Software Design

GeoLearn

2/14/2020



Mentor: Fabio Santos

Sponser: Dr. Chris Doughty

Team members: Samuel Prasse (lead), Kaitlyn Grubb,

Tyler Pehringer, Joshus Tenakhongva

**Table of Contents**

1. *Introduction…………………………………………………………………….pg3*
2. *Implementation Overview…………………………………………………....pg4*
3. *Architectural Overview……………………………………………………....pg5*
4. *Module and Interface Descriptions………………………………………...pg8*
5. *Implementation Plan………………………………………………………...pg14*
6. *Conclusion…………………………………………………………………...pg16*

**1.0 Introduction**

Throughout the years, the Earth's environment has changed drastically and this is due to different reasons. There are many studies today that discuss what has changed throughout the years in terms of how we as a species has changed it and what we can do to make the world around us better to live in. There is, however, one glaring issue; we struggle to teach this information to our children. Currently, teachers face many challenges trying to convey this information to students.

These challenges being:

* How to make sure students understand this information.
* How to make sure that students retain this information.
* How to interest students into this information.

Currently there are applications that teachers can use to make this information easier to understand, but they lack hitting all of these concerns. Teachers may not have the time to create easy to understand lectures that help convey all information needed to allow students to understand to its fullest what is changing in their environment and why. That is where our project comes in.

We are currently working on an application that takes in environmental information from the internet and formats that information into Google presentations for teachers automatically. These presentations will discuss different forms of environmental change in that person’s area and simplifies that information so that it is easier for younger students, mostly in an elementary school setting, to understand this information. We want the teacher to be able to type in their school info, what topic they are looking for, how difficult the content should be and what kind of activities they want in their presentations to help their students learn. We want the teachers to e able to take from 3 different topics:

* Biodiversity: This topic discusses the different animals that live in the environment around the user. It will discuss the info on different animals around them, such as:
	+ Their sizes.
	+ What they like to eat.
	+ Which are extinct.
	+ Etc.
* Land-Use Change: How humans have changed the environment around them through using the resources around them to develop housing and other structures in that nature.
* Climate Change: How the changes in the environment has created a steady change in the temperatures on Earth and how that affects the environment around the user.

**2.0 Implementation Overview**

 For most teachers, their workflow for creating a lesson plan looks like this:

* Research and follow the curriculum
* Create a plan
* Create presentation material and interactive material

Teachers aren’t paid enough to be spending this much time creating these lesson plans for their students. That is why we want to streamline the process and combine all of these steps into one easy step, using GeoLearn’s services to create a slideshow that will contain an abundance of interesting, local information with interactive content that will aid in students’ retention of the material. 

FIGURE 1: The work flow of Geolearn

 For GeoLearn, we will be using several technologies that will all perform different important tasks:

 Using satellite data from Google Earth Engine gathered over decades, we can create time-lapses of the earth that can show significant change. This is where we will be gathering the Climate Change and Land Use Change data from. The time lapses will be presented in gif form in the slide shows for ease of use.

For the Biodiversity categories, we will be using our own SQL database to store all of the mammal information. Currently, we are using the database from a previous Dr. Doughty Capstone, Ecolocation. This database has a plethora of animal information, but some of it is incomplete, and we cannot edit it. We will risk breaking the Ecolocation App. That is why we will make our own SQL database.

 For our website, we will use the Django framework. It allows us to use our Python scripts, connect to our SQL database directly, and it gives us a lot more tools to craft our website. From the beginning, our chosen programming language was Python, and allowing us to use it without website will provide us with the libraries, support, and functionality we will need.

 We will be using Amazon Web Service (AWS) to host our website and our database. Many of the university’s services are already hosted on AWS, proving that it is a service the university trusts and relies on.

 Our slide creation software is built in Google App Scripts. These scripts are what create the slideshows. We have a Google Drive account that contains the slide templates provided to us by our clients, and the script populates them with the appropriate data that is generated by our Python scripts.

 To provide the Google App Script with the information it needs, the data will need to be pushed to our server’s Google Drive account. To do this, we will use the Google Drive Python API. Using this, we will upload the appropriate data to the server Google Drive.

**3.0 Architectural Overview**

The technologies described above will allow us to develop a flow for our software by studying the functions associated with the technologies. This is an overview of how all parts of Geolearn interact. Please review our Technological Feasibility Analysis document for more information on why these technologies were chosen.



FIGURE 2: A figure showing the components of Geolearn.

3.1 Technologies

 Most technologies used are provided by Google. This provides uniformity and security to our software. The technologies will be discussed in the order they appear in the image above.

**AWS Server:** This is a server that will hold our developed code and a database created by a past capstone. This allows our website and code to be available to teachers anywhere with internet connection.

**Software:** The software we are developing is the website and access point for slide generation. This code is a website built using Django Framework that holds the slide generation for,m, interactive game page, and user logins.

**Ecolocation Database:** This database stores shapefiles and information about large mammals. Shapefiles is information about where that animal can be found on a map. This allows Geolearn to pull information on mammals based on location and insert it into a slideshow.

 **Google Earth Engine:** This is a database that holds satellite images and data such as temperature, tree height, and more. This information is used to create time lapses of satellite pictures and graphs for our localized presentations.

 **Google Drive:** Google Drive is a cloud storage platform that comes with a google email account. The primary job of Google Drive in our software is to collect and present the information gathered. Google Drive holds all of the templates before the localized information is input. In addition, a folder of the localized data will be stored in Google Drives in order to pass this information into the templates. Lastly the presentation will be presented in a Google Slides presentation stored in the user’s Google Drive.

3.2 Interaction Between Technologies

This will further describe the arrows in the figure above. They are listed in order from left to right.

 **1. Interaction Between Software and Ecolocation Database:** Once the software determines the information that is needed it will run an SQL script to get data from the Ecolocation database. The SQL script will get animal information based on location.

**2. Interaction Between Software and Google Earth Engine:** Once the template is chosen by the software, a python script will access Google Earth Engine and run code scripts to get satellite information based on location. This python script will run multiple other python scripts on different satellite datasets such as fire data and climate change data.

**3. Interaction Between the Databases and Google Drive:** The information gathered will be stored in a folder in Google Drives. Information will be formatted to best fit the data pulled. Ecolocation data will be stored in a Google Spreadsheet to allow the graphing of the information such as number of animals or weight of a specific animal over time. Timelapses pulled for Google Earth Engine will be stored as Gifs. Easily graphable information pulled from Google Earth Engine will also be stored in a spreadsheet for easy manipulation such as graphing.

**4. Interaction Between Geolearn Google Drive and User Google Drive** The Google Drive information will be inserted into a slide template using Google App Scripts and the template decided by the slide creation page. This script will correctly format any spreadsheet info and insert any pictures or timelapses in the Geolearn Google Drive Folder. After this the filled template will be placed into the user’s Google Drive.

**5. Interaction Between Google Drive and User:** The Google Slide Presentation can be accessed by the user by logging into Google and opening their drive. The slideshow is then ready to present.

3.3 Influences

 The architecture of this software is highly influenced by Google and their authentication services. In order to not have to keep passing in foriegn data to Google, for example a powerpoint presentation, all of our interactions after gathering data are all Google interacting with Google. This keeps our code and interactions between pieces uniform and familiar. More databases and information can be added as long as that information can be passed to Google Drive. This allows for expansion into more subjects like history.

 In addition, Google authentication guarantees we can pass information to users securely. Not only is it secure but many school accounts are linked with Google. This allows teachers and students to keep all of their information in one place. Google allows for a safe and effective way to format and distribute information to both teachers and students.

**4.0 Module and Interface Descriptions**

The following modules are the key components of our software application and are shown in the figure below. Each one of them performs one or several vital tasks in execution of our code. These components consist of a web server, SQL database, and Google API interaction. In this section the interaction between each other and the software component will be explained to give insight into how the application functions as a unit. This begins with a short natural-language description of each component which transitions into several diagrams and then an in depth look at the services that the software application provides the user.



FIGURE 3: A figure showing the components of Geolearn with functions.

4.1 Web Server

The first component of our software system is our web server. Our site will be hosted using a product of Amazon Web Services known as Amazon EC2 (Elastic Compute Cloud). The cost-effectiveness and the security of the Amazon servers is what drove us to utilize them in our project. Animaps, which is a previous project completed by Dr. Doughty and a past capstone team hosts their site and database on these same servers which will help our team in terms of transferring the critical data that we will be implementing in our web application. The process of implementing the website is a step-by-step process outlined perfectly in Amazon documentation.

EC2 is perfect for the needs of our team. It provides us with a virtual machine, SSD (Solid State Drive) storage, data transfer, DNS (Domain Name System) management, and a static IP (Internet Protocol) address. EC2 will allow our team to create an instance of our software on Amazon servers which will run the Django framework we have built our website with. Once we have transferred the code to the server using Amazon’s automated process, our website instance will be live on the web within minutes. At this point, we will have a fully functioning website containing our application with all of its required implementations.

This web server will be the home of the GeoLearn project. The homepage will have several different tabs to direct the user to the many types of tools we have created. The most important tool we have created is the slideshow creation tool. This is what will be most utilized by users when visiting our website. Users will be able to select a location (this can be a school or any other location of choice) from a Google Maps API implementation. Once the location has been selected, the user will be able to select the type of presentation they are interested in teaching or simply learning about (Climate Change, Biodiversity, Land Use Change, Fire). The user will also have the option of selecting the difficulty level of the presentation (Beginner, Intermediate, and Advanced).

Our web server will then run the back-end code to create the presentation with the selected parameters and because the user has logged in with their Google account, the presentation will momentarily appear in their personal Google Drive. To complete the task of creating the presentation, the back-end code takes advantage of our database information, and several Google API’s. Our database information is queried depending on the geolocation of the user. This information is passed to the database when the user enters their location of interest in the Google Maps API on the website. [1] Then, depending on the presentation parameters, region-specific animal data, climate change data, land use change data, or fire data is then placed into the presentation skeleton at specially designed points to ensure the user is getting accurate and interesting data, as well as a sleek presentation tool.

4.2 Database

 Our database will be implemented using Amazon RDS (Relational Database Service). Amazon Web Services were once again utilized for the database because they gave our team the option to host the site and the database with the same company’s resources. The Animaps team, a prior project sponsored and overseen by Dr. Doughty, stored their data in an Amazon database which will make transferring the data that our team needs from their database to ours very simple. This database will be implemented using the MySQL database engine. The purpose of this database is to contain shapefile information on animals all over the world. It will be utilized the most when the user selects the Biodiversity slideshow presentation type. Dr. Doughty and his team have many of these files available which gives the user the ability to explore data not readily available to the general public about animals they might not have known to exist. We are storing these in a relational database because each file has hundreds or thousands of identifying numbers that are much easier to query in a database than to search for using a looping method imposed on a spreadsheet or Comma Separated Values file.

As explained in section 4.1.1 ([1]), our database will be queried once the user has chosen the Biodiversity presentation type and entered the difficulty level of their desired presentation. Depending on the location selected (based on latitude and longitude pulled from the Google Maps API), the database will return the corresponding shape files in the area of 5 to 50 kilometers. Our back-end code will search for pictures of the animals by accessing a website called ArkHive. This site contains high definition photos of many types of animals which will be stored temporarily on the local machine and then inserted into the appropriate place in the presentation.

4.3 Google Earth Engine

 The Google Earth Engine is one of the most diverse and exciting tools the team is implementing in this project. It is how we gather satellite data of fire, climate change, and land use change. Earth Engine allows us to take advantage of Landsat, Sentinel, and many other satellite data with precision down to a specific date and time. If a user selects land use change, fire, or climate change in the slideshow creation parameters, then Earth Engine is going to be utilized. Our team has created graphs, time lapses, topographical maps, and photographs showcasing the amount and quality of raw data that this API presents us with. Once the location and topic of presentation has been selected in the website, these parameters will be inserted into variables in the code to generate satellite, or graph data about the location on the specific topic. When the data has been generated it will be stored in a Google drive, then inserted into the skeleton presentation at critical points to keep students hooked and invested in what is being presented. This tool is allowing our team to gather and present interesting and relatively unknown data with students and teachers around the world.

4.4 GeoLearn Drive

 The GeoLearn Drive will be home to every presentation skeleton we implement on the website. These presentations were designed to be the outline of our geolocation presentations which provide the user with location specific information based on the users whereabouts. The code will run from the website when the user selects their type, difficulty, and location of presentation. Then, the Python code will obtain the relevant Google Earth Engine data or Ecolocation database data and return the data to the GeoLearn Google Drive. Next, a Google Apps Script is run to gather this data and compile it into a brand new presentation which is then sent in an editable format to the users validated Google Drive.

4.5 User Google Drive

Upon visiting the site for the first time, users will be required to log-in using Google Sign-In. Once their account has been verified they will be prompted for access into their Google Drive. Once access has been granted and the user would like to make a presentation, they will select a location, type and difficulty which will run our server code and begin the creation of the presentation. Once the presentation has been created in the GeoLearn Drive (4.4), it will be dropped into a specified location or folder within the users Google Drive. This gives the user instant access to the presentation to display it in a classroom, or for personal use. Because the presentation is in their Drive, the user has the ability to add, remove, or change any sort of information they feel necessary.

4.6 Diagram with More Function Detail



FIGURE 4: A figure showing the components of Geolearn including functions within the components.

4.7 Public Interface

As shown in diagram 3 above, we have 4 components to Geolearn. The web service component, database component, and Google API component cannot be broken down as these just hold data. The web service component holds our software and makes our website available for access. Both the database component and Google API component hold data and are interacted with by making queries. Due to the simplicity of these we will take a deep look at the software component.

4.7.1 Gathering Slide Information

 These functions will deal with gathering data from a form and passing this information to generate slideshow.

**get\_subject() → string subject**

 This function will retrieve a string from a dropdown and return this String. Currently there are three options for the string: biodiversity, land use change, climate change. This string is used to determine what template to follow when generating slides.

**get\_location() →[int lat, int long]**

This function will retrieve a coordinate pair holding the latitude and longitude entered by the user in the create slides form. This pair will be passed to the databases to limit data to only a specific area.

**get\_difficulty() → string difficulty**

This function will return a string value from a slider. Currently there will only be three types of difficulty. Easy, intermediate, and hard. This represents how interactive the slides will be and what kind of interaction the slides will have. This in addition to the subject decided which slide template gets filled.

**generate\_slideshow(string subject, [int lat, int long], string difficulty) → int generated**

 Subject and difficulty will decide which template is going to be filled. Based on this the latitude and longitude array will be passed to the databases to gather the information needed to fill the slides. After this the data is manipulated and reformatted by Google as described in section 3.0 Architectural Overview

4.7.2 Formatting Slides Google Functions

 After the data is deposited into Google Drive it can be accessed and put into the Slide template. These functions are provided by Google and further documentation can be found at https://developers.google.com/apps-script/overview. Important functions used to fill the templates include:

**DriveApp.getFoldersByName("name").next()**

This is used to view the data saved in a Google Drive folder. The data can then be iterated through by using .next().

**image\_list = slide\_list[slide\_number].getImages();**

**image\_list[0].replace(data\_folder.getFilesByName("name.jpg").next());**

This is a code snippet used to get all the images on a specific page and then replace the first image with an image found in the google drives.

**slide\_list[5].insertSheetsChart(chart, position.left, position.top, size.width, size.height);**

 This inserts a chart made from Google Sheets into a slide show.

4.7.3 Formatting Slides Script Functions

 These are the functions that utilize the Google functions mentioned above,

**copy\_template() →int ID**

This function will copy a slideshow presentation and return the ID of the new presentation. This allows the reuse of template slides stored in the Google Drive.

**main() → none**

This function will format and place a slideshow in the user’s Google Drive. This slideshow will have all of the local data inserted.

**5.0 Implementation Plan**

****

MVP

* Everyone: Before this step, we all had our individual scripts that we were responsible for working separately. Now, we want to combine them into one singular piece of software.
* Kaitlyn and Tyler: At this point, we want the skeleton of our project working. From the website, a user should be able to link their Google Drive account and have a slideshow generated and shared to their account
* Joshus and Sam: So far, we have been tasked with getting the Biodiversity and Land Use Change slide categories working. These will be available in the MVP
* Kaitlyn: Our Google Drive API scripts will ask for authentication from the user, so we may use their account to share the slideshow with.

Alpha Build

* Kaitlyn: We will be altering our slideshow building scripts to be able to generate the final versions of the biodiversity and Land Use Change slide templates given to us by our client, Dr. Doughty. We will also begin implementing the Climate Change slide templates
* Sam: In order to create the climate change slides, we will need to create a new script to draw information from Google Earth Engine
* Tyler: Due to website creators such as Wix not allowing us to access the core files to the webpages as they are proprietary, we will have to create our own from scratch. We will transition to a Django server instead. Continuous improvement of the user friendliness of our application as well as the aesthetics of the website is a large priority.
* Joshus: We are using the database from the previous Dr. Doughty capstone, Ecolocation that contains a large amount of animal data. Unfortunately, it does not contain any images of the animals despite the Ecolocation application using animal imagery. We will have to find our own way to gather animal images, download them, then use them to populate our slides.

Beta Build

* Kaitlyn and Sam: For our Beta build, we will have made our slide scripts for the climate change category robust, and we will have implemented the final version provided by our client, Dr. Doughty. All of our slide templates will have been tested by Andrew.
* Joshus: We are currently using the Ecolocation application’s database for animal data. Since their app relies on this database, we shouldn’t change it, potentially risking rendering their app broken. We need to create our own database, so GeoLearn can be changed, reorganized, and expanded to fit our current needs and the needs of the developers that will maintain the GeoLearn project.
* Tyler: The majority of our software will be built using Python. Since our scripts are built to work alone, we will need to unit test them to ensure their function is as robust as possible.

Release Ready Build

* At this point, our project should be mostly completed.
* Joshus, Sam, and Kaitlyn: We will be conducting more testing of our scripts and looking into making our database and website more secure.
* Tyler: We will also be continuing to make the user interface more aesthetically pleasing and increasing ease of use. We will also be increasing the security and stability of our server

**6.0 Conclusion**

 What we want to achieve is making sure that users are able to get comprehensive and factual presentations while making sure that students of all ages can understand and retain the content. We believe that we can create a web application that can fill in the gaps of information between researchers and students of all ages. We will be using the Google Earth Engine and the Google Drive Api to allow us to get the environmental information. We will also be using a previous capstones (Ecolocation) app that gives information on animals that live in the area near us. We will gather information from both the Google Earth Engine API and Animaps (Ecolocations capstone project) to give us the information needed. We will be using an AWS server to host the application. Based on our schedule, we are hoping that we will have an alpha build by the end of this week. We hope to have our MVP done by the end of the month and we are hoping to have more features and a better design for the product going on for the rest of the semester. With the technology that we are using, we believe that we can achieve this for the users this application is used for. With this application, we have the potential to change the world for the better.