GIS to Assist in Real Estate Transactions Using ESRI APIs

Tajh Hetrick

GISC 5200K Programming for Geo Sci & Tech

Huidae Cho

Abstract:

In the age of rapidly evolving technology many different systems and applications service our daily needs. As the exponential growth of this potential continues so must the application to different aspects of life. Geographic Information Systems (GIS) are currently used in every industry imaginable. Whether it be from simple use of GPS to in depth spatial analysis GIS has a footprint. Yet despite the growth of GIS many industries underutilize its potential when it could provide a completely new way of doing business. One such industry is real-estate. While GIS is certainly used in the real-estate industry the potential for GIS can provide much more than is currently utilized.

In this paper we attempt to lay the foundation for a real-estate application that could manage the entire process from cradle to grave. We performed research on recent methods as well as some potential future features. Research was also conducted on Python and JavaScript ESRI ArcGIS APIs and then later compared in implementation. A foundation is presented for potential application server/client infrastructure. Unified Modeling Language diagrams show a foundation for the potential application end result. Finally, we discuss our results and present last thoughts as well as the future work necessary to realize the next iteration of the application.

1. Introduction:

In a rapidly changing society and with technology growing exponentially many industries are forced to adapt. The addition of computing technology and tools has changed the way we do business on almost every level. Some industries however still rely on older technology and methods to achieve their business goals. This in some cases may be due to necessity of legacy systems that are in use. In some cases, technology exists that could improve those industries if properly adapted and utilized to benefit those industry’s needs.

One such example of an industry underutilizing potential resources is real-estate. Geographic Information Systems (GIS) have the potential to change the way business in real-estate is conducted and potentially even boost revenue streams or create entirely new ones. While some real-estate driven applications do exist, they are generally not very robust and also are underutilizing what GIS is capable of accomplishing. In many cases the GIS just serves to provide location data and perhaps some listing data but not much more. This pales in comparison to what GIS can provide in terms of visualization and spatial analysis. To that end a real estate application could provide more than just a buy and sell tool to facilitate transactions. The potential exists for a GIS to also assist with city planning, land use management, parcel management, designing new property and modifying existing property. With enough resources and a well-built team, a GIS application could be a full comprehensive tool that could assist with the entire process of real-estate from finding the location to designing and building structures.

GIS has the potential to provide a lot of information useful to realtors, investors, and buyers alike. Compiling and presenting this information in a well-designed easy to use interface could save a lot of time and money for almost any part of the industry. A customer looking to make a purchase could look at accurate up to date pricing models and practices from their phone or computer. Sellers would be able to manage their firm's rates and provide price comparison to their clients using the same system. City planners and government officials could also collaborate better with updated information that would reflect changes on a daily or even hourly basis. This information would also be crucial for investors in real-estate looking to trade based on property values and projections.

To address the needs of any system we must look at the requirements of said system. To attempt to scope the requirements of a system like this we will discuss research conducted in the real estate industry with GIS. We will also discuss the potential for GIS to bring robust features to a real estate application. A logical place to start is real estate investment. While investing itself is different than your typical buy/sell market, it encompasses almost all of the same requirements a buyer or firm may have. This will provide us the ability to scope multiple user roles at once.

Like most business processes real estate investment modelling is essential (Belay, 2006). There are three kinds of models: Process, Market, and Value Model (Belay, 2006). Conveniently the process model can be broken down into 11 different phases which happen to also serve as a solid list of things a GIS can assist with handling (Belay, 2006). The phases are as follows: Real estate register, Planning, Marketing, Expansion, Building/construction, Utilization Management, Real estate sale, Valuation, Investment, information flow, Validation of interests (Belay, 2006).

When discussing investment, it is essential to consider the interests of the investors. In many cases these interests also fall in line with property owners looking to sell and individuals looking to purchase. The value of a real estate is determined by groups of factors such as Location, Position within the building, and the technical conditions/internal structure of the real estate (Belay, 2006). Among the most important aspects for investment are: Quality of assets, Compound of tenants, average period of lease, conditions of contracts, real rental fees and mortgage rates, cost of facility management and last but not least yield (Belay, 2006). At a glance it is easy to see how a GIS presenting up to date parcel, visual, and market information could assist investors, buyers, and sellers in the decision-making process.

Naturally, geographic location as well as organization is extremely important to the real estate process. We use parcels to help organize space for purchase and sale and naturally we have to have ways to identify these parcels. GIS can be effective in this respect as it has the ability to draw a layer showing all of the parcels in a region as well as make them selectable. One of the values of this is the ability to provide meaningful labels and valuable information such as state or federal statute (Donlon, 2007). This however presents us with some of our first obstacles when trying to compile this information. A large part of data management is cost and just a few regions worth of data can cost up to $2,000 just to attain (Donlon, 2007). This does not account for organization and in many cases, data will come in different formats that must be converted for effective use (Donlon, 2007). One of the potential uses for our GIS could be to provide more reliable and consistent data and formats that retain more portability with other GIS. Providing accurate labeling and parcel information as well as easily accessible data will assist in assuring the customer of quality service with reliable knowledge, experience, and reputation (Donlon, 2007).

Along with parcel identification comes parcel management, organization, and utilization. One of the issues with parcel data is ca be acquisition, and the static nature of collecting data in between long intervals of time. Without consistent updates things may change in one place and not be updated in another creating inconsistencies in data. Efforts could be made to keep data as consistent as possible but in the event there are wide gaps in data GIS can assist in tracking that change. Wu Bin, et. Al, performed research on determining the optimal parameters to detect parcel-based change (Bin, et. al, 2016). While this does not solve the issues of consistent data acquisition it can assist in finding the change in data given multiple data sets. This information can be used to show changes in the region over time and could assist in the marketing aspect of real estate as well as purchasing. While this does not fully account for all parcel related issues that may be faced it does show the strength GIS can possess in solving or mitigating these issues.

Once a parcel is selected another significant portion of determining the value is the buildings or potential for buildings in that space as well as simply, land use. In some cases, we may find a history of land use that may not be suited to future development. Land cover types such as boundary, impervious, tree, soil, water, etc. can be determined using multiple techniques such as aerial imagery and can change over time (Elgaali, 2018). An example of this could be farmland that is being considered for repurpose or new crops. Farms generally already have infrastructure and storage buildings that are specific to those occupations and may not be beneficial when building a new office building or residential neighborhood. Being able to view this information and make these determinations all in one place would be useful. In China a group of researchers designed a simulation in GIS that helped with the classification of land use using the Markov model (Sang, et. al, 2011). They were able to use GIS to determine land use change over a period of time and provide accurate reclassification and analysis of that land use change (Sang, et. al, 2011). In a more urban and smaller study Elgaali Elgaali was able to use Aerial photography to assist in determining lawn irrigation needs for residences in Colorado (Elgaali, 2018). Another similar use of GIS is from Andres Sevtsuk and his colleague Michael Mekonnen. They used GIS to perform urban network analysis assisting in urban spatial analysis (Sevstuk, Mekonnen, 2012). This toolbox gives architects, planners, Geographers, and scholars the ability to measure the accessibility and centrality metrics on a spatial network (Sevstuk, 2012). With the assistance of the San Diego

Association of Governments urban GIS applications assisted with land use classifications and city planning that improved the cities ability to prepare for future projects (Parrott, R. 1991). These types of tools can be combined to provide the crucial asset management information necessary to make informed real estate decisions about a piece of land.

Next, we will discuss different ways of visualizing space many of which include 3-Dimensional (3D) modeling techniques. These techniques have been used in various ways for different reasons but can certainly be adapted to be effective in a real estate application. The use of photogrammetry is common when developing 3D environments in GIS. There exists a range of sensors that can be used to capture and process imagery that can then be compiled into accurate data used for spatial analysis (Gruen, A. 2008). Perhaps most commonly aerial and satellite imagery is used though sensory equipment can consist of Film Cameras, Unmanned Aerial Vehicles, Special charge coupled devices with complementary metal-oxide-semiconductor cameras, video cameras, camcorders, linear array cameras, digital panoramic, electronic imaging, and assorted hybrid systems (Gruen, A. 2008; Lee, A. 2019; Scianna, A. 2018; Wang, T. 2014). Combining these types of tools with highly accurate Geometric objects like the ones used in BIM could create a platform for construction and analysis that would prove vital to the real estate process(Fosu, R. 2015). The integration of BIM and GIS has a range of applications such as Climate adaptation/Energy Analysis, View quality and Shadow effect analysis, utility visualization, facility management, spatial querying/location and space navigation, planning, emergency situations/natural disasters (Fosu, R. 2015). These 3D representations of terrain and buildings can greatly assist with visualizing and even exploring a location without having to be present.

The application of 3D environments has proven their usefulness, yet some groups have taken it even further and created virtual environments for exploration. These environments frequently incorporate the ability to navigate 3D rendered images in virtual reality from the first-person perspective. David Koller and a handful of his colleagues designed a real-time 3D GIS that was tested by the US army (Koller, D. et. al, 1995). In their environment users has a number of different navigation methods including a “free flying” method that allowed users a great deal of perspective to analyze environments (Koller, D. et. al, 1995). Scianna A. used virtual environments to recreate cultural heritage and historical locations. Using a combination of aerial imagery, photogrammetry, WebGL, and 3D meshes derived from a point cloud the group was able to recreate a virtual model of a castle that allowed for accurate walkthroughs and flyovers (Scianna, A. et. al, 2018). Hui Lin and his team analyzed the progression of virtual geographic environments and have found a few relevant use cases for virtual environments such as creating multi-participant-based collaborative geographic experiments (Lin, H. 2013). The range of use of 3D and virtual environments is evident and could prove invaluable to the future of real estate.

Given the wide range of uses of GIS and the many different possible applications real estate is an obvious industry to develop and apply the capabilities of GIS. This paper will focus on laying the groundwork for a real estate focused application that will combine as many of these features as possible while developing a user-friendly interface. We will design and implement a demo that has at least one BaseMap, parcels, data pulled from selection of features, address search, and an example of 3D BIM. This will illustrate the potential of GIS related real-estate applications that will serve as a launching point for future iterations and projects.

2. Materials and Methods:

2.1 Background GIS

When choosing the correct GIS multiple different systems were researched. While ESRI remains the industry leader in GIS and GIS related services, it is still important to research open source options to try and maximize development potential. Most of the related research used a variation of GIS some even combining use of GIS. In the future it makes sense to try and adapt this application to as open source as possible with as many different GIS that can handle it. While we will still discuss other researched options; with the current availability, skillsets, and knowledge I elected to use the ESRI APIs.

The World Health Organization(WHO) has done research on various GIS that can be used by their external partners(WHO, 2018). The WHO itself has a contract with ESRI for unlimited use of ArcGIS (WHO, 2018) but wanted to provide options for some of the smaller organizations they work with who may not be able to feasibly maintain such a contract. The research included stand-alone GIS applications as well as analytical tools with GIS/Mapping capabilities that consisted of QGIS, GRASS, R, GeoDa, Epilnfo, and Healthmapper. Out of these options QGIS(Quantum GIS) and GRASS (Geographic Resources Analysis Support System) were the most robust and most promising applications. QGIS however proved to be the most suitable option based on criteria such as user-friendly GUI, cartographical operations, Data manipulation and analysis, Basemap availability, etc. (WHO, 2018). While QGIS and GRASS compete closely on these criteria QGIS proved to be the more user-friendly option giving it an edge, the comparison is seen in the below **Figure 1**.

Created on: March 12, 2018
Author: WHO/HQ/WHE/HIM/MAP (GIS team)
WHO, Geneva

		QGIS	GRASS	R	GeoDa	EpiInfo	HealthMapper	ArcGIS*
Supported by all common operating system		fully supported	fully supported	fully supported	fully supported	partially supported	partially supported	fully supported
User-friendly GUI		fully supported	partially supported	not available	fully supported	fully supported	fully supported	fully supported
Cartographical operations		fully supported	partially supported	partially supported	partially supported	partially supported	partially supported	fully supported
Supports common data formats		fully supported	fully supported	partially supported	partially supported	partially supported	not available	fully supported
Data manipulation and analysis	Spatial analysis	fully supported	fully supported	fully supported	fully supported	fully supported	not available	fully supported
	Manipulate geometry	fully supported	fully supported	partially supported	not available	not available	partially supported	fully supported
	Statistical analysis (e.g. producing graphs and charts)	partially supported	partially supported	fully supported	partially supported	fully supported	fully supported	fully supported
Integration with other open-source GIS platform		fully supported	partially supported	partially supported	not available	not available	not available	partially supported
Basemaps		fully supported	not available	partially supported	not available	partially supported	not available	fully supported
Scripting functionality		fully supported	partially supported	fully supported	fully supported	fully supported	partially supported	fully supported
Training sources and technical support		fully supported	partially supported	fully supported	partially supported	partially supported	not available	fully supported

fully supported partially supported not available

* A paid license is required.

Figure 1. Open-GIS tool comparison (WHO, 2018)

When comparing QGIS to ArcGIS the difference in potential is obvious highlighting the necessity for more robust open source application and tool development. One of the issues with open source development can be the general lack of documentation and inconsistent use and recording of design patterns (Balaguer, 1997). ArcGIS provides a complete GIS application suite from ESRI (WHO, 2018). With the inclusion of applications such as ArcSDE, ArcGIS Desktop, ArcGIS online, and ArcGIS server most conceivable workflow is achievable using ArcGIS. With QGIS being mostly developed as a Desktop GIS application it suffers from certain limitations that may require outside tools (WHO, 2018).

ArcGIS also has extensive Documentation and online training resources (WHO, 2018). This also happens to be the primary reason I selected it to attempt to build an application demo for this research project.

When designing any application there are a lot of decisions that must be made about development. In researching application design for GIS, I learned a lot about common internet based GIS architecture. While the scope of this project will focus on developing a demo application that can illustrate the effectiveness of GIS in Real Estate, we will also look at some common internet GIS-Architecture. Any basic approach for deploying an Internet-GIS application or any application for that matter depends heavily on user requirements (Belay, A. 2006). Because of the server/client nature of the internet there are a few different strategies that can be applied.

We will discuss three strategies, server-side, client-side, and hybrid. In a server-side architecture, a web browser will generate a server request and display results on the client-side browser (Belay, A., 2006). The GIS server will likely combine a standard HTTP web server as well as GIS application server and GIS databases/functionalities on the server-side (Belay, A., 2006). Using this strategy when the GIS application receives a request it performs all of the necessary actions and then returns an HTML wrapped result that can be displayed on the client's browser (Belay, A., 2006). This results in a less client heavy interaction with the GIS but generally also tends to have poor performance and limited user interface/interaction (Belay, A., 2006). **Figure 2** details how the server-side interaction works.

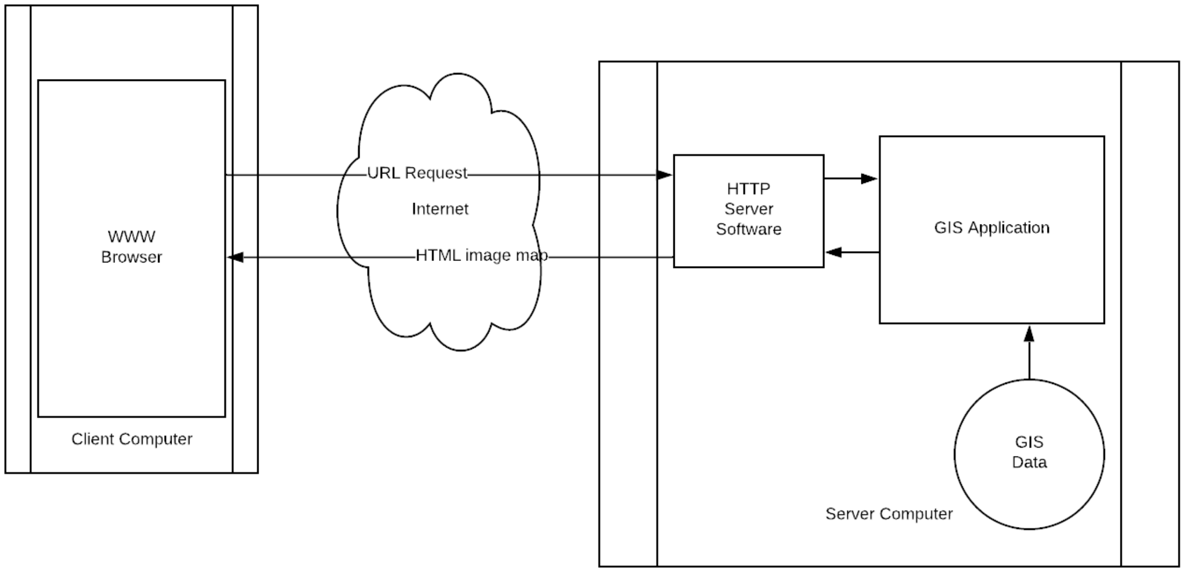


Figure 2. Server-Side Internet-GIS application Derived from: (Belay, A., 2006)

Next, we will discuss a Client-side strategy that moves more of the actual GIS operations to the local machine in a required client application install (Belay, A., 2006). This enhances user capability to perform analysis locally and also allows more controlled operations such as menu interaction that may not be available on a server-side application (Belay, A., 2006). Some of the more robust benefits are the use of core GIS operations such as editing, buffering, overlay analysis and route tracing (Belay, A., 2006). These can be accomplished without re-transmitting a request to the server. Also, in a client-side application the client and server have a division of labor in completing tasks that can speed up data transfer between server and clients (Belay, A., 2006). Weighing advantages and disadvantages in this case can be difficult. On one hand we can provide a more desirable interface combined with a significantly improved performance (Belay, A., 2006). There is of course an issue with distribution of both software and data. This can be problematic for a number of reasons but the most obvious is of course access. **Figure 3** shows what a client-side architecture might look like.

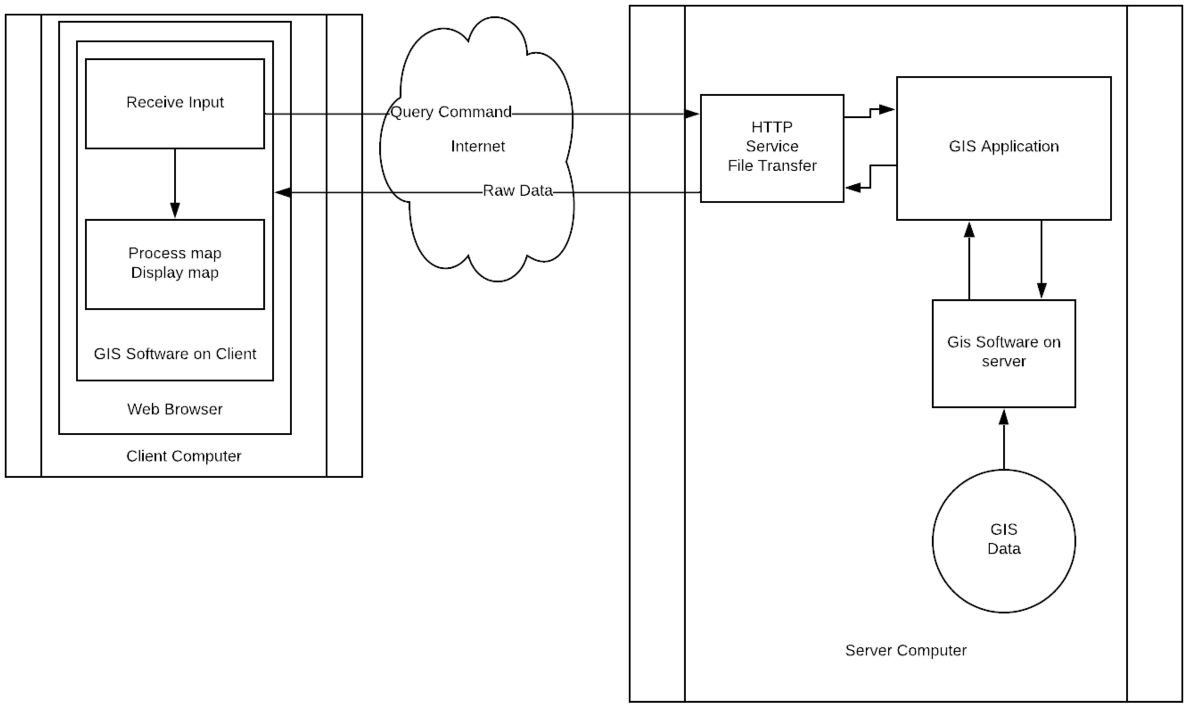


Figure 3. Client-Side Internet GIS Application Derived from:(Belay, A., 2006)

A possible solution to the downfalls of these two strategies mentioned is a hybrid architecture. The use of a relational database to manage a server that a client browser can make URL requests(Belay, A., 2006). The server will then process the requests and generate a map or perform analysis that once complete is transmitted back to the browser (Belay, A., 2006). The server side would need to also consist of Data, Databases, and GIS processing unit (Belay, A., 2006). This solution may work for some situations but may also be undesirable as the end product will always be static and any changes made will need to go through the request process mentioned above. **Figure 4.** Shows a hybrid architecture.

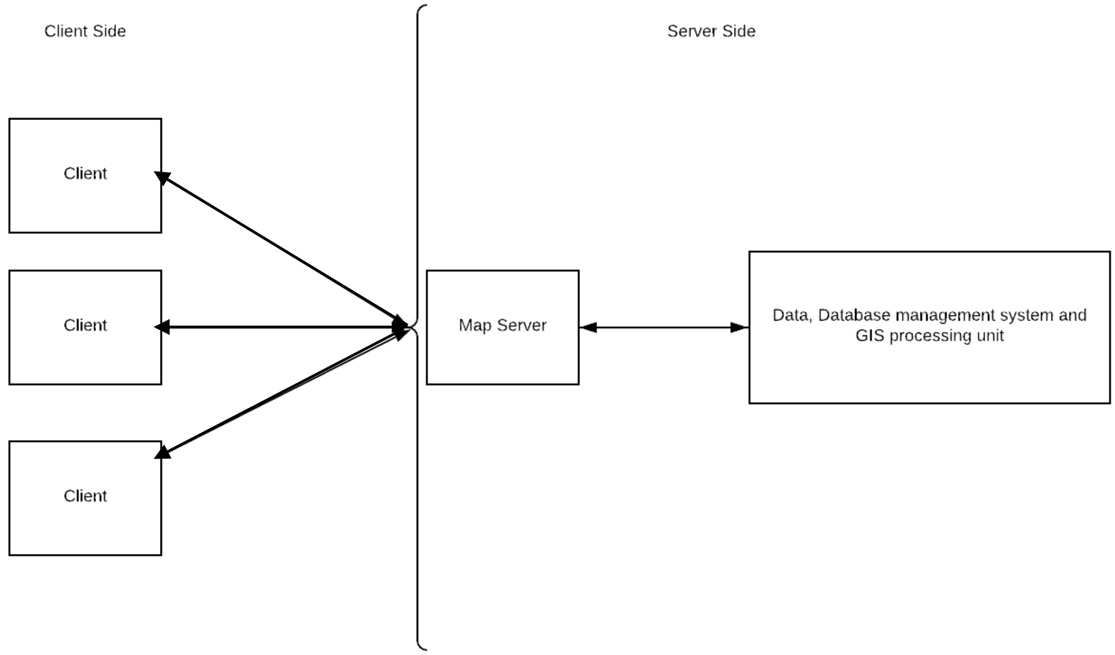


Figure 4. Hybrid Client/Server internet GIS application Derived from: (Belay, A.,2006)

After considering the potential for each of these strategies a gradual tiering of strategies may be necessary. The ESRI APIs chosen for this research uses Jupyter notebooks and codepen/HTML which use a Server-Side strategies. All of the data is hosted on ESRI servers and is pulled from those servers using API calls.

In the future however a Client-Side strategy makes sense to provide more robust functionality to the end user which would also provide tutorials and support. A hybrid solution may also be viable if the application ever makes it to more portable devices with less computing ability. It will be a challenge to create something that will have the right balance of features and portability but starting with the ESRI Python API is a good start.

Next, we will focus on diagramming some requirements, features, and interactions the application may undergo. To do this we will use a few different Unified Modeling Language(UML) diagrams. Each of these diagrams will illustrate the potential stages of development, requirements for those stages, and show how interactions take place. As part of the software engineering process these diagrams exist cyclically. What we will present may not be fully implemented in our finished demo, but it illustrates what we are trying to achieve and gives us guidelines we can use to progress application development. Frequently when designing and implementing software we must cycle back to previous development cycles to make adjustments and tweak behavior. These diagrams serve as the launching point for that process and allow us to visualize before we write our first line of code. For each diagram used we will discuss the basics of how the diagram works and all of the features and relationships they have. We will then discuss the specifics of the diagrams for this project.

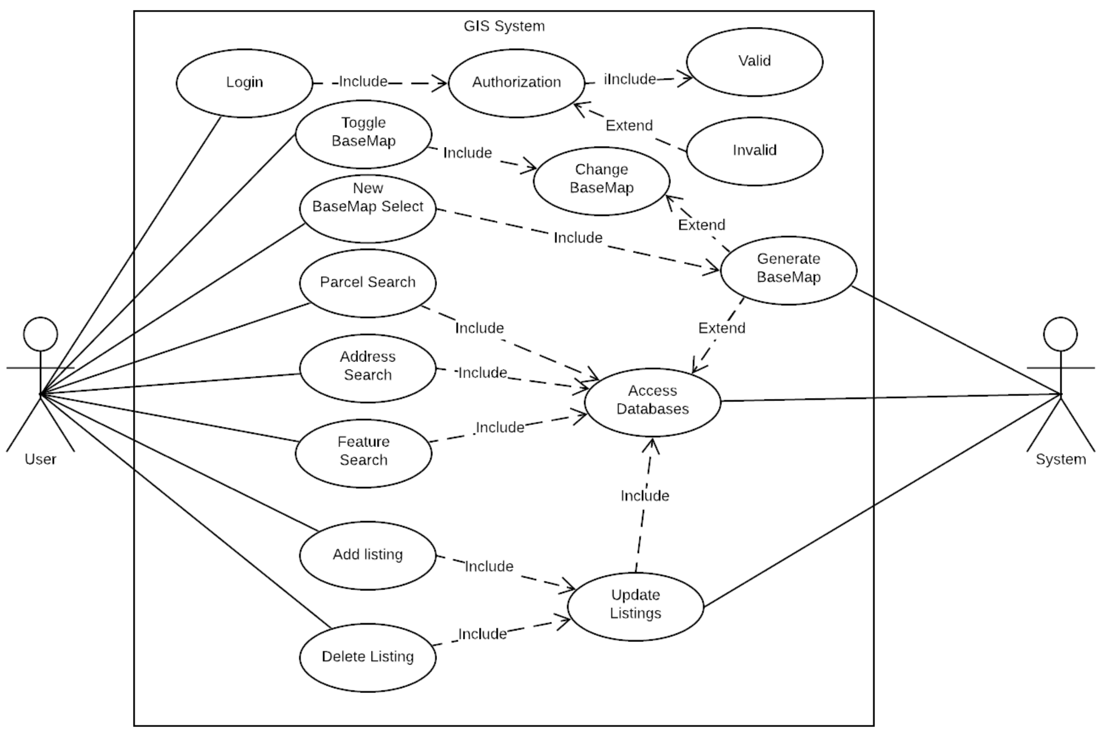
2.2 UML Diagrams:

2.2.1 Use Case Diagram

A use case diagram shows a system or application, the people or other systems that interact with the system, and the basic flow of what occurs in a user interaction with the application. It is a high-level view that does not always have a lot of specific details, but it models interactions well and is quite useful. In a use case diagram, you have some basic elements: systems, actors, use cases, and relationships. Systems are depicted as a rectangle or square in the center of the page with the systems name at the top. Actors are depicted as stick figures outside of the rectangle to represent interaction with the application but not being part of the actual application.

There are two types of actors: Primary actors are drawn on the left side of the rectangle and in most cases are the end users interacting with the system to accomplish tasks. Secondary actors which are depicted on the right-hand side are reactionary actors who respond to actions taken in the application. These are usually external organizations or systems that interact with the application or system in the center but are not part of it. Either way actors are external elements that should remain categorical in scope rather than specific individuals or organizations. Use cases are actions that accomplish a task and describe what the system does. Use cases are represented with an oval and have the name of the use case in the center.

Relationships describe how use cases and actors interact with the system. There are a few types of relationships, but we will only use direct, include, and extend relationships in our model. Solid lines between actors and use cases illustrate direct relationships and at least one direct relationship is required from each actor and some use case in the system. Each include and extend relationship requires a base case and are secondary use cases that occur only when that base case occurs. An include relationship is drawn as a dotted line with a directed arrow pointing from the base case to the secondary case and is labeled include. Include relationships denote a secondary use case that always occurs when the base case occurs. The extend relationship is another dotted line with a directed arrow pointing from the secondary use case to the base use case and is labeled extend. An extend use case occurs as an extension of the base cases and only happens sometimes but is still reliant on the occurrence of the base case. **UML Diagram 1.** is shown below. (Bruegge, 2010)



UML Diagram 1. Use case diagram for GIS application.

In the use case diagram, we illustrate the essential cases in the system. The user interacts with the GIS System to log in and perform searches and listing actions. Each login must be authorized. A typical valid include relationship and invalid extend relationship are present for each authorization. The 'Toggle BaseMap' use case includes the 'Change BaseMap' use case, this use case switches between predefined BaseMaps. The 'Generate BaseMap' use case has an extends relationship in the event that it must change the BaseMap being toggled with a new one from the database. The 'Generate BaseMap' also has an extend relationship with the 'Access Database' use case in the event that the BaseMap needing to be changed is not loaded into the GIS and needs to be retrieved. The 'New BaseMap Select' use case generates the new BaseMap selected by the user to be one of the toggled BaseMaps. Each search use case has an include relationship with the 'Access Database' use case because each search will require accessing the respective database holding the requested information. The Add and Delete Listing have an include relationship with the 'Update Listings' use case which has an include relationship with the 'Access Databases' use case. Each addition or deletion must be updated to the databases, so the information stays consistent. In this diagram the secondary actor acts as the server side of the architecture and illustrates actions that cannot be completed by the client and must be sent as a request fulfilled by the server.

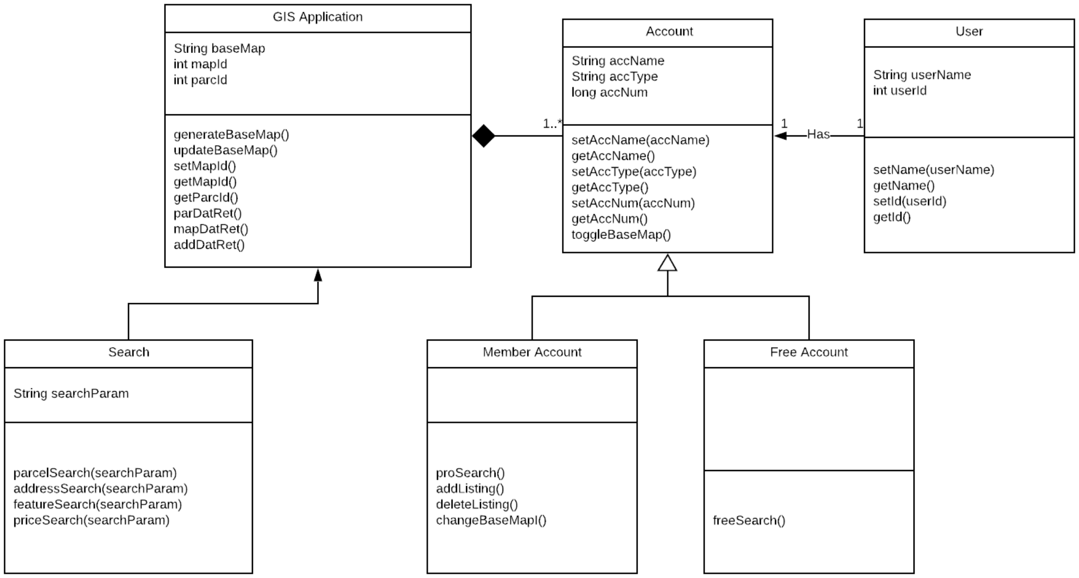
2.2.2 Class Diagram

Class diagrams show a basic structure of what a program designed for a particular system may look like. The diagram is again general and while it can be tailored to a specific programming language's syntax, there are still general principles that can be illustrated that apply to all or most programming languages. This process is also cyclical and does not need to be constrained by the potential of any specific programming language. If the implementation phase realizes a failure in this portion of the modeling process, we return to this stage and adapt for the situation. Sometimes a class is designed that ends up being cut entirely as the process evolves and different methods are discovered. For these reasons we will keep the class diagram for this GIS application as basic as possible. It will illustrate obvious needs of the application while allowing us to also look at how the class diagram works even if we do not end up with this in our demo product.

Each class or object is represented by a box separated into rectangles in the class diagram. The name of the class is in the upper most rectangle followed by the names of the attributes the class possesses. Each attribute is also accompanied by a data type that will support the attribute. These attributes will eventually be the variables maintained by the class that define what it knows. Below the attributes is the methods portion of the class box. In the methods portion we have what the class can actually do as well as what parameters can be accepted in the method or function.

Like the use case diagram, the class diagram also has relationships that represent the interactions and instantiation behavior of other classes and sub classes. Relationships are show by various types of lines and arrows or diamonds. They also have associations that can be defined and quantified. The most basic relationship shows that the class is a sub class that can provide input or return results to the class it is pointing to. Directions in simple associations are not always illustrated, in many cases it is simply a line. Directional lines in these simple cases also do not necessarily indicate a one-way path and can represent communication between the two classes. In this example the search class provides user input to the GIS application class that actually does all of the searching. Next, we have an inheritance relationship denoted by an open arrow. This relationship does have a directional dependency and indicates that the sub classes pointing to a parent class are inheriting attributes from that class. Inherited attributes simply mean that each instance of a sub class has all of the same attributes of the parent class. They can also have unique attributes specific to that instance of a class, but this is not a necessary component. The next relationship is a composition relationship denoted with a diamond. An instance of a class with a composition relationship means that the child class cannot exist without an instance of the parent class. In our diagram accounts are a composition relationship with the GIS application because it cannot exist without the application which is an obvious relationship.

Finally, each of these relationships can have multiplicity that represent how many instances of each class can exist within a relationship. These are denoted in different ways. ‘0..1’ is a zero to one relationship essentially meaning something is optional. An ‘n’ multiplicity is a specified number that is predefined dynamically. ‘0..*’ indicates a zero to many relationship. ‘1..*’ indicates a one to many relationship that means at least one is required but more than one is also possible. Finally, we have a notation for a specific range of numbers ‘m..n’. (Bruegge, 2010)

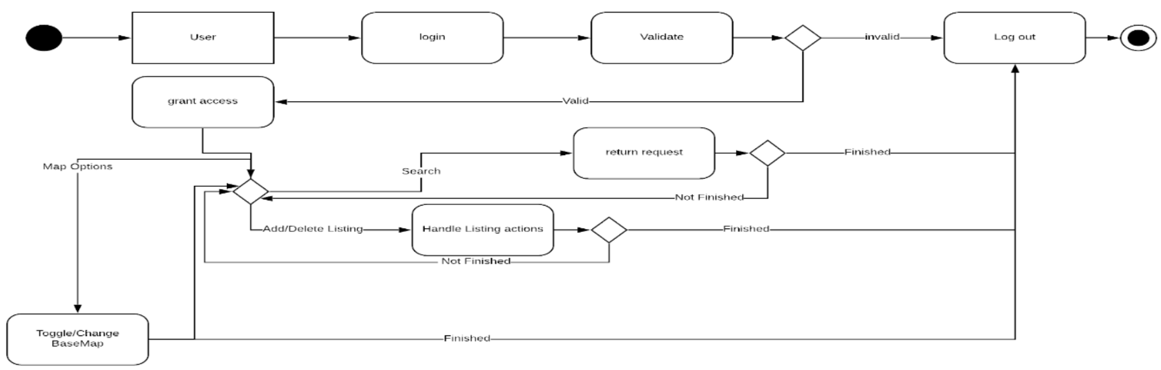


UML Diagram 2. Class Diagram

In the class Diagram the GIS application handles all of the actions taken within the actual application. This involves all Accounts which have name and type attributes. There are two types of accounts a user can create. Each User has identifiable information attributes and methods to set and get these attributes. There is a 1 to 1 relationship for Users and accounts and exists because each User has either a Free or Member account not both. Both accounts can perform searches however the information returned for searches is dependent on the account type. Member accounts receive more data than free accounts. Free accounts and Member accounts can both toggle the base map however only the member account can change base maps. The member account can add and delete listings. The Search class handles all types of searches and passes the appropriate types of search parameters to the appropriate database handling method in the GIS Application Class. The GIS application handles all map generation and database accessing.

2.2.3 Activity Diagram

Activity diagrams simply show a flow of interactions within a single session using the system from start to finish. The black circle denotes our start state which is initiated by a user in the rectangular box. The following rounded edge rectangles represent activities and are known as activity symbols. These are the activities that make up the modeled process and include short descriptions making them the main building block of an activity diagram. The arrows in this diagram show the control flow of the activity and are directional. The diamond symbol represents a decision node that has at least two paths the user can choose from. These paths have condition text that represent the variable options the user can take. These are simple but useful diagrams. (Bruegge, 2010)



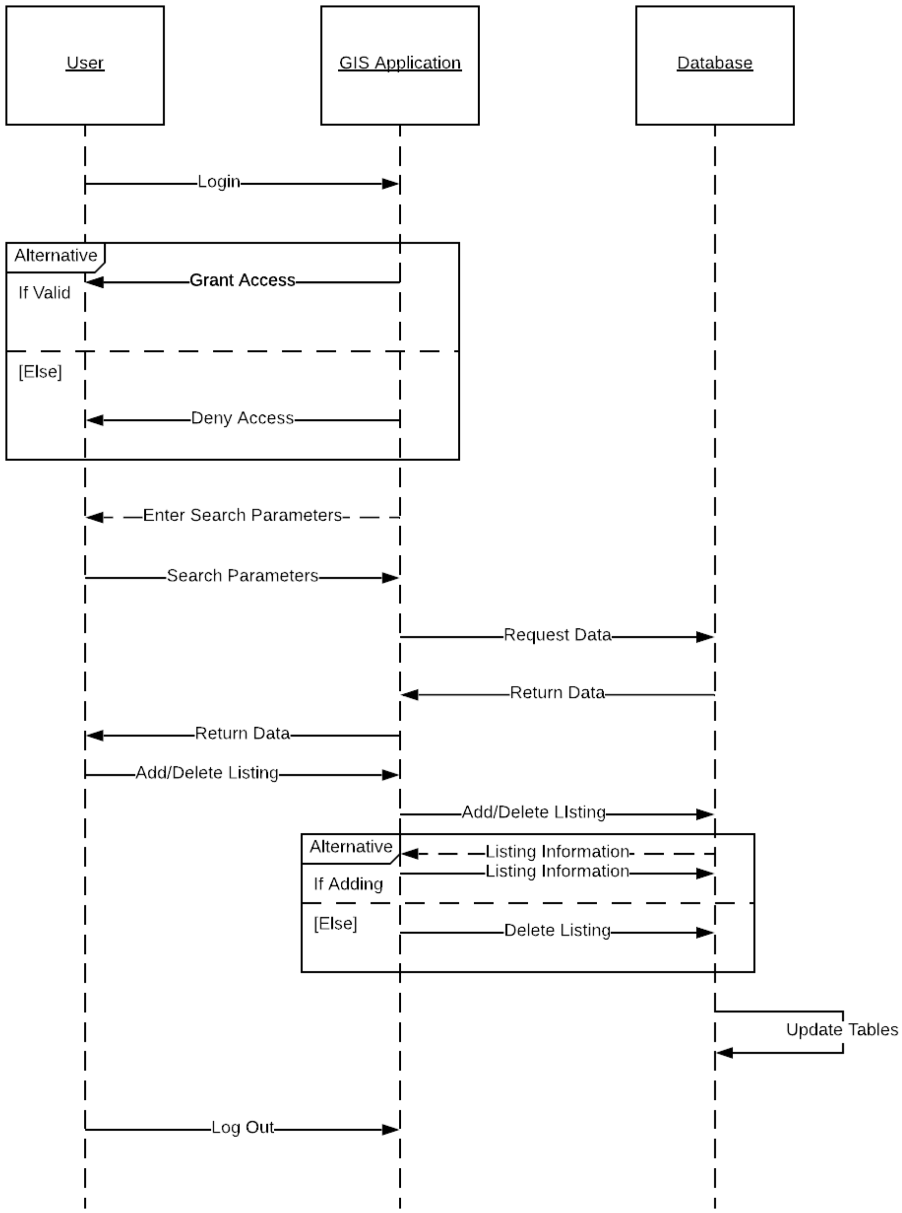
UML Diagram 3. Activity Diagram

In the activity diagram the system initiates when a user logs in. validation is performed and assuming success grants access. If denied it logs the user out of the system. When access is granted the user has the choice to perform a search, do listing related actions, or toggle/change the base map. Each action has decision nodes that either loop back to the previous decision node or allows the user to log out of the system where it reaches the termination state.

2.2.4 Sequence Diagram

Sequence diagrams show the chronological process and order of an interaction with a system. Like other UML diagrams there are objects that must be represented. In this case our actor object is represented using a box, but a stick figure can also be used. Each object also has a lifeline represented by a dotted vertical line. Flowing from the object down, this line represents the passage of time. Each interaction is started with the actor object in this case the user and is represented with a message line. The message line shows what action is being taken and the direction it is being sent. Response messages which occur in reaction to a message are represented with a dotted line pointing back at the actor or system object that sent the initial message. Messages are always drawn from lifeline to lifeline the direction being from the starting system lifeline to the destination system or actor.

Logic interactions can be denoted with an alternate box that generally represents if logic. If one interaction is true, then take that action if not then take another action. Each action that is possible in the alternate box is represented through messages. The sequence diagram below illustrates a hypothetical interaction that takes place between a user and a GIS application that must also communicate with a database. While this may not reflect how our end result demo works due to API restrictions or difference in design. It allows us to consider the system interactions in real time which will assist with requirement scoping as we learn and use more of the available APIs. (Bruegge, 2010)



UML Diagram 4. Sequence Diagram

This sequence diagram illustrates an event where a user logs in, performs a search, and/or alters a listing. The sequence begins with log in and an alternative statement handling successful authentication. Once authorized the system immediately offers a search bar that take in search parameters based on search type for example Parcel ID or Address. Once the Application has the search parameters it queries the necessary database to retrieve the queried data and return it to the application which returns it to the user. Next is an add/delete

Listing message sent from the user to the application with the necessary listing information. The Application then queries the database which has an alternative statement for handling adding a listing or deleting a listing. In the event of adding a listing the database sends a return message asking for the listing information which is returned by the application. If it is a delete, then a delete listing message is sent. After either action the database updates tables. Finally, the user Logs out of the system.

With all of these modeling tools combined we have a foundation that allows us to move forward in the development process. It is worth mentioning again that this process is cyclical and when stepping into the implementation phase you may encounter any number of issues that brings you back to the drawing board with these diagrams. At the end of the process things may look nothing like they did when they began but having the hypothetical foundation is an obvious step in tackling the complexity of the software development process.

3. Results:

3.1 Python:

Preliminary research of the ArcGIS API for Python seemed promising however after working with the API it had a lot of proprietary restrictions that limited progress. One of the glaring issues was the necessity of the Jupyter notebooks to develop with the Python API. This comes with ArcGIS Pro and can be started through the command prompt. It opens in the default browser and runs in that window. While they provide the ability to view the results of implementation in the same window it is a small window that cannot be resized. Limited tutorials made learning about the API more difficult than expected and reading the API documentation proved challenging as well. **Figure 5** is an example of Python code in a Jupyter Notebook that renders a base map and provides some minor geocoding and feature layer services.

```
from arcgis.gis import GIS
from arcgis.geocoding import geocode, reverse_geocode
from arcgis.geometry import Point


trailheads_id = '883cedb8c9fe4524b64d47666ed234a7'


gis = GIS()

trailheads_item = gis.content.get(trailheads_id)


geocode_result = geocode(address="Hollywood sign", as_featureset=True)

geocode_fs = geocode(address=None,
                     location=[-118.71511, 34.09042],
                     category="Coffee shop",
                     out_fields="Place_addr, PlaceName",
                     max_locations=25,
                     as_featureset=True)

geocode_df = geocode_fs.sdf
geocode_df.head(2)
```

Figure 5. Python code in Jupyter notebook to initialize map connection

In **Figure** 5 the first few lines are import statements that call the necessary GIS, Geocoding, and Geometry packages. The "trailheads_id" variable is set to a feature ID value that is hosted on a sample server. Unfortunately, the full address for the sample server is not available so it is difficult to find other relevant information that fits the needs of real estate. This limits us to only being able to draw the data provided in the tutorials or created on our own. Creating our own data is difficult and requires more resources than is currently available. In "gis = GIS()" we open a private connection with

ESRI that will later allow us to pull a map from the server. The following geocode variables define the features we are trying to pull from the sample server using the geocode package. The geocode package is used in this case to pull addresses from the server based on attributes such as ID or category. There is one for a single address the "Hollywood sign" and then another statement for a multiple addresses.

```
m = gis.map("Los Angeles, CA", zoomlevel=11)

m.center = [34.09042, -118.71511]          # `[latitude, longitude]`
m.zoom = 11



m.add_layer(trailheads_item)
m.draw(geocode_fs)
m.draw(geocode_result)



m.clear_graphics()



location = {
     'Y': 34.13419,                # `Y` is latitude
     'X': -118.29636,              # `X` is longitude
     'spatialReference': {
         'wkid':4326
     }
}
unknown_pt = Point(location)

address = reverse_geocode(location=unknown_pt)
address

m.draw(address)


m
```

Figure 6. Python code to apply layers, draw geocode results, and display the map

Figure 6 shows the Python code that creates a map object with the map ID "Los Angeles, CA" and the default zoom level of the map. The center and the zoom values can also be set manually using the appropriate methods. Next, we add the trailheads layer and draw the geocoded address defined previously. There is also a method to use reverse geocoding to attain addresses, but it returns the same results. Finally, a simple "m" diplays the map with all of the applied features and layers. **Figure 7** is the final result in the Jupyter notebook.

```
}
unknown_pt = Point(location)

address = reverse_geocode(location=unknown_pt)
address

m.draw(address)


m
```

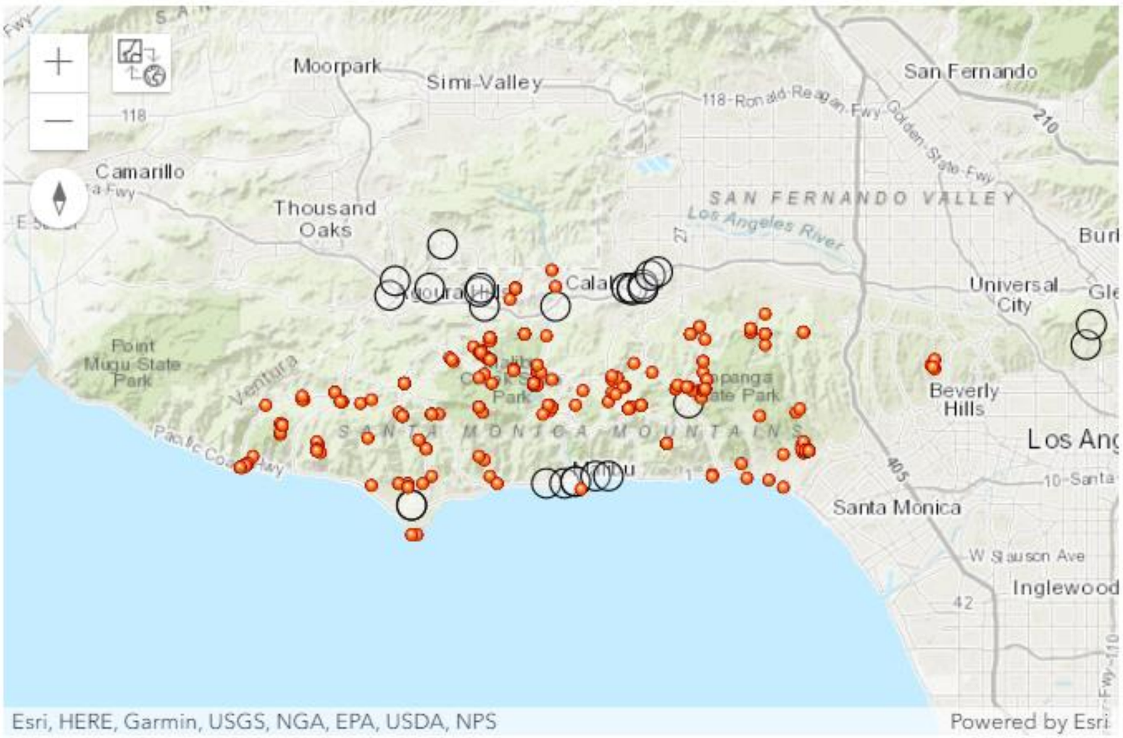


Figure 7. Final projected map in Python API using Jupyter notebook

The view provided is very basic and does not provide the ability to interact with any of the layers. You can view the coded information but to receive new addresses or features you must alter the code to provide that information. This makes the process less fluid and without access to more locations than provided in the tutorial, it limits the ability to demonstrate the API as effective for developing a real estate application. With these discoveries I decided to change directions with the language I am using to attempt to implement the application to JavaScript/Html.

3.2 JavaScript:

The JavaScript API has a lot of similarities to the Python API in the way that it handles items and uses sample servers. There is also a significantly larger volume of tutorials and available sample server data. ESRI recommends the use of Codepen to develop using the JavaScript API because it updates changes live so you can see how your scripts are affecting your application. This is a vast quality of life improvement over the Jupyter notebooks which require you to run your code every time you make a change. You can also simply use any text editor and a browser to test your code which improves portability significantly as you do not need an environment like Jupyter notebooks to develop your app. Based on these differences alone the JavaScript API appears to be superior in developing applications. I decided to stick with using NotePad++ as my editor for HTML and then used a browser to view updates and test the code.

The JavaScript API offered by ESRI can be implemented in any properly formatted HTML document. An internet connection is necessary to make calls to the API as they all rely on a live connection to the sample servers and databases that are hosting the maps, features, data, etc. required to build and utilize the GIS.

```
<!--this is the CSS information followed by critical link informaton necessary to make calls to the ESRI API-->
<link rel="stylesheet" href="https://js.arcgis.com/4.11/esri/themes/light/main.css">
<script src="https://js.arcgis.com/4.11/"></script>
```

Figure 8. ESRI API script call

This GIS is highly reliant on sample servers and databases provided by the free ArcGIS API. Many hours of this project were spent finding and digging through each of these sample servers to locate the features and data that I wanted to use. Part of this involved reading through the different feature attributes to find out what information is present in each of the different callable features. A lot of testing was done trying to make or find different features that would work, some more successful than others but almost all provided more insight on how the API works and different methods to use to achieve different goals.

The initial steps involve deciding on what type of base map you think will provide the data relevant to your task. In this case topographical maps make sense because they provide level of elevation detail that are frequently useful and necessary. To accomplish this a call must be made to the API with a map variable that establishes the map you want as the initial base map.

```
//This variable establishes the basemap for the GIS
    var map = new Map({
      basemap:"topo-vector"
    });
```

Figure 9. Creating map object with a specified base map

These maps are provided by the servers hosted by ESRI the “topo-vector” map is a topographical map made from vector tile layers. The next steps involved choosing relevant data acquisition tools and compiling relevant data using the ESRI GIS API. More work is being done on finding more data that can be used outside of the data in the free ESRI database and from a wider range of locations. Testing the application occurs with each new addition to the scripts and features. Some widgets and tools present in the map interface have similar formatting and native placement which if convenient is maintained for simplicity. For example, the base map gallery object is used to initialize the gallery, but it is drawn to the map with an expand object that actually places the gallery access in the application. The base map gallery itself uses the standard formatting.

```
var basemapGallery = new BasemapGallery({
  view: view,
    source: {
    portal: {
    url:"http://www.arcgis.com",
    useVectorBaseMaps: true,
    }
  }
});
```

Figure 10. Object that establishes the base map gallery

```
/*A view is created which has a library of utilities used to interact with maps.
    This particular view is used to establish a 2D map to interact with and build layers on*/
    var view = new MapView({
      //the popup settings here force all popups to be docked and provides some formating information
      popup: {
            dockEnabled: true,
            dockOptions: {
              // Disables the dock button from the popup
              buttonEnabled: false,
              // Ignore the default sizes that trigger responsive docking
              breakpoint: false
            }
          },
      container: "viewDiv",
      map: map,
      center: [-107.88897, 37.261180], // longitude, latitude of the map when opening
      zoom: 11
    });
```

Figure 11. Defining the view variable as a MapView

The view variable is an instance of a MapView which takes in relevant formatting and positioning information for the map. It also has the popup setting and format as well as the container for the HTML formatting and the map that was defined before. The MapView is then referred to when adding other widgets and object later.

```
var popupParcel = {
    "title": "{NAME}",
    "content": [{
      "type": "fields",
      "fieldInfos": [
        {
          "fieldName": "SITE_ADDR",
          "label": "Site Address",
          "isEditable": true,
          "tooltip": "",
          "visible": true,
          "format": null,
          "stringFieldOption": "text-box"
        },
```

Figure 12. Defining the selection popup menu

The popupParcel variable provide formatting information for the popup that displays when selecting a parcel. It also defines what data is pulled from the database and filled into the different text fields.

The biggest challenge is finding which methods and functions in the API will provide features the way you want them to. A lot of the available functions perform similar tasks but based on different types of data. Some may show trail data that is simply a feature layer being drawn to the map while others may show trail data based on spatial analysis and use different types of data to achieve the same or similar goals but with the appropriate data. This makes understanding these data types important to ensure you are not over or under using resources. A good example of this is querying the database for the parcel feature layer as shown below:

```
var parcelLayer = new FeatureLayer({
  url: "https://services3.arcgis.com/GVgbJbqm8hXASVYi/arcgis/rest/services/Parcels/FeatureServer/0",
  id: "parcels",
  outFields: ["SITE_ADDR","ACRES", "Shape__Area", "ADDR_1","SALEDT","GIS_ACRES","pdParcelId", "SALEP", "TAX_DIST", "YR_BLT"],
  popupTemplate: popupParcel,
  visible: false

});
```

Figure 13. Creating the Parcel Layer object and querying the database of the sample server

In this example we use the URL:
https://services3.arcgis.com/GVgbJbqm8hXASVYi/arcgis/rest/services/Parcels/FeatureServer/0

This calls the sample server that hosts the parcel layer and database with the respective fields. The server which is reachable by going to the URL and looking through the provided feature options. In this case the provided feature is the parcel layer which has meta data containing things such as the database schema to query data from. The "outFields:" field of the feature layer constructor is where the names of the database objects desired are placed to query data relevant to the feature layer. When deployed each parcel object will query the requested data when selected by the user.

At this point it makes sense to talk a bit about generating proprietary data especially if possible. In the case of real estate having better data could provide a market edge. The ability to use the ArcGIS API to generate layers of your own exists however it may come with some costs and is challenging to understand. Still it would be beneficial to generate and maintain your own data in many cases. The profit margins of a business may be able to compensate for the potential costs and challenges of this endeavor.

3.3 Application in use:

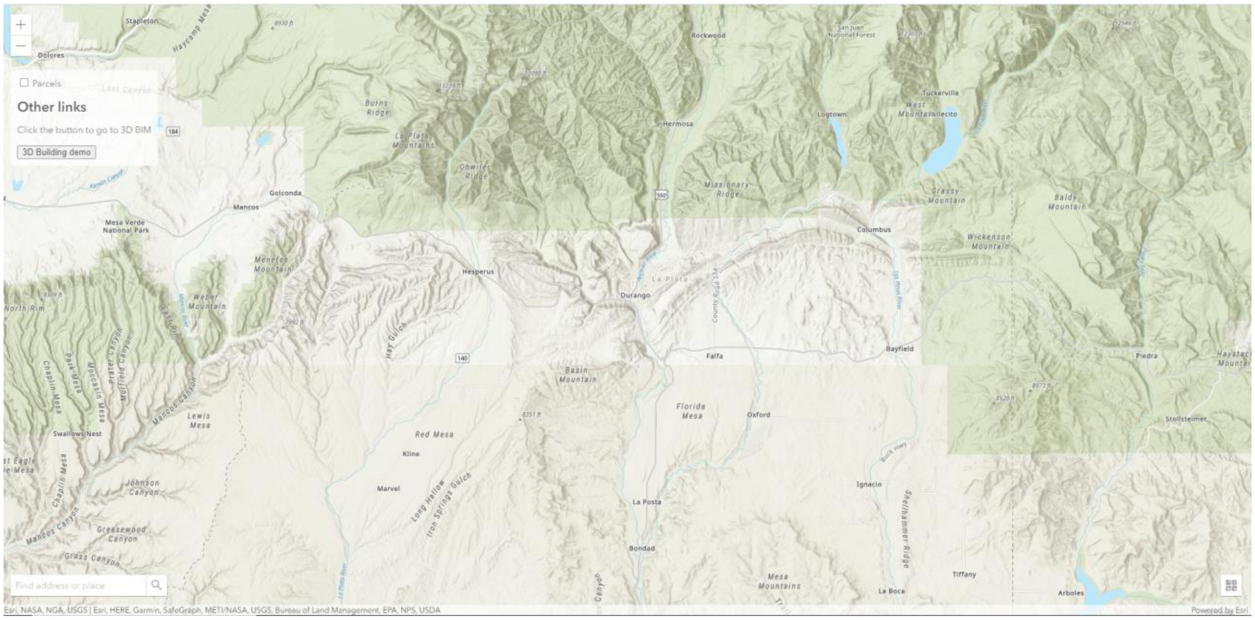


Figure 14. The application upon opening

Here we see the application when it is opened. Its location is set to the city of Durango in Colorado and in the upper left corner we can see the parcel checkbox used to turn on and off the layer visibility. Just below the "parcels" checkbox is a button that will bring up the 3D building demo. This demo was built by ESRI and is just present to show the potential for 3D visualization. In the bottom right we can see the button that expands and detracts the base map gallery for selecting different types of base maps.

Figure 15. Different countries in different base maps

Here we can see the United States, United Kingdom, and Spain each in a topographic base map as well as either a satellite imagery map, a charted territory map, and a dark mode navigation map. Each show different types of data relevant to the area. The ability for these features to work internationally further increases the viability of a real-estate application.

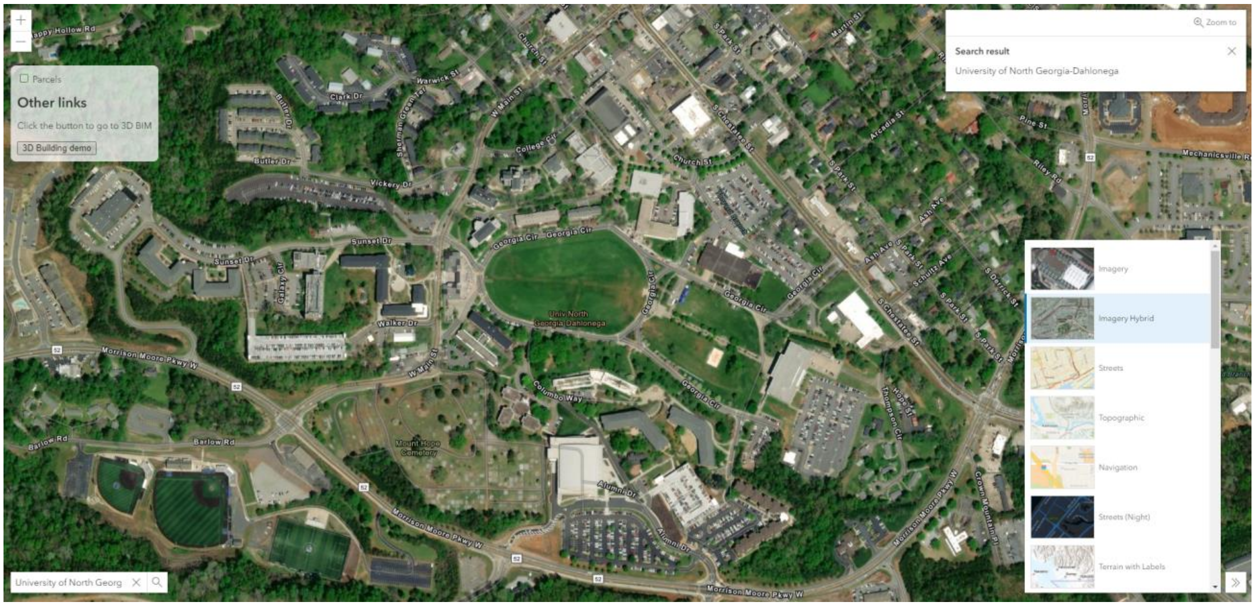


Figure 16. The Dahlonega UNG campus located using search feature

Figure 17. The Gainesville UNG campus located using search feature

Here we can see that the search bar works by entering the addresses for my home campus Gainesville and the Dahlonega campus. A popup with the result is presented in the top right corner and the selected base map is shown in the gallery to the right.

Figure 18. The application with an imagery hybrid base map

This map shows the Durango, CO location but now with an imagery hybrid map that shows satellite imagery of Durango as well as topographic and infrastructure features present in the previous. It also shows that the parcel layer button is currently unchecked, so the parcel layer is invisible. These types of views also make a lot of sense for real estate.

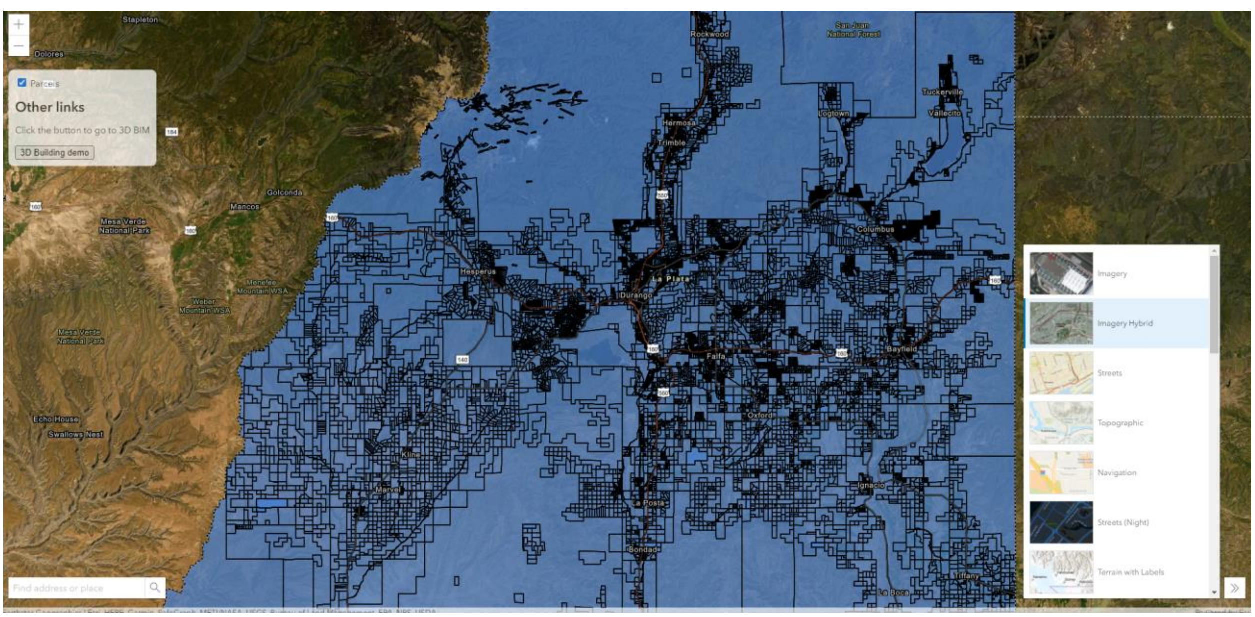


Figure 19. Durango with the parcels

Here we see Durango with the parcel layer for the county drawn on top of the base map. You can see the checked parcel button in the corner as well as the selectable individual parcels present in the layer. Next, we will see some examples of selecting a parcel and the relevant queried data.

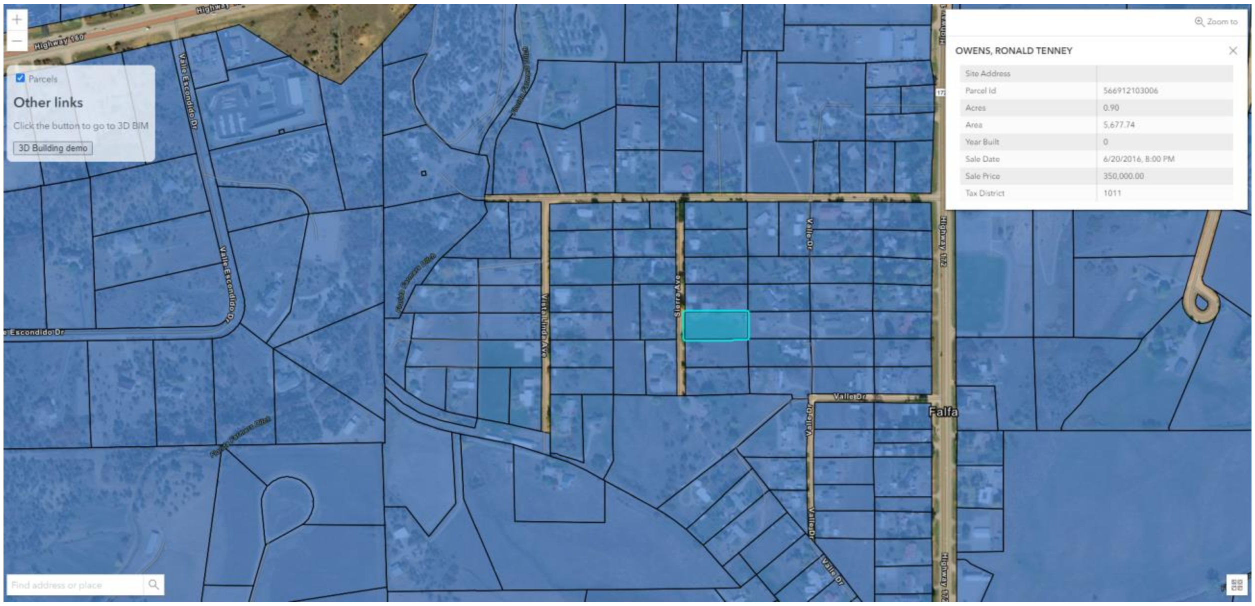


Figure 20. A close up of a city block in Durango with a parcel selected

In this image we can see the parcel selected and some relevant data presented in a table docked in the corner of the screen. Currently we have to work with information provided by the ESRI sample servers. In the future more real estate related information could be provided as well as trends and graphs comparing other selections in the region. The ability to select multiple properties and compare them would be a good goal for future work on the project. As is we can see how this could provide useful information at a glance for purchasing/investment decisions.

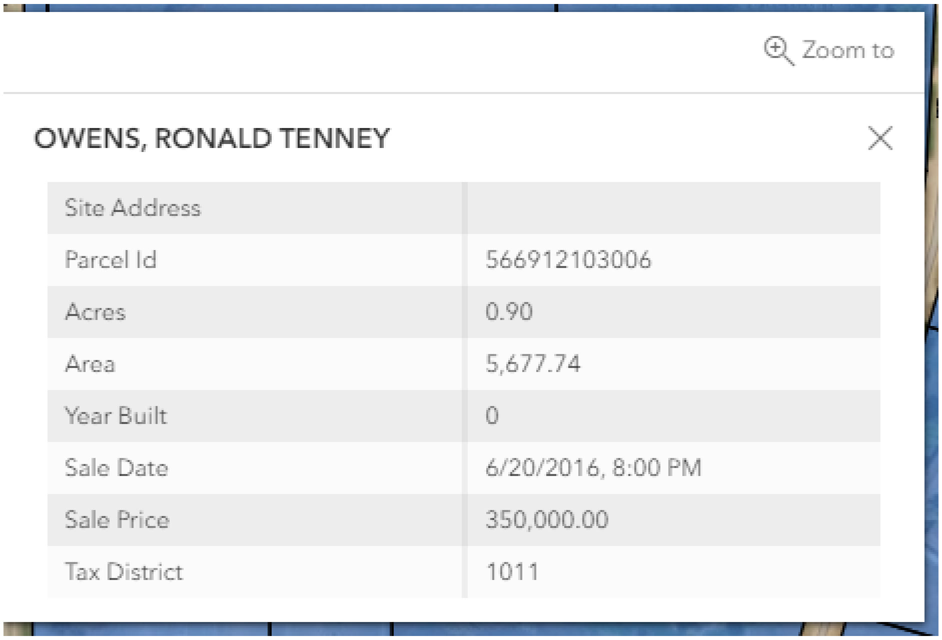


Figure 21. Popup with data corresponding to selected parcel

In this image we can see the data being queried presented in the defined table format in popuptemplate. We can see the current owners of the parcel at the time of the data collection. We can also see certain relevant data involving the parcel such as size in acreage, parcel id, the area of the parcel, year it was established, tax district, sale prices and when the last time the parcel was sold. This could be generated as proprietary data and in this project eventually will be. This could potentially display listing information, community details, average cost of living etc. all of which valuable when determining the viability of the current real estate market. The potential for an application like this to simplify the buying and selling process of real estate is vast.

Figure 22. Slice of the 3D building demo

When the button for the 3D building demo is pressed it takes you to another page with a 3D BIM demo developed by ESRI. This is used to illustrate how in the future a selected parcel could also include a 3D building model with accurate BIM to assist in all aspects of making a valuation decision on the property. This particular demo includes a slicing widget that allows you to view objects in the building such as doors, windows, and furniture. With enough work and resources, a component could be added that allows customers or investors to customize their buildings before purchase. I spent a large portion of time trying to embed this demo link in to parcel selections but was unable to get it to work. Still even with the simple button link it demonstrates the potential of more work and research.

4. **Conclusion:**

After endless hours of research were conducted and discussed. Infrastructure decisions were presented and decided upon and models produced to guide development. We set out to determine the viability of using GIS for a robust real estate application. To research use and design the foundation of an application and implement a Demo to illustrate the potential for such an application. With the time invested and work done I see the current application demo as a success.

Currently the application is useful and though we are working with a limited set of data it is already providing a large amount of the information desirable when making real-estate decisions. With enough time and effort, we could develop a polished real estate application that could change the way business is conducted. Generating proprietary data and hosting your own data for the application could also specialize the GIS enough to make it valuable to every aspect of the process.

There is still a lot of research and work to be done. As it stands the interface is limited. Future work may include the ability to customize the interface, but this will require more research and time with HTML, Python/JavaScript, and the ArcGIS API. I would like to spend more time working with the Python API and trying to bridge the gaps that were encountered during this research. Having access to a more powerful and robust programming language when designing this application may provide more opportunities moving forward. After all is considered however, the foundation has been successfully laid and future work can only improve what is currently available. It makes sense to attempt to incorporate more 3-Dimensional components and views. By extension it also makes sense to look into more detailed imagery for ground level observation. From a real-estate perspective it also makes sense to attempt to incorporate customization features for clients. With these things in mind we can begin future iterations.

5. Acknowledgments:

I would like to thank Dr. Huidae Cho for providing guidance and information throughout the process of developing this research and development paper. His instruction in the course provided good experience with using and researching the documentation available from ESRI. I look forward to working with and learning more from Dr. Cho during my time as his student.

6. References:

1. Podor, A., Nyiri, J., 2010. GIS Application in Real Estate Investment. Scientific Journal of Riga Technical University 20, 94-99.
2. Belay, A., 2006. An application to Real Estate and Housing management in the City Government of Addis Adaba. AAU School of Graduate Studies. URL: http://etd.aau.edu.et/handle/123456789/902
3. Donlon, K.H., 2007. Using GIS to Improve the Services of a Real Estate Company. Papers in Resource Analysis, St. Mary's University of Minnesota Central Services Press 10, 1-11.
4. Koller, D., et. al, 1995. Virtual GIS: A Real-Time 3D GIS. IEEE Visualization Conference.
5. Virtanen, J., et. al, 2015. Intelligent Open Data 3D Maps in a Collaborative Virtual World. ISPRS International Journal of GEO-Information 4, 837-857.
6. Scianna, A., Guardia, M. L., 2018. Globe Based 3D GIS Solutions for Virtual Heritage. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4, 171-177.
7. Wang, T., et. al, 2014. Research on construction of Web 3D-GIS based on Skyline. Institute for Geographic Sciences and Natural Resources Research 9221, 1-5.
8. Parrott, R., Stutz, F. P., 1991. Urban GIS Applications. URL: http://lidecc.cs.uns.edu.ar/~nbb/ccm/downloads/Literatura/URBAN%20GIS%20APPLICATIONS.pdf
9. Gu, X. et. al, 2014. Evaluation of farmland quality with the integration of GIS and RS technology at village scale. The Third International Conference on Agro-Geoinformatics. URL: https://ieeexplore.ieee.org/document/6910578
10. Balaguer, F. et. al, 1997. Patterns for GIS Applications Design. In Proceedings of Programming Language Patterns. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.94&rep=rep1&type=pdf
11. Fosu, R., et. al, 2015. Integration of Building Information Modeling and Geographic Information Systems. Proceedings of the 32nd CIB conference. 196-204.
12. WHO/HQ/WHE/HIM/MAP, WHO GIS team, 2018. Comparison of Geographic Information Systems Software. WHO Geneva. 1-5.
13. Lee, A., Insung, J., 2019. Visualization of City Model with VWorld. IoT Research Division, Electronics and

telecommunication Research institute of Daejeon, Republic of Korea.
14. Gruen, A., 2008. Reality-based generation of virtual environments for digital earth, International Journal of Digital Earth, 1:1, 88-106.
15. Bin, W., et. al, 2016. Research of Optimal Parameters for Parcel-Based Change Detection. IEEE International Geoscience and Remote Sensing Symposium. 2320-2323
16. Sang, L., et. al, 2011. Simulation of land use spatial pattern of towns and villages based on CA-Markov Model. Mathematical and Computer Modelling 54. 938-943.
17. Sevtsuk, A., Mekonnen, M., 2012. Urban Network Analysis: A New Toolbox for Measuring City Form in ArcGIS. International Journal of Geomatics and Spatial Analysis 22. 287–305.
18. Elgaali, E., 2018. Use of Aerial Photography and GIS in Estimating Urban Lawn Irrigation Requirements. Advances in Science and Engineering Technology International Conferences.
19. Lin, Hui., et. al, 2013. Virtual Geographic Environments (VGEs): A New Generation of Geographic Analysis Tool. Earth-Science Reviews 126. 74-84.
20. ESRI Python API for ArcGIS 2019. URL: https://developers.arcgis.com/python/api-reference/
21. ESRI JavaScript API for ArcGIS 2020. URL: https://developers.arcgis.com/javascript/3/jsapi/
22. Bruegge, Bernd., Dutoit, Allen H., 2010. Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition. Technical University of Munich. 31–70.