

CS 235 Project and Presentation

Hola: Connecting People

Team Members:

Paridhi Agarwal
paridhi.agarwal@sjsu.edu

Piyush Bajaj
piyush.bajaj@sjsu.edu

Abstract

A web-based application which enables a user to locate other users in a user-specified location. Intent of the application is to provide users a user friendly and easy to use application to see people in it's vicinity. The application can be expanded to include search based on multiple user's parameters.

1.0 Introduction:

In order to represent the software application and all its intended objectives, the UI design plays a crucial role. The software will not be communicated to the user for its possible benefits if the UI design is not created with the relevant and significant amount of information. Therefore this application follows the path of the aesthetic design process so that the user should be able to understand the system fully for utilization. The login page and the home page of this application are the only faces of the software which is visible to the user which abstracts the user from the complex flow of data and interaction between the various components in the system architecture and hence, if the user is unable to understand and make use of it fully then he will not be able to use the developed software at all.

The highlighted objectives of this application are it enables users to find people in their vicinity. It shows the distance, direction and the ideal time it takes to reach from one user to another end-user via different modes. Learning goals include efficient use of Facebook API's to gather relevant user data for analysis and application. Google Map API's was used to leverage the user's data visualization. Dynamically store user data and maintain a single repository for analysis and rendering visualization. Development of an interactive user-friendly interface is of key importance.

2.0 Functional Design

The design is kept simple in order to reduce the cognitive load for the User. It was kept in mind that how much information should be provided and how many UIs should be generated on a single subject. If the user has to traverse across a wide number of pages, it causes frustration. Hence this application propagates between only 2 web pages to serve its purpose.

The application will maintain a single data repository which will hold all user data fetched by Facebook APIs. MongoDB collection serves the purpose for this repository.

Login Page:

User is presented with the login page on opening the application. (ref. Fig.1) Here user is presented with a FaceBook login button that is integrated with Facebook's API (ref. Sec:7 [2]). User will be asked permission to grant access to the application to fetch it's Facebook details. An alert window is displayed to grant access to User's location. Upon successful login, user will be redirected to the app's home-page. Data fetched on this page, is stored dynamically onto the user-database.

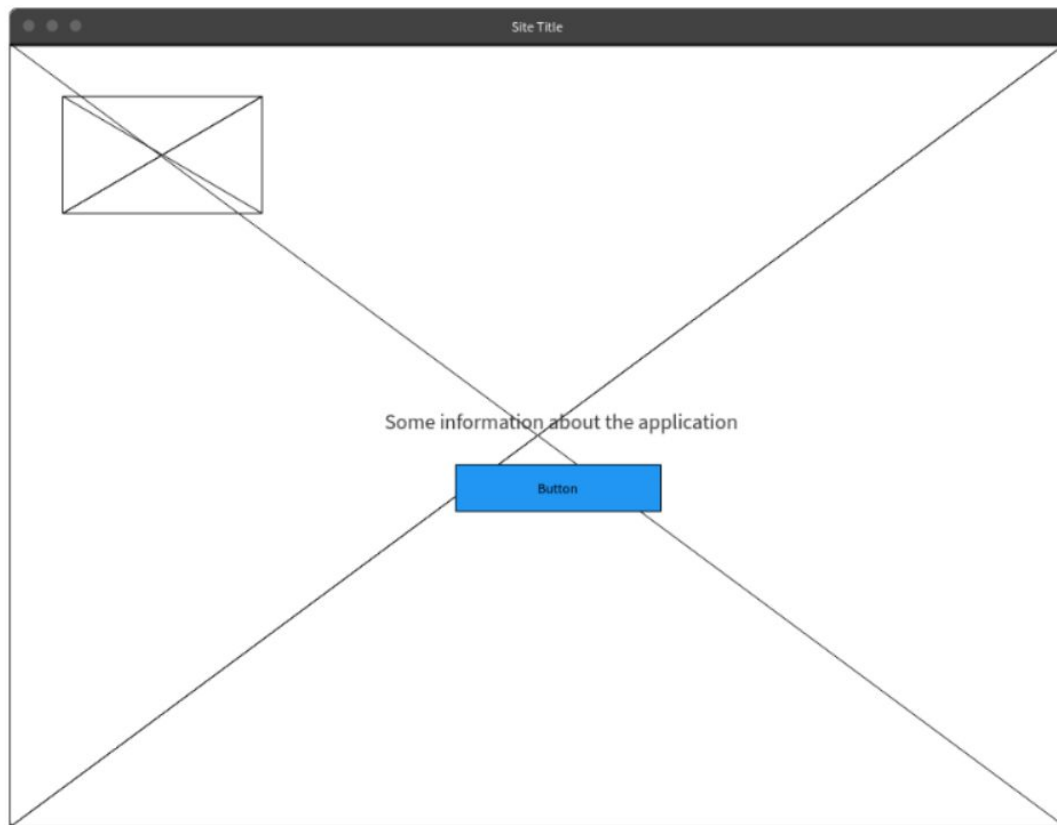


Fig 1. Wireframe for Login Page

Home page:

Application's home page is based on model view controller paradigm, where a user is presented with the view of the actual model and can manipulate it using manipulators such as buttons and dropdowns lists.

It is divided into two sections. (Ref. Fig.2 Wireframe for Home Page)

Section1:

This is the interactive section of the application. User is presented with a placeholder to specify a location in order to view other users in that locality.

Once a user is viewing a given locality, user can use two dropdown lists in order to get the distance and direction of other users in the area.

Dynamic queries design pattern is implemented through labels and dropboxes.

Dropdown1 is used to filter users that are "x" time away from the current user.

Dropdown2 is used to filter the mode in which the user wants to travel to other user.

A logout button helps the user to return to the disconnect from the application and return to login page.

Section 2:

This section holds the map generated via Google Maps API.

User will be able to see map markers of other registered users of the application.

As a pre-attentive measure: color is used to differentiate markers.

All the markers are highlighted in red. On hovering over the marker, color is changed to blue. To show selection marker is drawn yellow in color.

Design patterns such as Heat maps, Data-tips, Dynamic queries.

HeatMaps are implemented to show density of users in an area.

On clicking the marker, user will be presented with a Data-tip box, providing information about the marked user. This data tip includes the user name and a Google-Map Street-view image of the marked user's location.

On selecting the draw button, User can draw a polygon over the map to filter users that are present under the drawn polygon. Other markers that lie outside the drawn polygon will be hidden when in draw mode.

This page is always in sync with the User database, so when new user register, current users will be able to see all markers.

On selecting the logout button, user will be disconnected from the app.

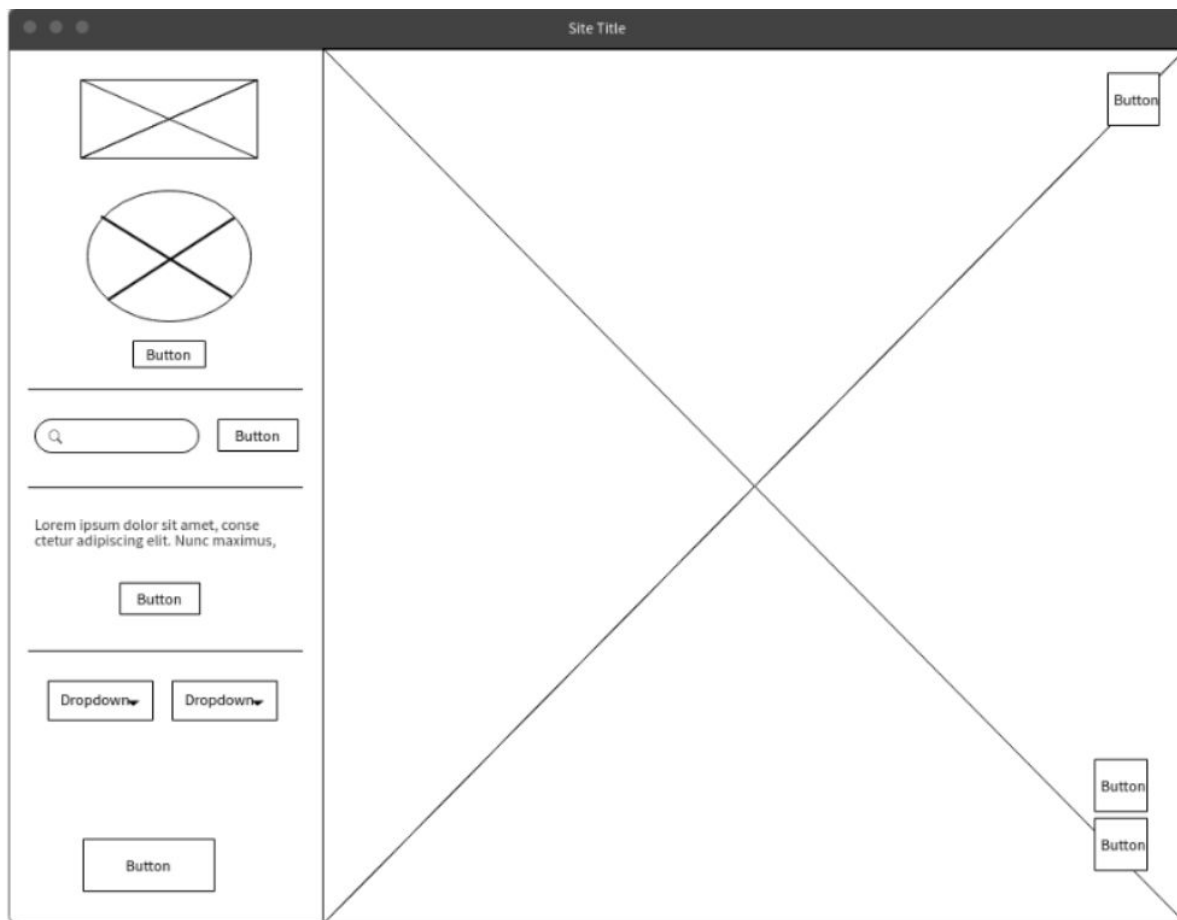


Fig 2. Wireframe for Home Page

Key Considerations:

Real-time user's geo location update:

A typical user is interested in knowing the status of his friends status based on precise current location at all times during the correspondence and communication. Therefore, the current location is being continuously updated in order to provide an updated information and seamless experience to the user.

System Consistency:

Consistency is an extremely important feature for the positive impact on the UI design of this application. Consistency was maintained for this application across all the UIs as well as in terms of platforms. All the UIs developed for the application had similar kind of design, choice of colors and platform. The UI that followed the certain theme and objective of any larger software system that it might be used to integrate into later stage, so that the proper message and intention of the application can be communicated to the user.

Error Handling:

It is important for the UI design to provide the end-user with a convenience that the run-time errors can be handled. In this context of our application, even if the user is unaware of the errors that are possible to occur at the back-end of the system still the UI design was capable of catching as well as providing a solution for these errors.

These errors can be of two categories runtime and compile time errors, but for a UI design, the run-time errors are of utmost significance. The UI design caters all possible forms of inputs that a user can enter and a design was finalized after testing the UI for all possible error types.

Efficiency:

The efficiency of the UI is measured in terms of the amount of related information it brings to the user. If a user visits the Hola webpage, he has certain requirements and expectations in his mind. The application brings all the requirements that he wishes and most of where he wishes.

The placement of information in the light of a user's requirements plays a vital role in making the application efficient. The application web pages are designed based on this concept. UI that has the best sorted information, will provide efficiency to the user in ideal time and with precision.

User Involvement During Design:

It is important to involve the user and get proper feedback at all stages during the design of the UI. Even after the design has been deployed, it is important to know what the users are expecting with the growing information about the varied use of the application across the internet and otherwise. As the users will see the UI evolving according to their expectations and requirements, they will engage themselves more and more into the process and the system will have more responses in future for changes as well as improvements.

3.0 System Design

The project architecture is graphically portrayed in Fig. 3.

Libraries used:

- Facebook API
- Google API
- Node.js
- Express.js
- Mongoose.js

Languages used :

- JavaScript

- HTML
- Database used:
- MongoDB

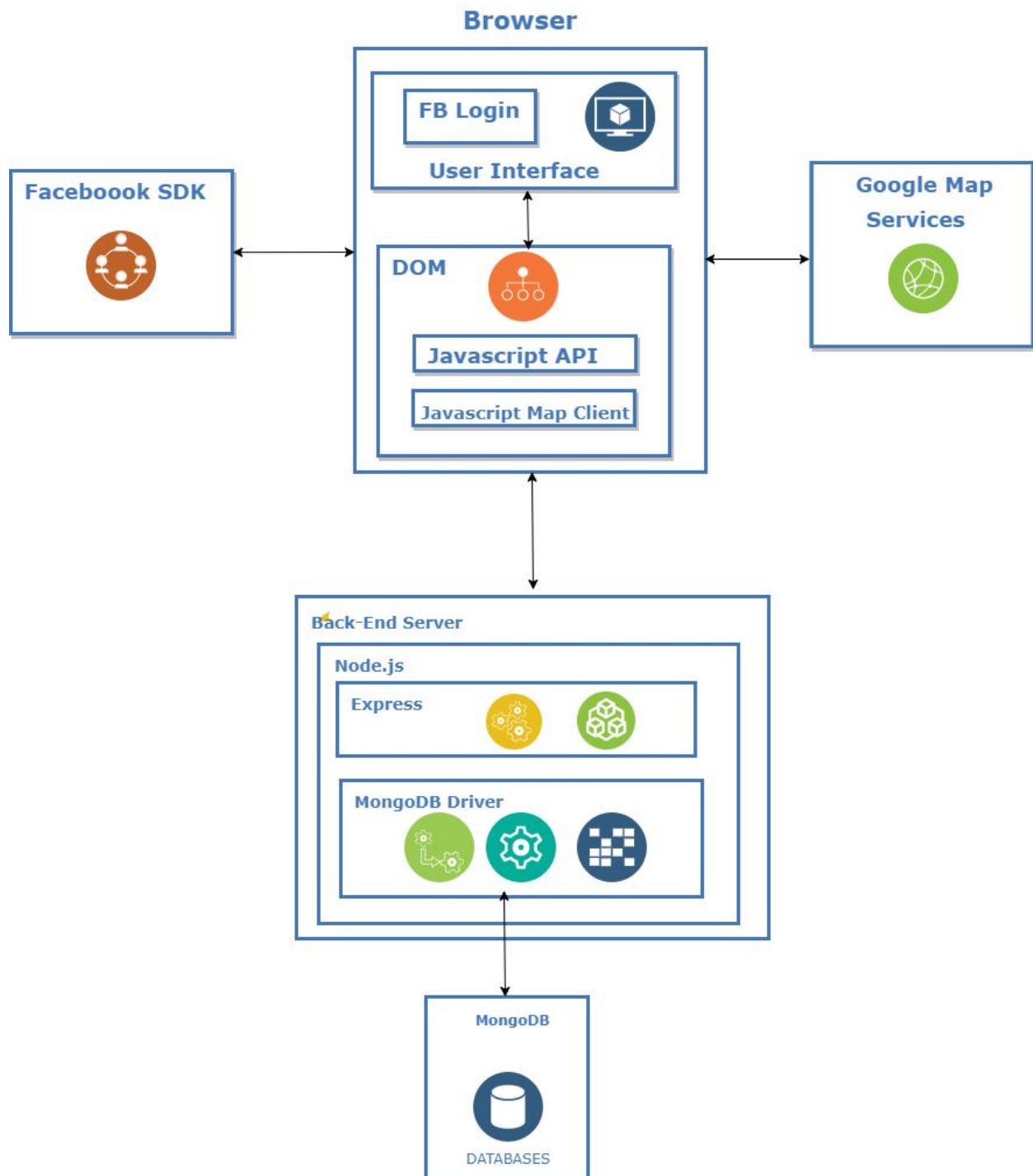


Fig 3. Halo System Architecture

4.0 Implementation

The application is divided into 3 components:

1. Server Side
2. Client Side
3. Back-end Database

Server side:

Server is setup through express and path. Server listens for port 8888 and waits for any clients to join. Server will have complete access to MongoDB and abstract clients from the database. All client requests to access/query the database will be resolved on the server. Server will connect to database to fetch results and these results will be passed on to client via XMLHttpRequest/Response.

Insert into database:

On getting request from Client to insert user's data in the database, server will fetch the data sent by the client. This data is of the given form {UserName:, latitude:,longitude: }. Then it will query to the database to check if an existing entry is present for the Username given in client's data. If an entry is found, it will get updated with the most recent attributes of data,in this case the use location. If no such entry is found, a new entry is inserted in the DB. This is done using Mongoose.js' findOneAndUpdate() method.

FindAll from Database:

On getting findall request from client, server will call the find() method on the database. This returns a list of all the entries present in the database. Data cleansing is done here to make sure client receives only valid entries with all three fields populated with valid data. This list is then sent to the client.

Client Side:

Client side UI is set up in HTML and scripting is done using JavaScript.

Facebook API libraries are embedded to allow user to login via FB_login button. This login call returns with a response that includes the user's name and FB_id. A list of friend using the same app is also returned in this response.

Upon successful login, Google map API is called to get user's current location. When the user uses the application for the first time, it will be prompted to grant permission for sharing the location. This data is collected and stored in JSON format.

Client can then do a XMLHttpRequest to the server to insert the user's data in the database or to get all the entries from the database.

Zoom feature, lets the user to specify a location in the placeholder and zoom into it. On click of the zoom button, the area displayed on map is zoomed to the location specified by the user.

Draw feature, provides a graphical interface for users to draw polygons on the map. The Drawing Tools are a self-contained library, separate from the main Maps API JavaScript code. It takes a set of options that define the set of controls to display, the position of the control, and the initial drawing state.

Distance Matrix feature, it computes travel distance and journey duration between multiple origins and destinations using a given mode of travel.

Backend Database:

MongoDB is used as the sole repository for storing data. MongoDB must be installed and running in background for the application to be successful.

Application uses Mongoose.js module that provides command-line queries like methods that can interact with MongoDB collection/Schema.

Schema used for the application of the given form:

```
{UserName : String, Latitude: Number, Longitude: Number}
```

5.0 System Testing

Feedback System

It is important for a designer to consider that the UI design is basing its updates as well as construction on the user feedback. For this matter, the application has gone through various beta versions with constant involvement of the end-users which was made sure during the designing process. Even after the intermediate product has been launched still, the improvements and later versions of the UI was based on the feedback it has got from its potential users. This process helps in constructing and designing the UI in such a way that it becomes more user-friendly.

Compliance of UI Design with Application Objectives

The UI design should clearly show and depict the software system objectives. The user should be able to distinguish between the websites he is visiting. For this application, the design clearly shows the user what the application is about. Landing quickly on the login page, the user was aware of the medium to connect with people by its visual components with a simple one click access. In this way, the user will be able to surf

faster and extract the related information quickly without having to go through the entire text, just to find out what the website or the system is all about. [3]

In addition to this, the objective of the system was kept in mind during the design of the UI. The user was shown what the application supports and what it does not supports. By spending a short span of time on this application, it clearly shows that it is meant for social connection and not some social media forum. The user is not misled by the displayed information and gets the benefit that was intended by the system during its formation.

Compatibility with Varied Platforms

During the UI design of the application, the UI is compatible at a variety of platforms. Sometimes a user views the information presented in the UIs using different browsers. This normally occurs when the system is web based and not installed on the user's computer framing to this application. In this way, sometimes due to the different adjustments of the browsers, the compatibility is lost. A good UI design for the application was designed and tested across varied platforms to see if it is displayed in the same way or not. Results showed that it is functional over cross platforms.

Proper Amount of Displayed Information

If the system's relevant information is too much for the user, he might try to switch to another website that presents the information in an organized manner. Therefore, for the benefit of the users and the system at the larger scale, the UI displays the right amount of information that is required by a user. So that the information is gathered by keeping in view how much of it is actually necessary for the particular end-user. As a result, the UI was accepted by the end-user and they found it matching to their requirements.

Information Design

The information of the map markers was displayed properly so that the user finds it easy as to who is the intended recipient, the adaptation of information, text, graphical objects, photos, diagrams. Information regarding the relationship between values and change of information was present. Use of colors were done in a proper way when showing the change in the location status and color coding was used to let the users know about the routes they might take in order to meet explore other user's location.

6.1 Conclusion

User interface plays a vital role in software. In terms of visibility, its design and precision hold the primary importance for depicting the exact amount of information for the intended user. For achieving the high goals of progress and effective communication of the software system's perspective, the UIs designed for the application was done under close and calculated supervision of the users who are

people who did like to use this application in real life, and the end users were involved in the process of construction and design. Requirements gathering, although, are separate dimensions, but it played a vital role in designing the UIs. This application has been developed to let users find and explore new people around them over the map interface and it also provide intuitive tools and rich functionalities to filter down the queries the users want to search. It can be seen that the systems that are successful today comply with the software quality standards and they also involve users in the process of UI development. This is of primary importance that the user knows what system they are seeing versus their expectations and the UI designer knows what the user is expecting from them. This all is possible, only if the software quality standards are kept under consideration and the constant feedback is taken from the user, which is the highlighted part of this application.

6.2 Future Work

This working application has been developed to be sent to the Facebook Review Team to get approval for the Facebook user's data by showing the ways on how this application is going to use the user's data to serve the purpose of the application and justifying its usage in the application. The other user data will be used to implement features such as user interests, user photos, user likes, places visited and many more. Facebook Review team will also check that our application is not violating their Facebook's policies. Once the application gets approved, this data of the user from the Facebook APIs can be accessed and thus will be utilized to let visiting users know about the people around them to connect by various aspects and show those data graphically.

7.0 References

[1] <https://developers.google.com/maps/get-started/>

[2] <https://developers.facebook.com/docs/graph-api/>

[3] Mandel, T., The Elements of User Interface Design, New York: John Wiley & Sons, 1999.

[4] Zhu, Ying. "Introducing google chart tools and google maps API in data visualization courses." IEEE computer graphics and applications 32.6 (2012): 6-9.

[5] Nielsen, J., and Molich, R. 'Heuristic evaluation of user interfaces', Conference on Human factors in computing systems, pages: 249-256, ACM, New York, NY, USA, 1990.