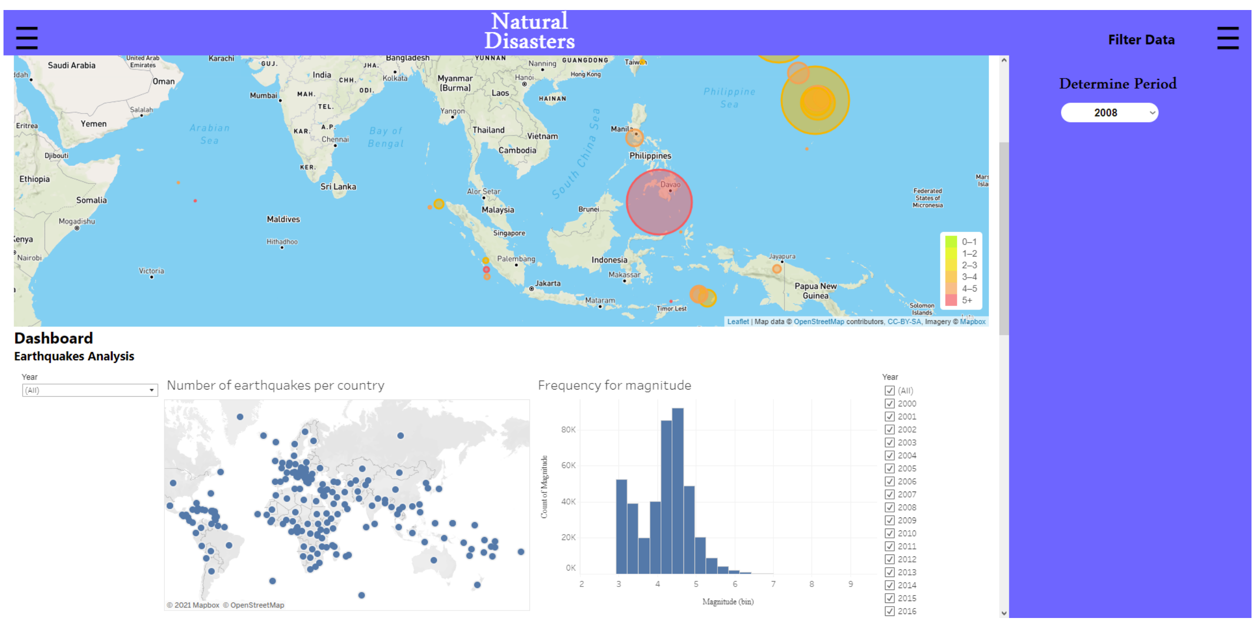
4.4.1. User Signup/Login



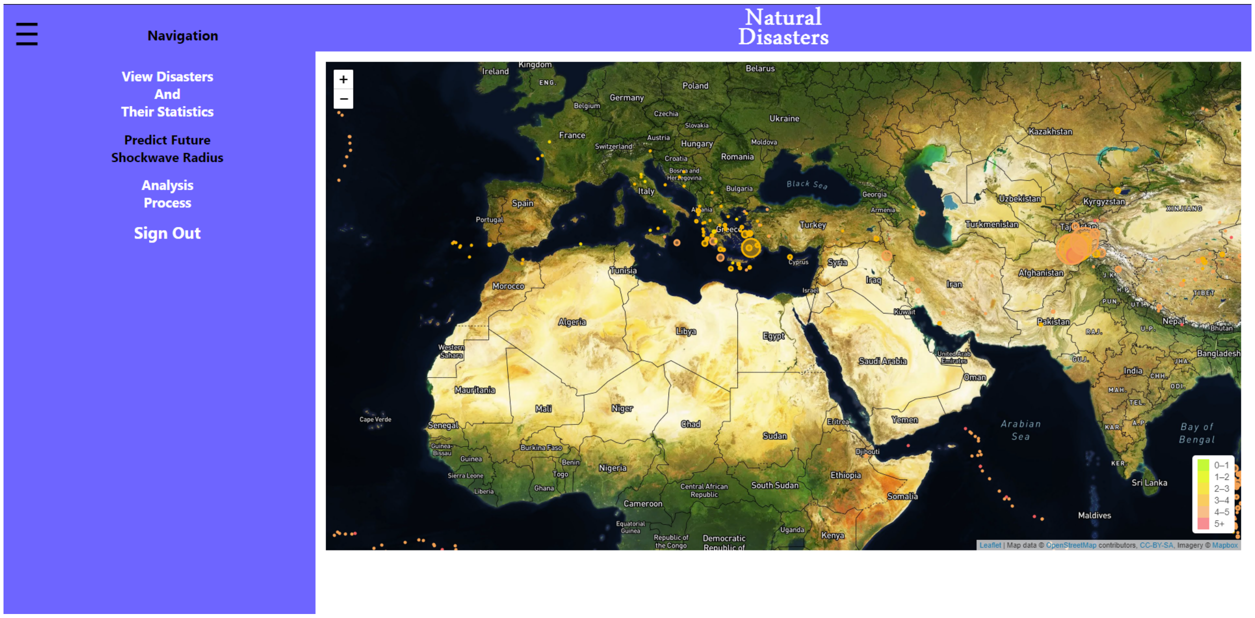


4.4.2. Home Page

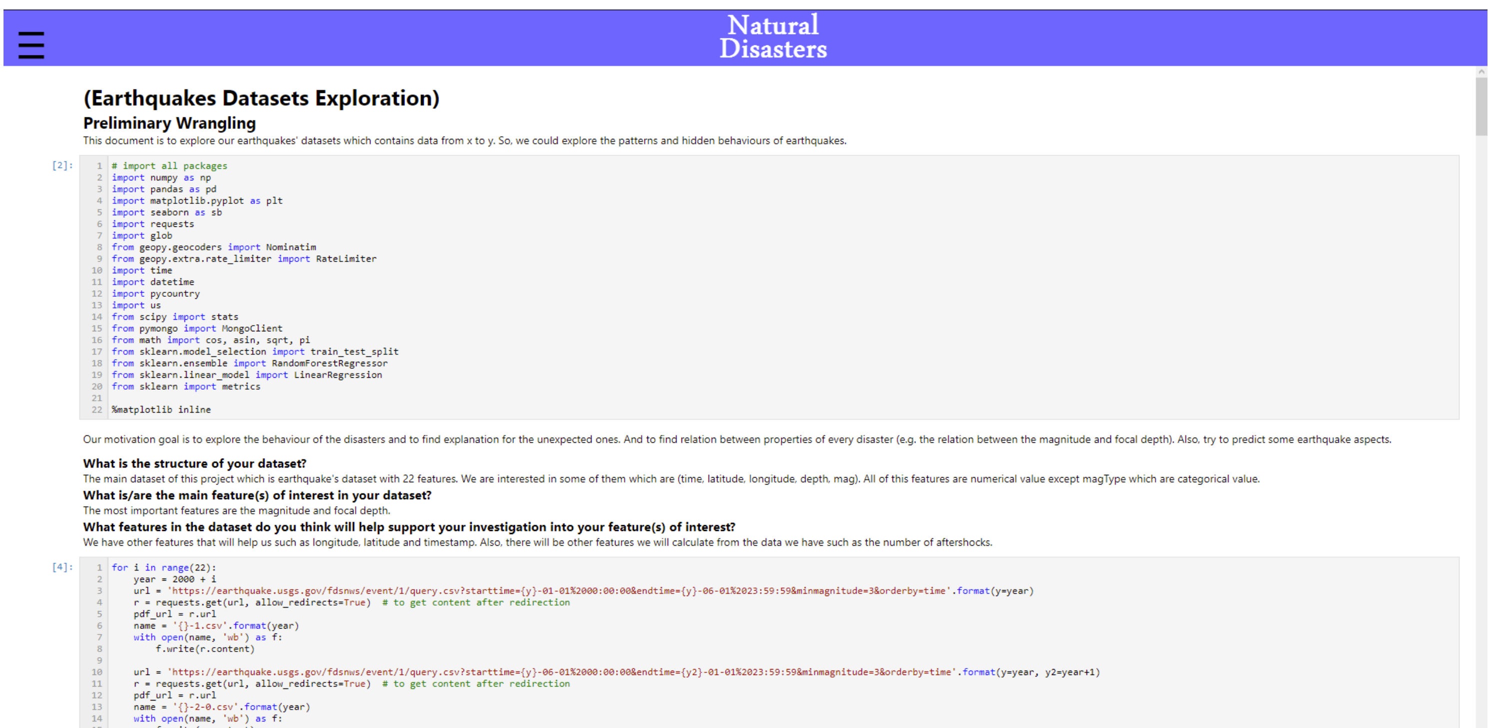
4.4.3. Statistics Page



4.4.4. Prediction Page



4.4.5. Analysis Page



Natural Disasters

(Earthquakes Datasets Exploration)

Preliminary Wrangling

This document is to explore our earthquakes' datasets which contains data from x to y. So, we could explore the patterns and hidden behaviours of earthquakes.

[2]:
```python
# import all packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
import requests
import glob
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiter
import time
import datetime
import pycountry
import us
from scipy import stats
from pymongo import MongoClient
from math import cos, asin, sqrt, pi
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn import metrics

%matplotlib inline
```

Our motivation goal is to explore the behaviour of the disasters and to find explanation for the unexpected ones. And to find relation between properties of every disaster (e.g. the relation between the magnitude and focal depth). Also, try to predict some earthquake aspects.

What is the structure of your dataset?

The main dataset of this project which is earthquake's dataset with 22 features. We are interested in some of them which are (time, latitude, longitude, depth, mag). All of this features are numerical value except magType which are categorical value.

What is/are the main feature(s) of interest in your dataset?

The most important features are the magnitude and focal depth.

What features in the dataset do you think will help support your investigation into your feature(s) of interest?

We have other features that will help us such as longitude, latitude and timestamp. Also, there will be other features we will calculate from the data we have such as the number of aftershocks.

[4]:
```python
for i in range(22):
    year = 2000 + i
    url = 'https://earthquake.usgs.gov/fdsnws/event/1/query.csv?starttime={y}-01-01%2000:00:00&endtime={y}-06-01%2023:59:59&minmagnitude=3&orderby=time'.format(y=year)
    r = requests.get(url, allow_redirects=True)  # to get content after redirection
    pdf_url = r.url
    name = '{}-1.csv'.format(year)
    with open(name, 'wb') as f:
        f.write(r.content)

    url = 'https://earthquake.usgs.gov/fdsnws/event/1/query.csv?starttime={y}-06-01%2000:00:00&endtime={y2}-01-01%2023:59:59&minmagnitude=3&orderby=time'.format(y=year, y2=year+1)
    r = requests.get(url, allow_redirects=True)  # to get content after redirection
    pdf_url = r.url
    name = '{}-2-0.csv'.format(year)
    with open(name, 'wb') as f:
        f.write(r.content)
```

4.5. Testing

4.5.1. Unit Testing

This phase of testing is taking place during the coding phase as we focus on individual units and components to make sure that every component is fully functional.

4.5.2. System Testing

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result
T1	Sign up	Enter information	A user is added to the database	As expected
T2	Login	Enter credentials	A user is allowed to use our features	As expected
T3	Show map	Go to maps and stats page	A map is shown with flags for every earthquake	As expected
T4	Show dashboard	Go to maps and stats page	A dashboard with many graphs is shown	The dashboard appeared for a while then gives an error
T5	Show dashboard	Go to maps and stats	A dashboard with many graphs is shown	As expected
T6	Filter map	Choose the year you want	The map refreshes with new flags that matches the filter	As expected

Chapter 5: Results

5.1. Evaluation of Data Analysis Process

There are some steps every data analysis project should pass through them. Define questions, Collect Data, Data Cleaning, Analyze Data and Interpret results to answer the questions we defined earlier.

In this chapter we will discuss what we did in detail for both USGS dataset and NOAA dataset.

5.1.1. USGS Dataset

In this dataset, we define some questions that we want to find answer for (e.g. The most 5 countries that suffer from earthquakes, The year with the most earthquakes in the 21st century). Also, We want to know if there is any usual behaviour for earthquakes or is it just one of the natural hazards that still has no explanation till now?

We started collecting data through the API that USGS has provided users with. We faced our first limitation here as the API can’t let you download more than 20,000 records at a time. We solved this issue and started to download data about every earthquake that has magnitudes larger than 3. The reason behind considering this filter is that any earthquake with a smaller magnitude is not felt by most people.

After collecting the data we take a look in it to know what features we have, what features are we interested in etc. We started cleaning our data by dropping the features that we thought wouldn't be necessary. Then we started to change wrong data types and this was exclusive for the time feature only which we change its data type and then split it into day, month and year. After that we started looking for null values and we found that there are more than 2,000 records that have no country data so we started to solve this issue by using geopy library to find the country using latitude and longitude. Successfully we could reduce the number of these null values by almost 700 records and we removed the ones we couldn't get its value.Also, some countries were holding the US state name not the country name so we changed it one by one to USA. Then we removed duplicates we found. At this point we thought that our data was ready for analysis.

We started our analysis with univariate analysis to investigate every attribute’s distribution to see if there are any outliers. So, we started to investigate magnitude and depth features and luckily we didn’t find any outliers. Then we plot a countplot for country, month and year features to see if there is anything unusual.

After univariate analysis we jumped into bivariate analysis to see the relation between each two features. So, we looked to see if there are any correlations between the variables. We found that there is a very weak correlation between depth and magnitude. We didn't think that this relation is worth more investigation. We were surprised with these results as we expect that there will be no usual behaviour for earthquakes but we thought that there will be a strong relation between depth and magnitude.

At this point, we stopped with our analysis and started to get some answers for our questions. -Will be shown as graphs in next chapter-

5.1.2. NOAA Dataset

This dataset was a little bit different than the previous one. As this one didn't record every single earthquake, it records the ones which did damage in its area. So, here our questions were a little bit different (e.g. Which country suffers from the most damages) and is there by any chance could it be the same as the country with the most earthquakes or not?

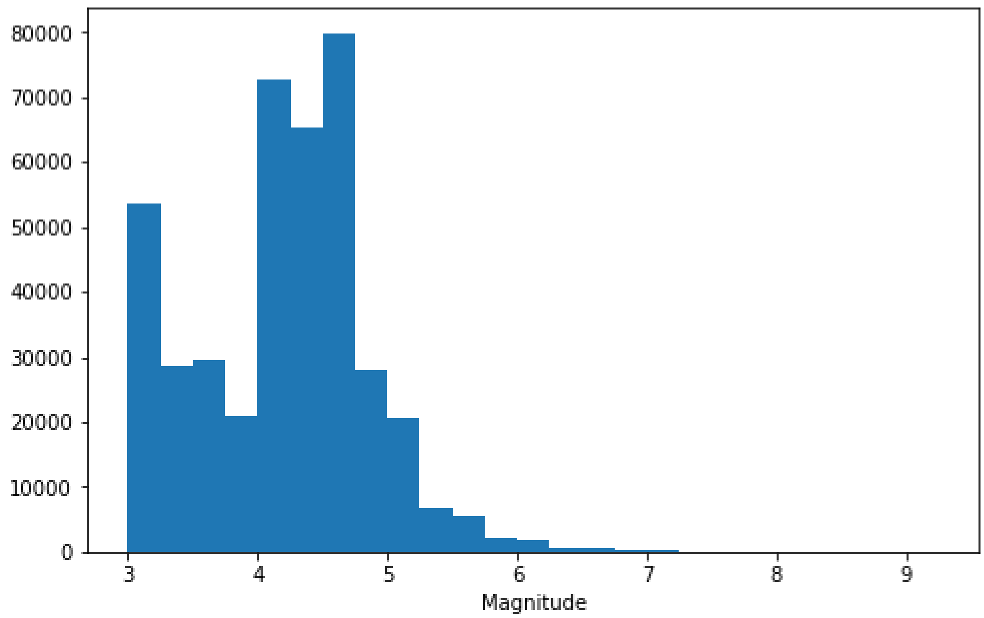
We started with the same start as we did in the first dataset by dropping the features that we aren't interested in. Then we changed some features' names and dropped some missing values that we can't calculate.

This data didn't have so much to do with cleaning. So, we jumped straight to univariate analysis. Where we checked some count plots to see which values are repeated the most in every attribute.

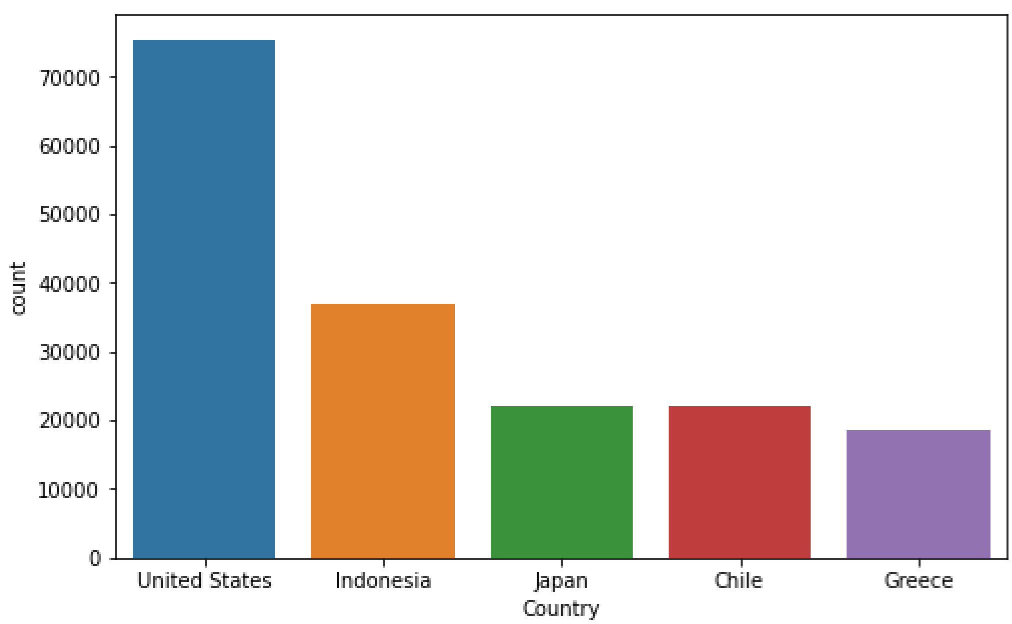
After that we did some correlation analysis to the data and saw that there is a relation between the houses damaged and the damage measured by millions of dollars which totally makes sense. Also, we found that there is a slightly weak correlation between magnitude and total number of deaths which also makes sense. At this point we thought that our analysis had ended for this dataset and we started to get some answers and look for some insights from the data we have now.

5.2. Visualizations

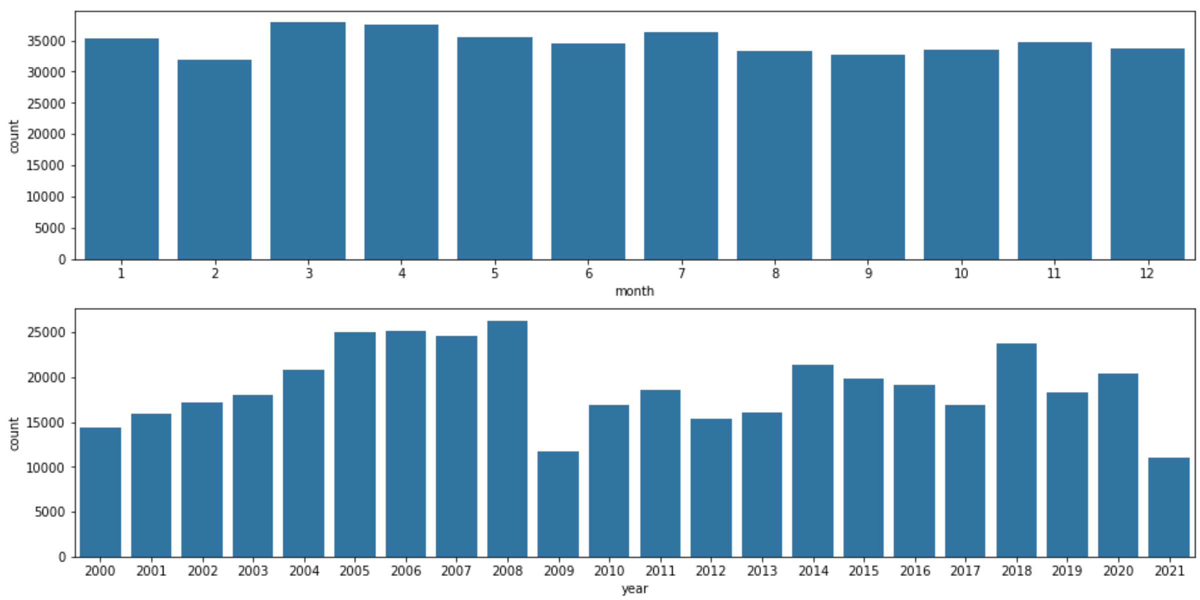
In this chapter we will be presenting visualizations of the output which answer all of our questions.



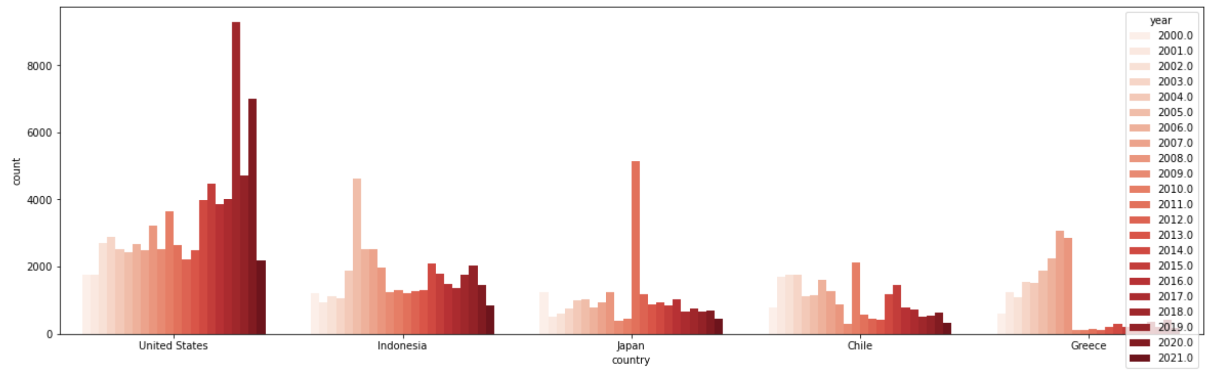
In this histogram we could see that earthquakes with magnitude between 4 and 5 are the most common earthquake and it's very rare that we could witness an earthquake with magnitude larger than 7.



Here we could see the top 5 countries that suffered from earthquakes the most. The USA is the most with a huge gap between it and the next one which is Indonesia, then comes Japan and Chile with almost the same number of earthquakes and in the fifth place comes Greece. What is interesting in this histogram is that the 5 countries are from 4 different continents which may mean that there is no specific area on the earth that earthquakes are most likely to happen at. But, we didn't see any African country so we took a look for the top 10 and still we didn't see any African country and when we checked the least 10 countries we found Nigeria and Cameron with just 1 earthquake in the last 21 years which is surprising.

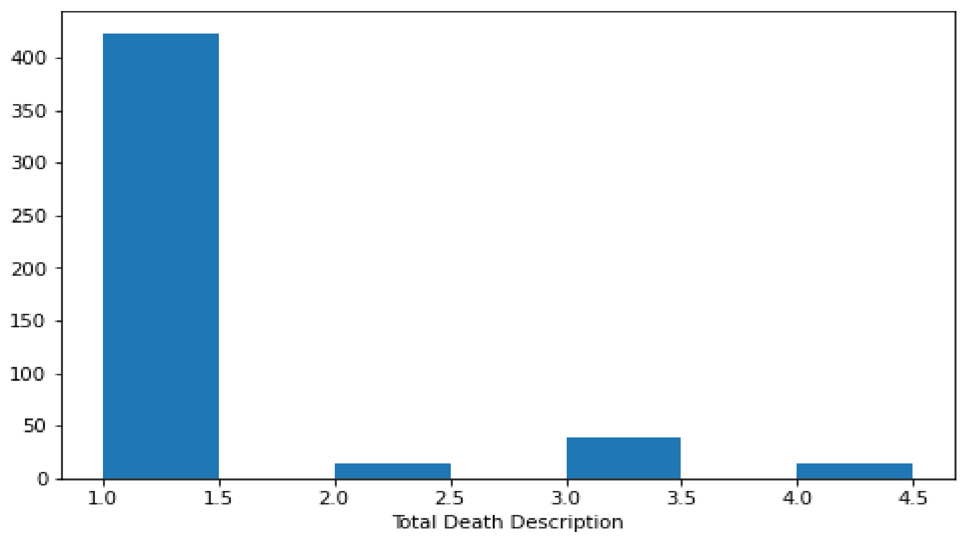


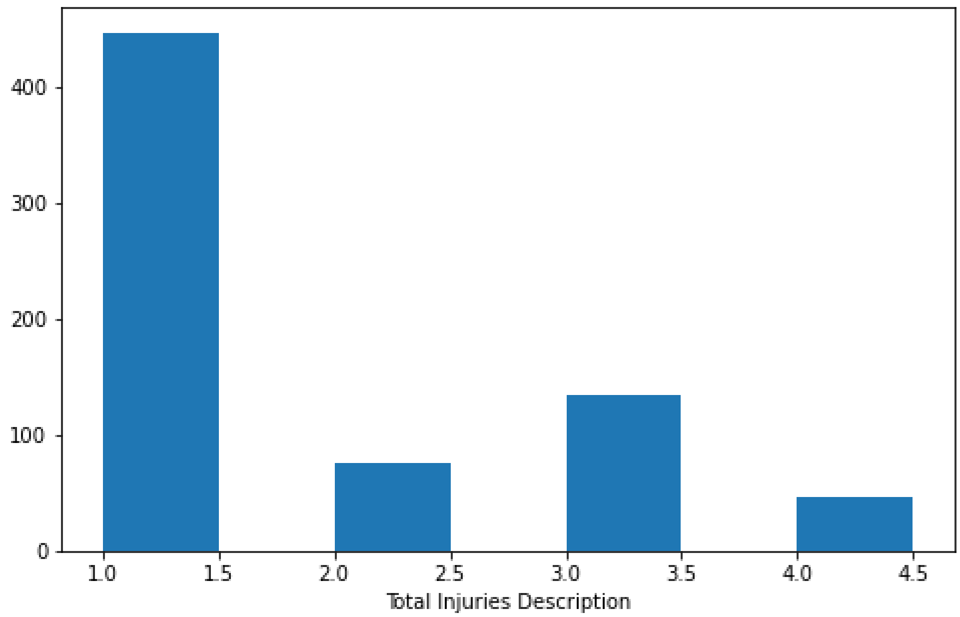
Then we started to see if there is a season where the weather affects in any way the happening of an earthquake. But, we didn't find anything that could prove this point. Also, as shown in the year histogram is that the most year with earthquakes is 2008 and the least one is 2009 which could be a little bit surprising.

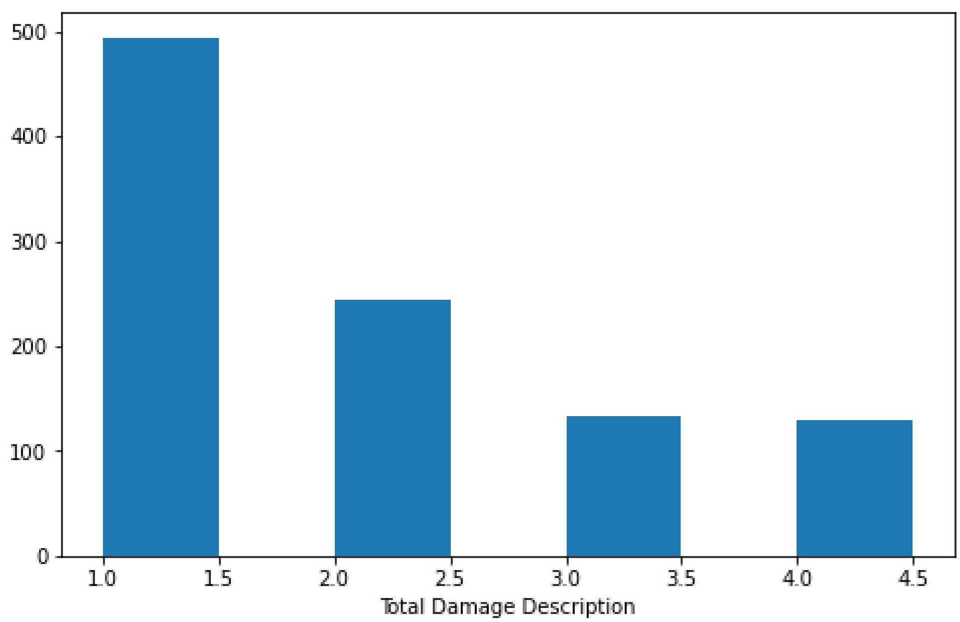


Here we thought that we could investigate every year's count plot but for every country alone. We can see here that since 2008 Greece hasn't suffered from earthquakes as it was before this year but still it is in the fifth place among all countries. We could also see the change that happened in the USA as after 2017 the number of earthquakes has increased a lot and they recorded the most two years that witnessed the causings of an earthquake.

Now after we know the frequency of earthquake events throughout the years and which countries suffer the most from this natural hazard lets see what damage could this quakes cause.

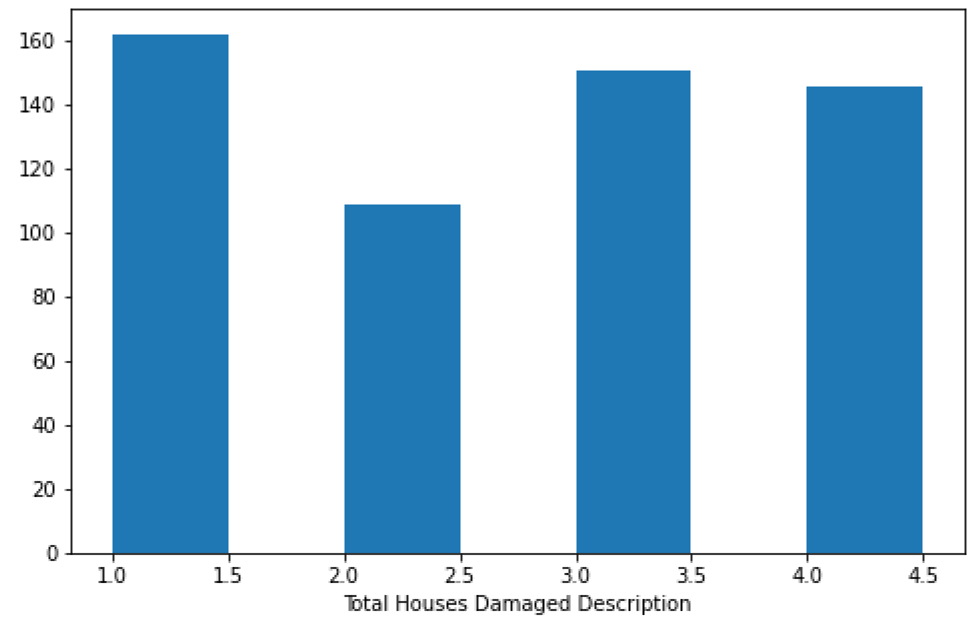




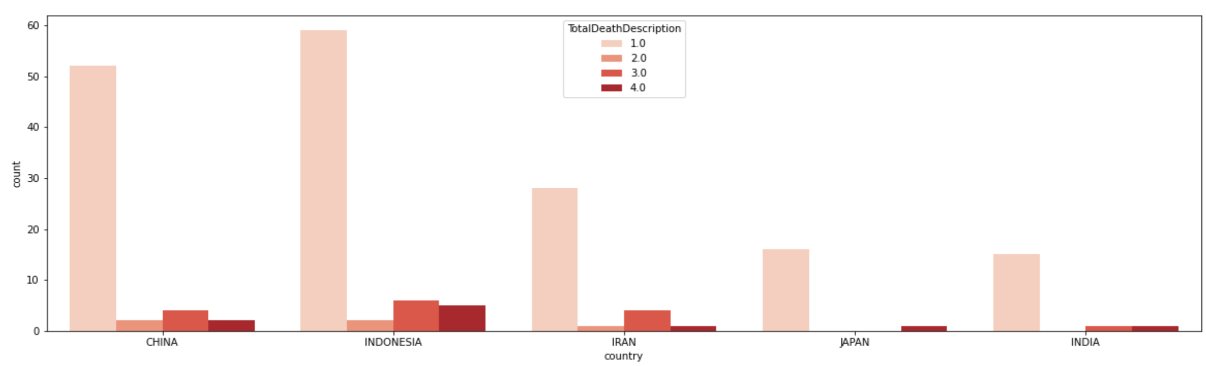


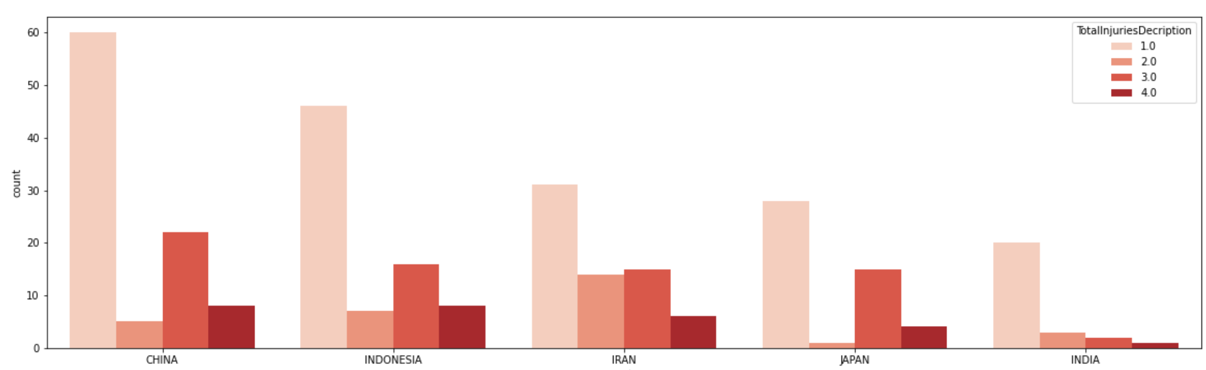
First to explain the values of this feature it is a code to the number of deaths distributed as follows (1 indicates 1 to 50 deaths, 2 indicates 51 to 100 deaths, 3 indicates 101 to 1000 deaths and 4 indicates more than 1000). And this is the same code distribution for Total Injuries Description. But, for Total Damage Description there is a difference this feature describe the damages caused by the earthquake in terms of millions of dollars so it is distributed as follows (1 indicates less than 1 Million, 2 indicates $1 to $5 Million, 3 indicates $5 to $25 Million and 4 indicates more than $25 Million)

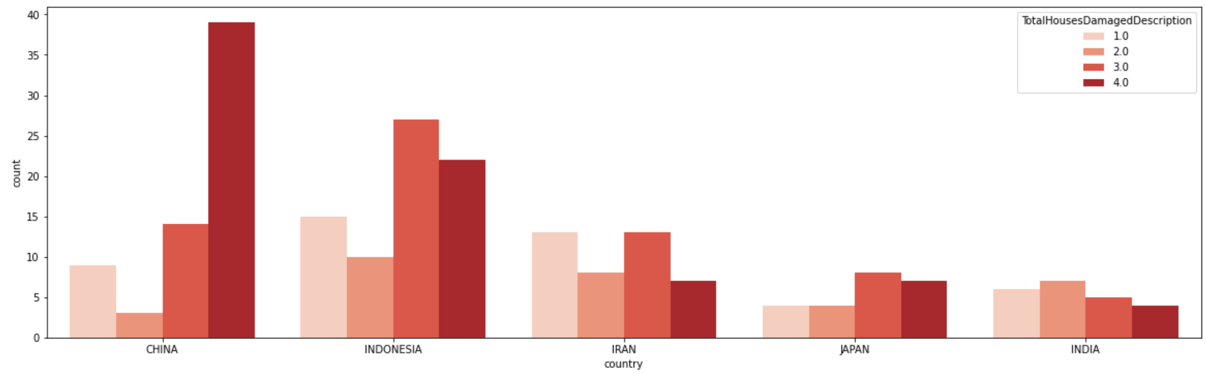
So, as we see most of these events caused the death of at most 50 people and 50 injuries and the damage was about $1 million.

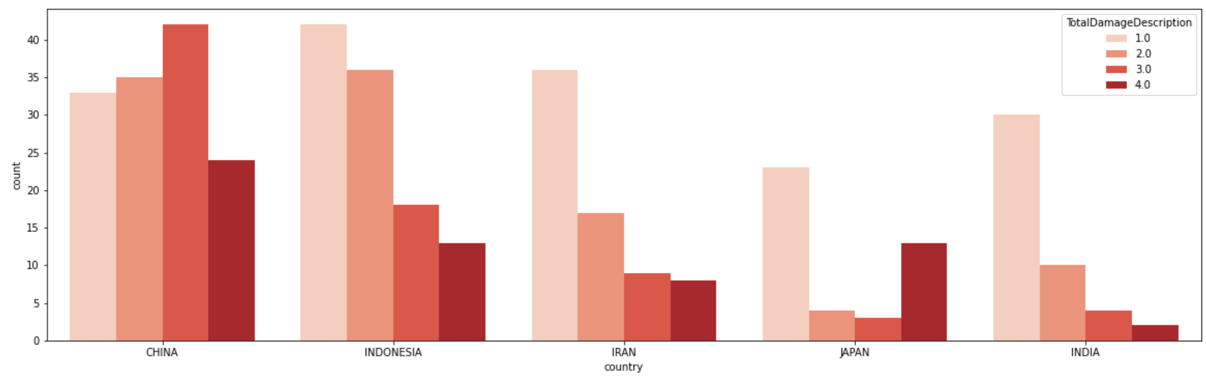


When investigating the Total Houses Damaged Description we found a different result than the other casualties as we found that if the event was so strong to damage houses it would damage a large number of houses which could be over 1000 houses.









In these four plots we describe the most 5 countries that suffer from casualties due to earthquakes. We noticed that there are two countries that have appeared in the top 5 countries that witnessed a lot of earthquakes in the 21st century those countries are Indonesia and Japan. Also, we could see that China suffers from a house damage problem as most of the events there cause the damage of more than 1000 houses.

5.3. Conclusion

Earthquakes could be devastating so we need to understand more and more about it to protect ourselves and try to save people's lives.

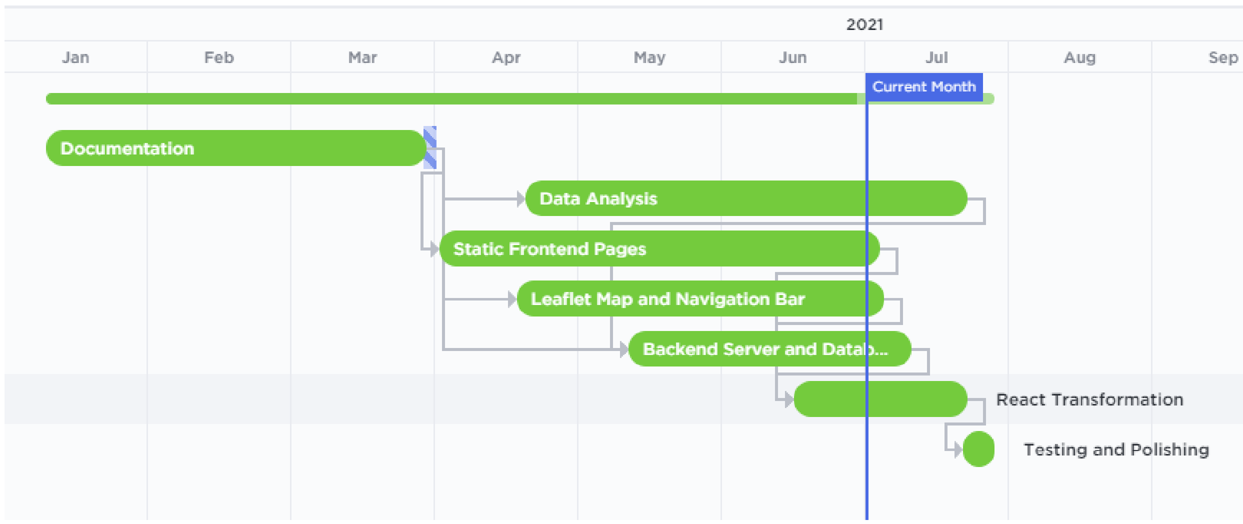
We wanted to contribute to this with our knowledge and skills so we started to analyze data about this disaster, predict some of its aspects and design a web application to display what we achieve.

Our analysis results could let people know more about earthquakes and to be more aware of how dangerous this disaster is so they realize that they have to be more cautious. Also, these results could help governments to understand that earthquakes cause a huge damage specially on houses so they can take steps to make stronger building structures. Our prediction results also will help everybody to let them know if an earthquake will happen, how strong it could be. So, we all have to focus more on natural disasters and know more about them to save our lives.

Appendix A: Work Distribution

Task Name	Resources
Requirement Phase	
Introduction	Abdelaziz Mohsen - Marwan Ahmed - Mohamed Khaled
Overall Description	Abdelaziz Mohsen - Marwan Ahmed - Mohamed Khaled - Seifeldin Mohamed - Youssef Hassan
External Interface Requirements	Abdelaziz Mohsen - Mohamed Khaled
System Features	Seifeldin Mohamed - Marwan Ahmed
Non Functional Requirements	Marwan Ahmed
Design Strategy	Youssef Hassan
User Analysis	Youssef Hassan
Data gathering	
Collect data	Seifeldin Mohamed - Youssef Hassan
Study data	Seifeldin Mohamed - Youssef Hassan
Preparation Phase	
Search for technology	Abdelaziz Mohsen - Marwan Ahmed - Mohamed Khaled - Seifeldin Mohamed - Youssef Hassan
Search for appropriate model	Abdelaziz Mohsen - Marwan Ahmed - Mohamed Khaled - Seifeldin Mohamed - Youssef Hassan
Design Phase	
Use-case	Seifeldin Mohamed
UML	Seifeldin Mohamed
BPMN	Seifeldin Mohamed
Choose Database storage	Abdelaziz Mohsen - Marwan Ahmed - Mohamed Khaled - Seifeldin Mohamed - Youssef Hassan
Data Analysis and Prediction Phase	
Data wrangling	Seifeldin Mohamed - Youssef Hassan
Data Visualization	Youssef Hassan
Prediction Model	Youssef Hassan
Implementation Phase	
Backend Server	Seifeldin Mohamed
Database	Seifeldin Mohamed
Static Frontend	Abdelaziz Mohsen - Mohamed Khaled
Leaflet Map	Marwan Ahmed
Navigation Bars	Marwan Ahmed
React Transformation	Seifeldin Mohamed
User Authentication	Seifeldin Mohamed
Testing Phase	
Security Testing	Marwan Ahmed
Testing and Polishing	Marwan Ahmed

Appendix B: Time



Appendix C: Technology Used

Name	Description
Tools	VS Code, Bracket, MongoDB Atlas, Github, Jupyter Notebook
Languages	Python, HTML5, CSS3, Javascript
New Technologies	Node.js, React.js, Predictive Analytics, Tableau, Leaflet

Appendix D: References

- https://www.seowebsitedesign.com/the-waterfall-model-of-software-development/
- https://www.ngdc.noaa.gov/hazel/view/about
- https://earthquake.usgs.gov
- https://www.adobe.com/mena_en/products/xd.html
- https://app.creately.com/diagram/
- https://mongoosejs.com/docs/guide.html
- https://docs.mongodb.com
- https://leafletjs.com
- https://react-leaflet.js.org
- https://reactjs.org
- https://blog.jumpstartinsurance.com/can-aftershocks-be-predicted/