

Deployment and Maintenance phases. Agile development methodology was followed throughout the development process.

2.1 Requirements Gathering

The Dimension-Oriented Requirements Engineering (DORE) process was followed to gather requirements for this project. The process involves identifying atomic requirements and classifying them in specific categories like User Interface, Business Logic, Data Validation, etc. About 90% of the requirements were identified before the start of the application development by following this process.

2.2 System Design

The scope of the project was analyzed after gathering the requirements. The system was designed according to this scope. Entity-Relationship Diagrams were developed before creating the backend database. The servers hosting the client application, the database and the middleware were designed with the necessary software installations. The system interaction and security was designed using the most optimal approach. Current web 2.0 concepts developed by Microsoft and ESRI are utilized. This application is entirely cloud based and no physical infrastructure is currently in use to keep it running. The security aspect was fulfilled using firewalls and windows integrated authentication.

2.3 Implementation and Testing

After the design, the system was implemented using Microsoft and ESRI technologies. After implementation, the system was completely Unit tested at the code level. Performance testing and tuning were done using Fiddler 2.0. After the system was deployed to the cloud servers, Integration testing was done to verify the performance, security and scalability of the application.

2.4 Deployment and Maintenance

This project is currently in the deployment and maintenance phase. The entire system is deployed on an Amazon cloud. The system was deployed by moving the entire application from the development console to the appropriate cloud servers that supported different interfaces of the application. The system has a backup database and services that can be used to keep it running during maintenance or for additional development work.

3 Technical Features

System environment: Microsoft IIS 7.0, Microsoft SQL Server 2008 R2, Microsoft .NET Framework 3.5, ArcGIS Silverlight API 2.4, ArcGIS Server 10.0

Development tools: Microsoft Visual Studio 2010, Microsoft SQL server 2008, ArcSDE, ArcCatalog, ArcMap

Languages: C# .Net

3.1 System Architecture

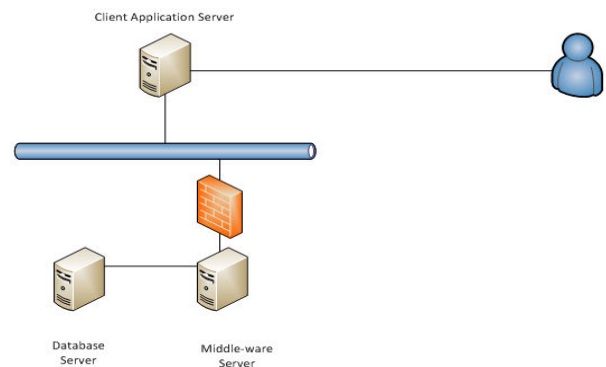


Fig.1 System Architecture

The entire system is built using three distinct servers which serve as a Database Server, Middleware server and a Web Application server. The end user can connect to the system only using the Web application Server and has no connection with the other two servers, thus solving the problem of system security. All the servers are authenticated to connect to each other internally through Windows Integrated Authentication. The Web Application server can reach out to the end user using Anonymous Authentication mechanism. The security for the end users is managed inside the application using a username and password. The

following sections will talk about more about each of the servers in greater detail.

3.1.1 Database Server

Operating System - Windows Server 2008
Database - SQL Server 2008 R2

As the application is a GIS application, where the data is visualized and edited on maps, the data needs to have a spatial component that will identify its geographic relation with the real world. To solve this purpose, a spatial schema is built into the database using software known as ArcSDE (Spatial Database Engine). Using this software, multiple tables can be created in the database and all the spatial information and spatial relationships between the tables can be managed. The tables are created, viewed and modified only by ArcSDE.

All the tables in the database need not be visualized geographically. The tables that have information with no value for its geographical information need not be registered with ArcSDE. For the purpose of this project, the Locations table is the only table that holds spatial information. All spatial tables will have a primary key, 'Object ID' by default, generated by ArcSDE.

3.1.2 Middleware Application Server

Operating System - Windows Server 2008
Web Server – IIS 7.0

The Middleware Application server acts as a medium of communication between the Database server and the Client Application. This server hosts the Web Services that pull and push data between the client and the database. There are two categories of web services that are hosted within the environment – ArcGIS Server based Map Services and custom WCF web services.

ArcGIS Server based Map Services: These services are hosted in a third party web application server known as ArcGIS Server. These services are used to serve spatial data to the client application that can be displayed on a map. These web services are accessed using the REST protocols, GET and POST methods, and can be consumed by multiple users at the same time in a stateless fashion.

Custom WCF web services: These services are built from scratch using the WCF framework of Microsoft and accessed by the client application using the SOAP protocol. These services are used to server non-spatial operations inside the application

such as username and password validations, creating, accepting and denying friend requests etc. The data transfer process between the client and these services is validated using Data contracts inside the web services. These Data contract would essentially define the signature of the proxy classes that will be used by the client to communicate with the WCF web services.

3.1.3 Client Application Server

Operating System - Windows Server 2008
Web Server – IIS 7.0

The client application Server consists of the Silverlight Web Application that will be used by the client. It is the only medium of communication between the end user and the backend system. The application is built using an MVVM design pattern, which is the highly recommended design model for Silverlight based web applications.

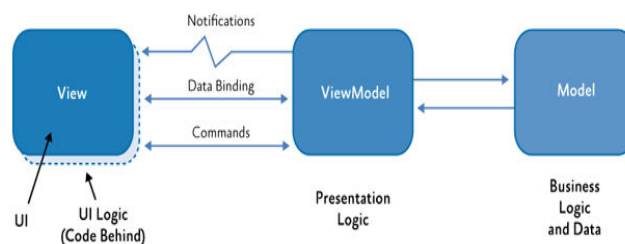


Fig.2 MVVM Architecture [1]

MVVM stands for Model-View-View Model [1]. The Model defines the raw data that is used by the application. This raw data can be conceptualized as a flat table with certain number of rows and columns. The schema of this model lives outside of the application and does not change during the lifetime of the application. The View Model defines the serialized form of the raw data and is compiled along with the application. The View defines the user interface that will display the data to the end client. The View is a very flexible component and can be bound to any view model during the lifecycle of the application.

MVVM within the Application:

In the scope of this application, the Model refers to the Web services that serve data to the client application from the Middleware server. Therefore, the proxy classes that serialize the data from the web services will serve as the Model for this application. The View Model is defined in form of C# classes and also as Data templates inside the XAML that creates the Silverlight pages. The View

is straight XAML markups that define the user interface of the application.

3.2 System Security

Security is very important in network applications because the applications are accessible to a large population of users. When this network is Internet, security becomes even more important because the list of potential users grows even larger. Apart from the web server security, system security of the application also has to be considered. [2]

This system authenticates the users when they type credentials (typically a user name and password) into a Web form. Each Web application will have a Web.config, in which the login page URL will be defined. Any user that tries to access the site for the first time will be redirected to the login page. The login page will call a Local authenticating method to query database and check the validity of the username and password. If it returns true, the user will be allowed access to his home page. The user's full name will be displayed on the top of home page. A Logout button will allow the user to log out of the application at any time.

3.3 System Performance

Silverlight applications run all the UI code on Client. Some service call back to Server to pass data back and forth is needed sometimes. Most of the time the applications mainly use client side resources. It reduces the bandwidth usage to Server. But it requires a longer initial loading time when the application is first started and there is higher memory/CPU usage on the client machine.

4 System Description

This system supports the following modules and functionalities: Creating location based stories, sharing information with friends with different levels of security, creating a friends list, searching by address, file attachments to location stories and high resolution basemaps.

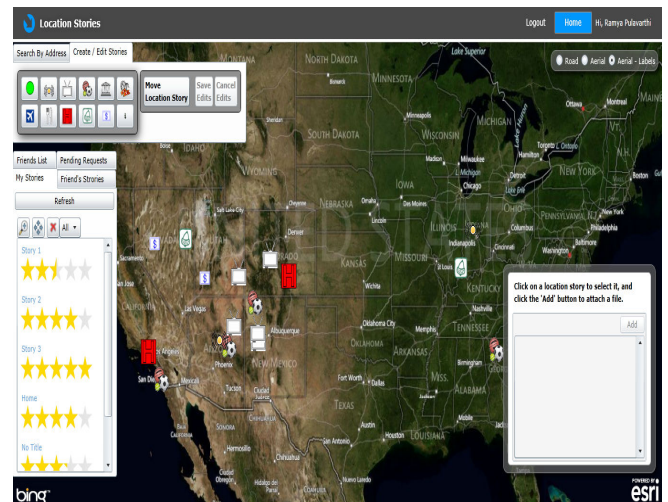


Fig.3 Home page of the user

4.1 Basemaps

Basemaps are used as a reference for better visualization of spatial data. The basemaps used in this application are obtained from Microsoft Bing using a Bing Token that authenticates the client application using the imagery with the token based authentication.

This application presents the user with three different basemaps, that can be chosen from. The three basemaps are; Aerial imagery, Street imagery and Aerial imagery with labels. The system allows the user to switch between basemaps at any scale and extent of the map without affecting the location stories that are displayed on the map.

4.2 Friends List

The user can add friends to his list by searching through the database. The database can be searched by the first name, last name or email address. The application will query the database through the middleware layer using the REST protocol and return all the friends that match the search criteria and will exclude the people who are already in the user's friends list. The user can send a request to any user in the list. Once a friend request is sent, the information about this request is maintained in a table named Pending Requests. When the user with a pending friend request logs into this account, the pending requests will appear as a list and give the user an option to accept or reject the request. If the request is rejected, this record will be removed from

the pending requests table. If the request is accepted, the record is moved to a table called Friendship. This table hosts all the friend pairs. For example, if A and B are friends, there will be two records for this friendship, one where B is a friend of A and stores the friendship level that A has assigned to B, and the other where A is a friend of B and stores the friendship level that B has assigned to A. This type of storage allows for retrieving a friends list for any user in the order of their friendship level.

4.3 Geocoding Addresses

The application contains a tool that helps the user search for an address by taking the street name and the zip code as the input. The Latitude and Longitude associated with that particular address is the output of this tool. Once the Latitude and Longitude are obtained, the tool will enforce the map of the application to zoom to that area and display a region which is a 100 meter radial buffer around the given address.

The process of converting an Address to a Latitude and Longitude is known as Geocoding [3]. Geocoding is done by looking up a table which has the Addresses and associated latitude/longitude stored in it. A geocoding service published by ESRI on the internet is used since it is practically not possible to host a database that contains all the addresses in the world for the scope of this project. This Geocoding Service is a web service that takes the address as the input and returns all the possible Latitude and Longitudes for that address. It also assigns a score to each of those Latitude-Longitude pairs based on the accuracy. It is a common practice to use a minimum score of 80 for geocoding via ESRI's geocoding service. However if more than a single Latitude-longitude pair has a score of more than 80, then the pair with the highest score is chosen as the correct Latitude and Longitude. There are cases when the address does not find any match with a score of more than 80 percent. In that case a message is displayed to the user to correct the address that has been entered.

4.4 Location Stories

Location stories in this application are stories about events that occurred at a certain location. These stories can be created by a user and shared with

other users. A location story has a category, a title, a time of creation and the message or story. The time of creation is decided by the system assigned to the story. All other attributes are entered by the user and can be modified at any stage. A location story can also have attachments that it can hold and shared with all the users who can see this location story. The attachments can be of several data types. For the scope of the project, 46 most common data types are included.

A location story is stored in the form of a table in the geo-database. This is the only table in the geo-database that has a spatial schema and is capable of storing spatial information. The tables in a geo-database that hold spatial information are called as Feature Classes. Each feature class can hold one geometry type. The geometry type defined for the Location Story table is a point. The spatial data that is hosted in the database supported by ArcSDE is created, viewed and modified by ArcGIS Server that is hosted on the Middleware server. The client application sends messages to the ArcGIS Server for creation, visualization and modification of spatial data through web services using a REST protocol.

4.5 Sharing Location Stories

The location stories created by a user can only be seen by the users who are in his friends list. The application also allows the user to configure security within his friends list. Each friend is given a specific number of stars (between 0 and 5) that are proportionate to amount of transparency that will be maintained with the friend. In a similar way, each story that is created by the user will also be given a specific number of stars (between 0 and 5). A friend of a user can only see the stories posted by the user that have lesser number of stars than those assigned to him. For example, A friend who is given 4 stars can see all the stories posted by the user that have less than 4 stars.

4.6 File Attachments:



Fig.4 File attachments view

This is a simple yet powerful tool that allows the user to attach a file that carries specific information about a location story. To attach a file the user must first select the location story and then browse to the file on the hard drive. This attachment can be downloaded and visualized by the user’s friends that have the permissions to see this story. The application allows 46 file types for attachments.

5 User Interface

5.1 User Login

When a user logs into the application for the first time, the user is required to complete a registration process. For registration, the user is required to enter the following details; first name, last name, email id, user id, password.



Fig.5 Registration window

The system validates the registration process by checking that the user id and email address are unique. Once the registration process is complete, the user can go back to the login page and log into the system by using the right user id and password.

In case the user forgets the password, the system shall require the user to enter the user id and click on the ‘Forgot Password’ hyperlink. This will send an email to the email address the user specified while registering with the system.

5.2 Add Friends

The user can search for other users in the system by providing their first name, last name or email address, by clicking on the Add Friends button on the ‘Friends List’ tab. A dialog box shall pop up when the user clicks on the ‘Add Friend’ button. This dialog box provides an interface for the user to search for other users.

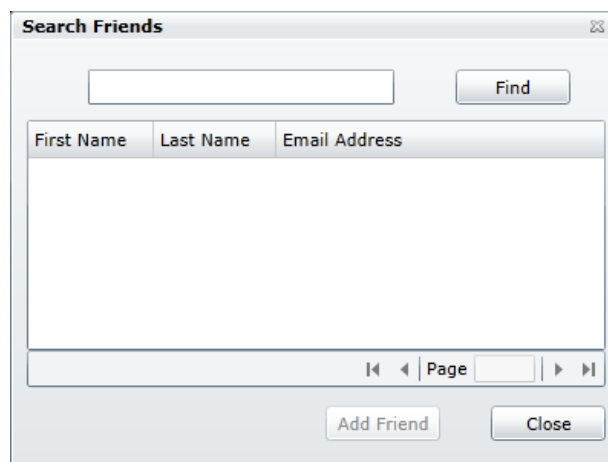


Fig.6 Add friends window

After the user enters the keywords into the textbox and clicks on the 'Find' button, a list of matching users are returned and displayed in the list. The system shall enable the 'Add Friend' button when the user selects a user in the list. This list will exclude the current friends of the user.

5.3 Location Stories

A location story is created by selecting the appropriate template from 'Create/Edit Location Stories' tab. There are 12 categories of location stories in this application. Each template has a tooltip indicating the name of the category it represents.



Fig.7 Location Story template

When the user clicks on the template, the Add Location tool is activated and when the user clicks on the map, a location story is created at that point and a dialog box is displayed for further information, i.e Story title, story and the number of stars for privacy settings.

5.4 Editing Location Stories

5.4.1 Editing Spatial Information

The spatial extent of the location story can be modified by using the 'Move Location Story' tool in the 'Create/Edit Location Stories' tab. This tool can be used to move the location story around the map by clicking on and dragging a story around. The spatial edits made can be saved in the database by clicking on the 'Save' button.

5.4.2 Editing Non-Spatial Information

The non-spatial information related to a location story can be edited by clicking on the hyperlink under the 'Stories' tab. This interface is the same as the Create Location Stories interface with pre-populated information. This information can be edited and saved in the database by clicking on the 'Ok' button in the window.

6 Backend

6.1 Database

The database used in the application is ArcSDE. ArcSDE is software that provides a spatial schema to a regular database [4]. This spatial schema can be used to store spatial and non-spatial data objects. This schema can be understood by other ESRI softwares. The ArcSDE software runs as a service, using which all other ESRI softwares can connect to the database. This service always runs on port 5151 in the current system. It also supports versioning capabilities within the database using which several versions of the database can be maintained simultaneously. These versions can either be timestamped or have a distinct name. The Locations table is the only spatial table that is registered with ArcSDE. All other tables do not have a spatial context and therefore not registered with ArcSDE. Installation of ArcSDE creates specific set of tables that are used to maintain the spatial objects in the database. Below is the schema of the database used in the project.

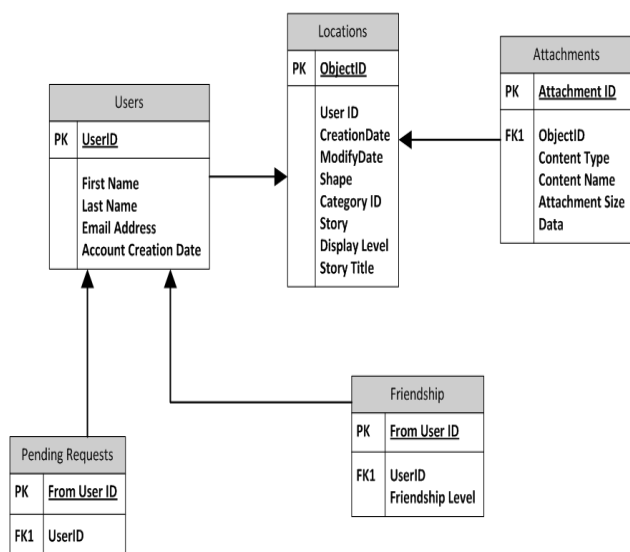


Fig.8 Database Schema

Table Name	Description
Users	Table that holds the information of all the users registered in the application
Locations	Spatial table that contains information about the location stories
Friendship	Table that holds information about friendship between two users
Pending Requests	Table that holds all pending friend requests for users
Attachments	Table that holds all file attachments for location stories

Table 1 Database tables

6.2 MXDs

MXD file format used by ESRI mapping software, such as ArcMap; saved in a compound format that includes the map description, layout, and embedded objects saved in the map; organized into units called storages and streams.

6.3 ArcGIS Server

ArcGIS Server is a Web GIS that helps you take your geographic information and make it available to others. Perhaps you've already had some experience using GIS software to create GIS resources, which are the maps, geodatabases, and other tools that you need for storing and using your geographic information. ArcGIS Server gives you the power to take the GIS resources on your computer and make them available to a wider group of users throughout a network of computers. In ArcGIS Server, the way you publish a GIS resource to others is through a service.

6.4 Map Services

Map Services are web services that are hosted in the Middleware server inside the ArcGIS Server. These services have a REST and SOAP interface. This project only uses the REST interface. These services facilitate the creation, visualization and modification of spatial data. These services can be hooked to the client application in such a way that any updates made to the database will automatically be seen in the client application without having to refresh the middle ware services. The services can also transport the symbology of the spatial data that is defined in the MXD.

All the changes made to the spatial data on the client side are stored in the form of a session in the browser and the map services validate these edits before posting them to the geodatabase. The map services can be turned off during database maintenance activities without having to disturb the client application. This allows for uninterrupted usage of the application even during maintenance times.

7 Conclusions

It is becoming increasingly necessary to build applications that stand out and provide users with superlative and immersive experiences, as a result of the high competition in the web application market. The rise of cloud computing and various mobile platforms has increased the complexities involved in building and supporting web applications.

Traditional techniques of building web applications rely on the web server to do most of the heavy lifting with the browser posting back to the server for every little action. This method is not possible in clouds and is uneconomical. Modern web applications, on the other hand, make the browser do a lot of work. The browsers these days and the machines that run them have extremely fast interpreters and graphics engines. They also have high levels of local storage and offline capabilities. A good modern web app leverages these capabilities to provide a rich user experience and also reduce the load on the server, thereby reducing costs as well.

8 Future Work

The current scope of this project defines a technical platform that can be used to create a social networking application to share information that has a spatial context. The limited scope of this project is with respect to the time and resources that have been available to accomplish it. Below are the functionalities that are to be further accomplished to make the system more user-friendly and useful;

1. A mobile interface for this application that allows the users to create location stories at current locations by utilizing location sensors in today's smart phones.
2. Allow multiple users to provide information for a single location story.
3. Define a large number of categories for the location stories.
4. Use oblique satellite imagery for the base map.
5. Allow the user to use a profile image and store more information about the user.
6. Allow the user to display the current location.
7. Provide an option to display driving directions to a certain location story from any place.

References:

- [1] <http://msdn.microsoft.com/en-us/library/gg405484%28v=pandp.40%29.aspx>
- [2] <http://msdn.microsoft.com/en-us/library/ff648183.aspx>
- [3] http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?topicname=geocoding_a_table_of_addresses
- [4] <http://www.esri.com/software/arcgis/arcscde/index.html>