Chapter 1

Project Overview

1.1 Project Description

SpartaMaps is a project, which is designed to offer interactive navigation and much more to the application users to know more about the San Jose State University Campus.

Why SpartaMaps?

Navigation

The need for a proper navigation application arises when a student who is the principal user of the system wish to find a shortest path from one source to the destination. In a hectic environment, a student has to have a proper route system, which not only delivers the correct route, but also must show various places of interest like restaurants, grocery outlets, tourist spots etc. to the user. Though, a student can easily reach a specific building, he might wish to see the proper floor-plan of each building to find his way to the correct room. Since the buildings of San Jose State University houses many classrooms and is vast, a need for a floor plan becomes an indispensable entity. Although there are many direction boards, which direct the user towards a room, a user may wish to see that navigation on his device.

Fig 1: Navigation Application in Use

A system has to be defined to help the user see through the building apart from just walking from one place to another. The application must not only direct a person to a specific place, it must also be designed to detail a new student about the place. Enough detailing must be done about a specific place must be done to help every user to get used to the campus and save enough time, trying to navigate through the campus.

Safety

Apart from navigation necessities, let us consider the safety requirements. The school neighborhood is always considered unsafe for students.

Fig 2: Crime Report of San Jose State University

Each day, there are alerts, which are sent to the students to report any crimes happening in the school environment. The San Jose Police department wants the students not to walk alone and be vigilant to strangers near them. What if the students find their friends who are near them and connect with them? This will reduce the strain imposed upon the students to guide them safely.

New students must also be able to connect with other students of their interest and make friends with them. They must also be able to share their pictures, find about events happening around the campus, view other pictures and also socially network.

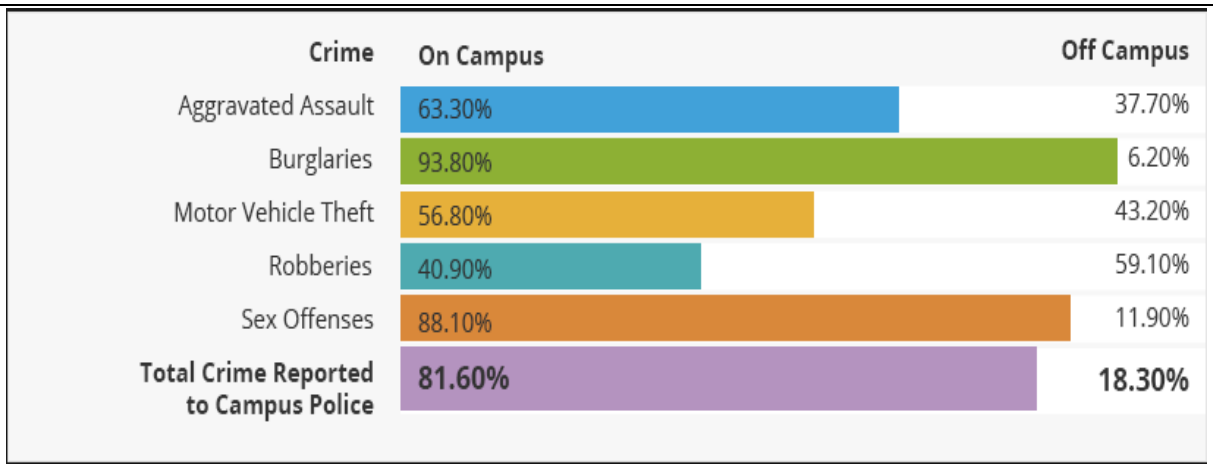

Fig 3: Statistics of on-campus and off-campus crimes

Health

Health experts agree that walking 6,000 steps a day can help to improve health and walking 10,000 steps a day can help to lose weight.

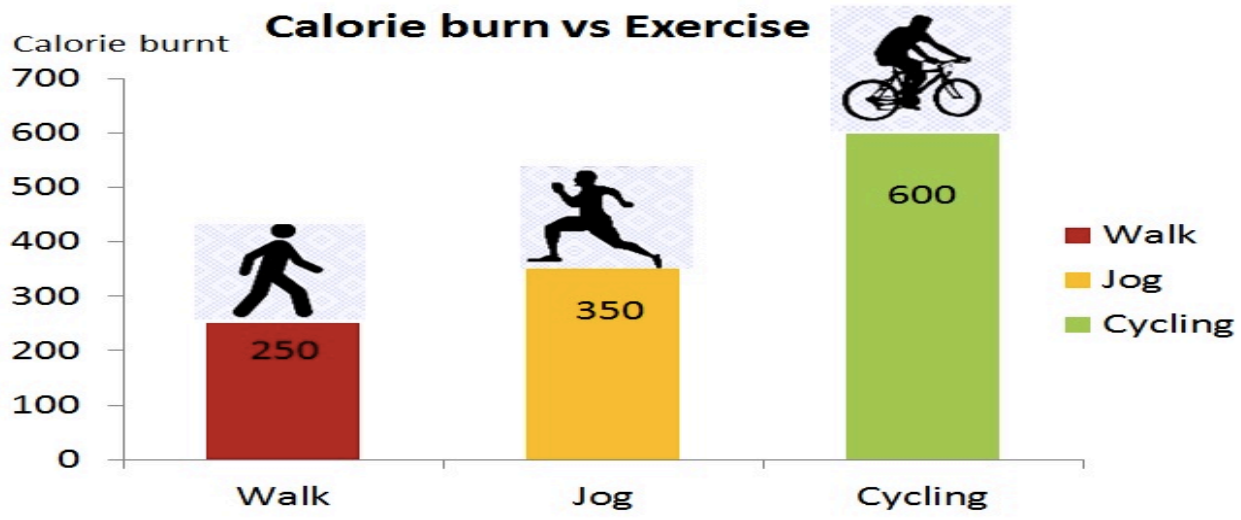

Fig 4: Calories burnt vs Exercise Graph

There is sufficient walking experience for a person who walks or cycles daily from one building to another inside the school campus. Each calorie burnt is a measure of a healthy goal to be accomplished. Why not an app, which offers navigation also show the calories burnt depending on the number of steps walked by the person by the end of his trip?

Imagine a single application, which accomplishes all these tasks in one go. One has to just login SpartaMaps.

How does SpartaMaps help?

The SpartaMaps application was designed to help the students navigate, find places of interest, show their health statistics and also connect with their fellow students. All of these

goals are satisfied in a interactive way to bring the benefit of the application to the fullest use. So, What all features does SpartaMaps contain?

Navigator

SpartaMaps helps the user to navigate between locations inside the school campus.

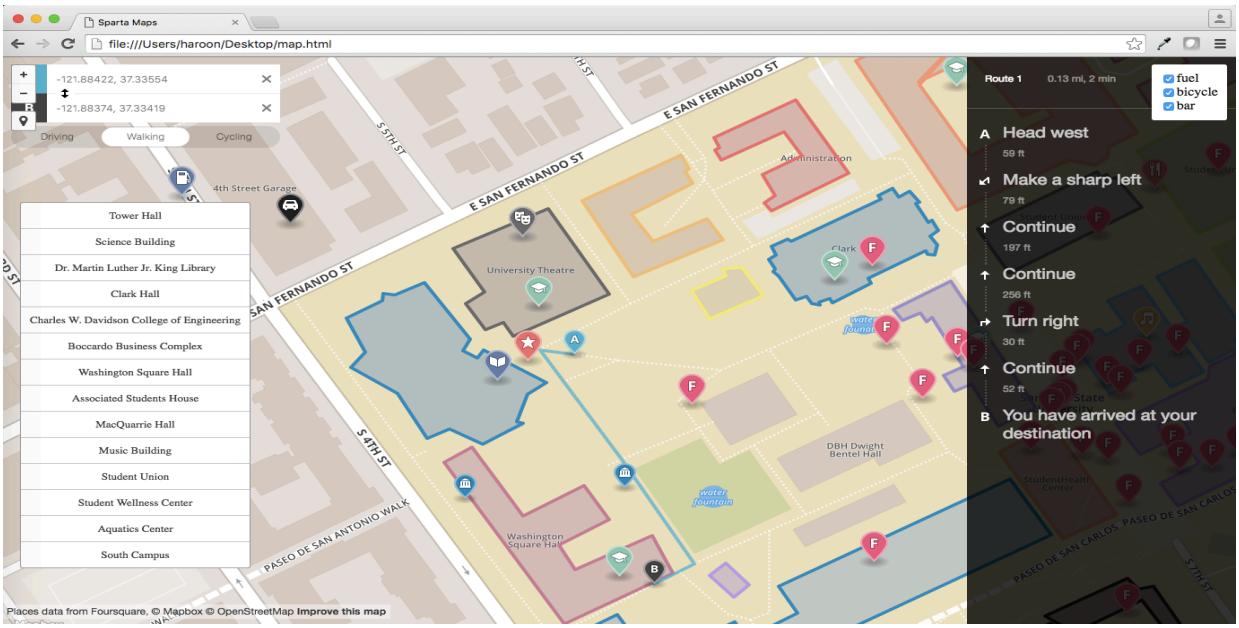

Fig 5: Navigator feature of Spartamaps

The powerful MapBox powered API tracks the shortest route to be covered by the user and aids him to reach the destination in the shortest way possible.

The map navigation is restricted to a limited area surrounding San Jose State University, thus helping the user to have a clear viewport of only the school environment and prevents from navigating to unknown locations. This feature also allows the user to choose his start and end point inside the campus rather than specifying the names of the location explicitly. Giving this option to the user saves time rather than typing the name of the location he might be heading to.

The means of transport can also be selected and the feature specifies the directions to reach the place.

Points Of Interest

Colorful polygonal shapes of the buildings make the UI attractive to the user and also give a detailed structure of the campus. Different markers indicate various points of interest such as

- Grocery Outlets
- Restaurant

- Hospitals
- Music Academy
- Fitness Center
- Swimming Centers
- Tourist spots
- Monuments
- Educational Spots
- ATMs & Banks
- Parking Lots
- Gas Stations

Markers can be selected based on the requirements. Each marker in the application is more than just a marker. Clicking on any polygonal space gives a short description about each building and the marker navigates to a new page, which gives a detailed description of each building.

Fig 6: Different Markers specifying Points of Interest

Gallery

Each marker, when clicked gives the detailed description of each structure.

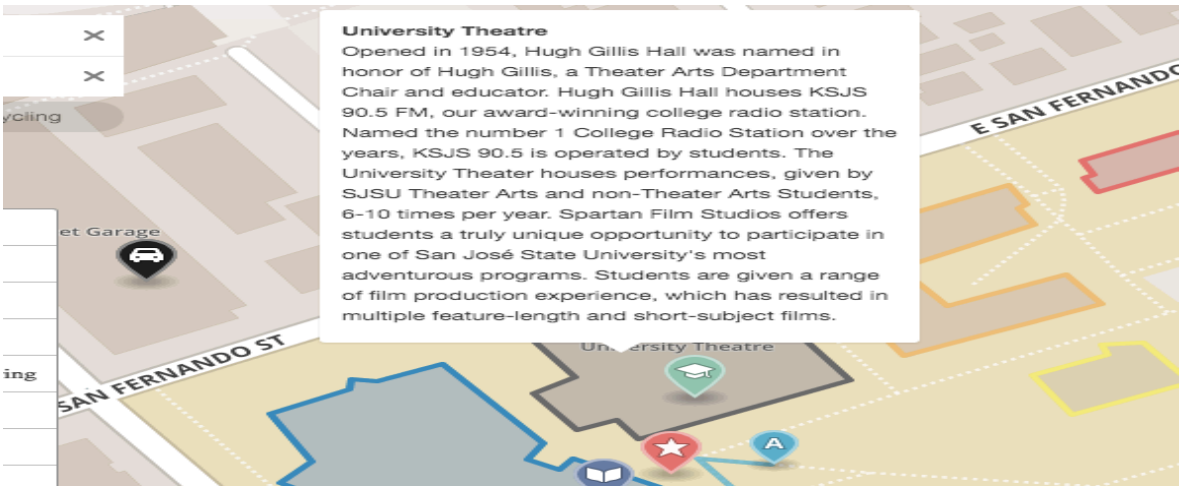

Fig 7: University Theatre Description

The Gallery for each building showcases cool images including 360-degree panoramic videos, floor plans of each building, tweets, yelp reviews, Statistics and YouTube videos in an attractive fashion. There is also a description about the history of the building. There is also a foursquare social integration to help the users know about the social rating of each building. They can also upload their photos and tag the buildings and can see their hash tags and tweets being displayed.

Fig 8: 360-degree view of the MLK Library

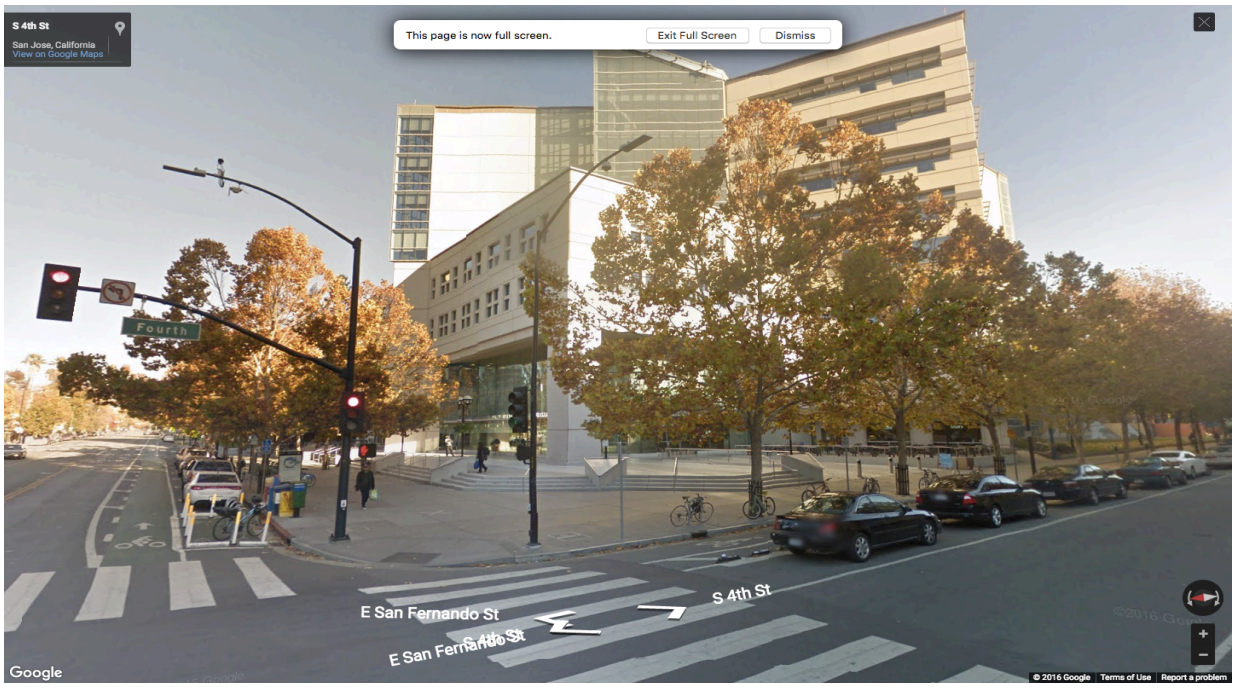

Fig 9: Street view of the MLK Library

Dashboard

The interactive dashboard helps the users to view their profile, their health statistics, the images uploaded by them, the friends whom they connect to, the various events happening around the school campus, Most visited buildings with various interactive features like charts, graphs etc. There is also provision to add events and reminders, which they can access to, Google calendars, favorite pictures, and many more.

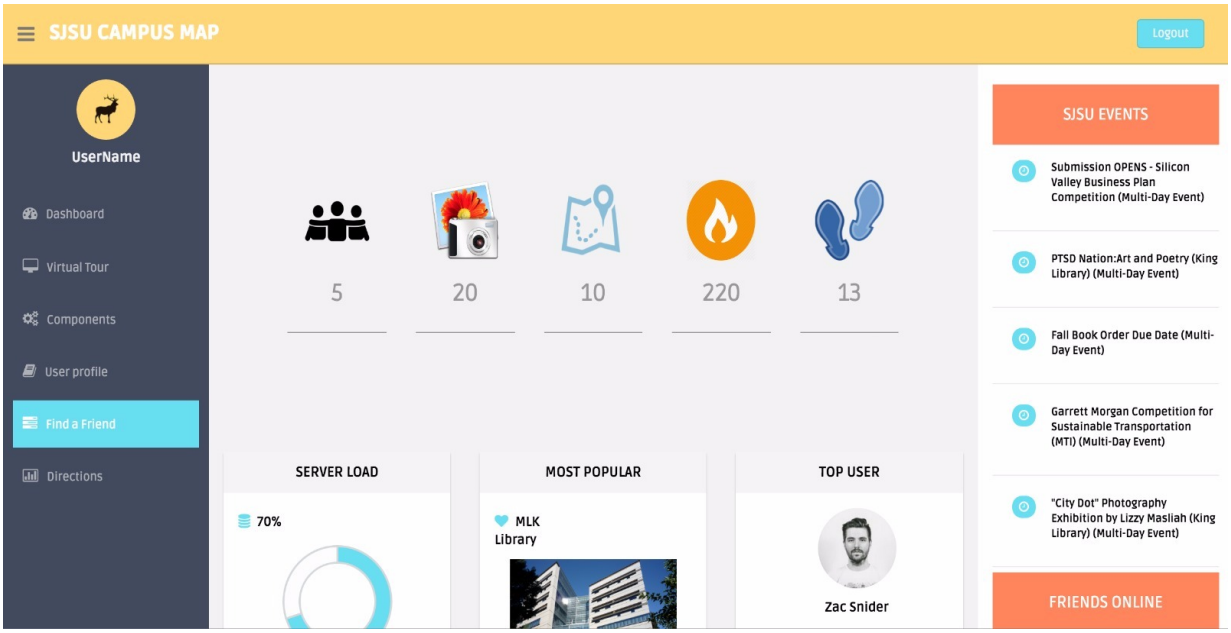

Fig 10: SpartaMaps Dashboard

Health Statistics

The application provides options to know about the distance travelled by a person from one place to another in miles, the number of steps walked and the calories burnt. This option is helpful for the users to keep track of their health and also promotes walking as a best form to burn calories and to induce good health.

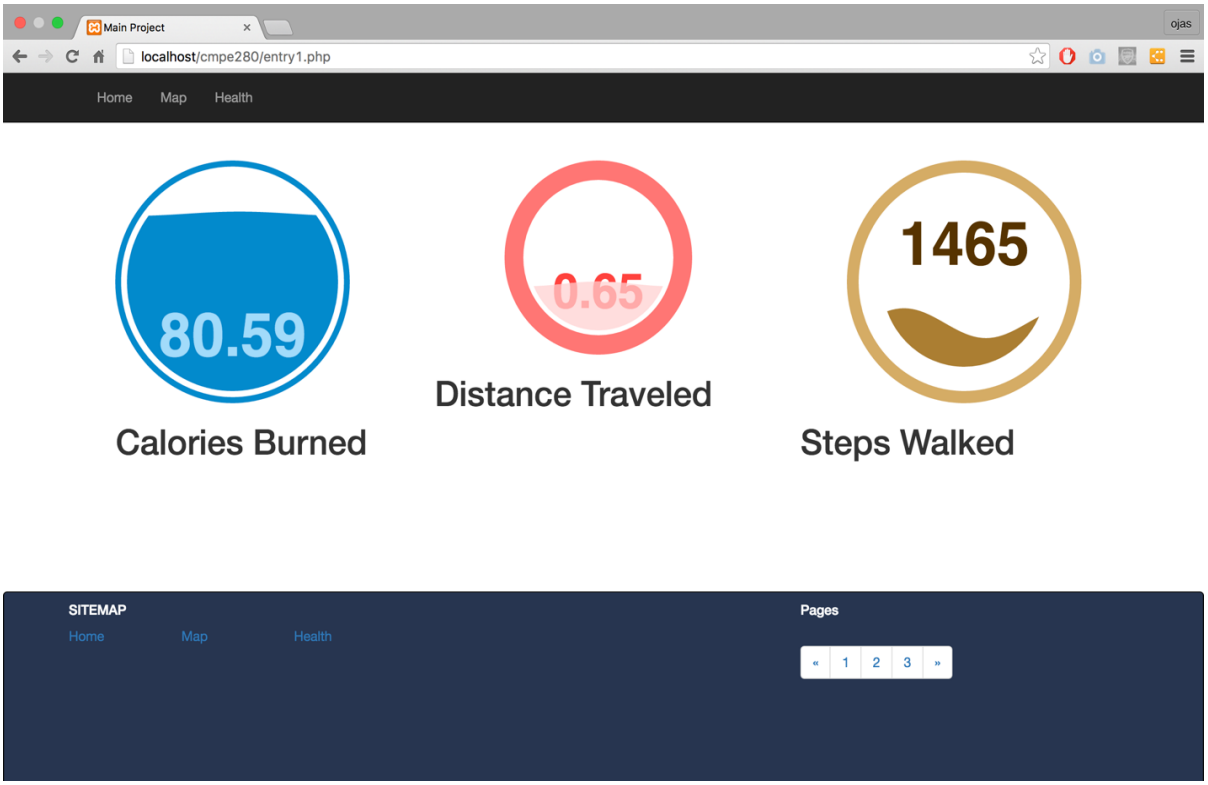

Fig 11: SpartaMaps Health Statistics

Friends Finder

This feature of the SpartaMaps uses the publish-subscribe model to notify the GPS location of the mobile to the maps. By this way, a user can connect with other person and know about his/her location. The location of the user is updated each time the application is loaded. The marker for each person also specifies the details about the user so that the user can know more about the user and may also help to find a new friend.

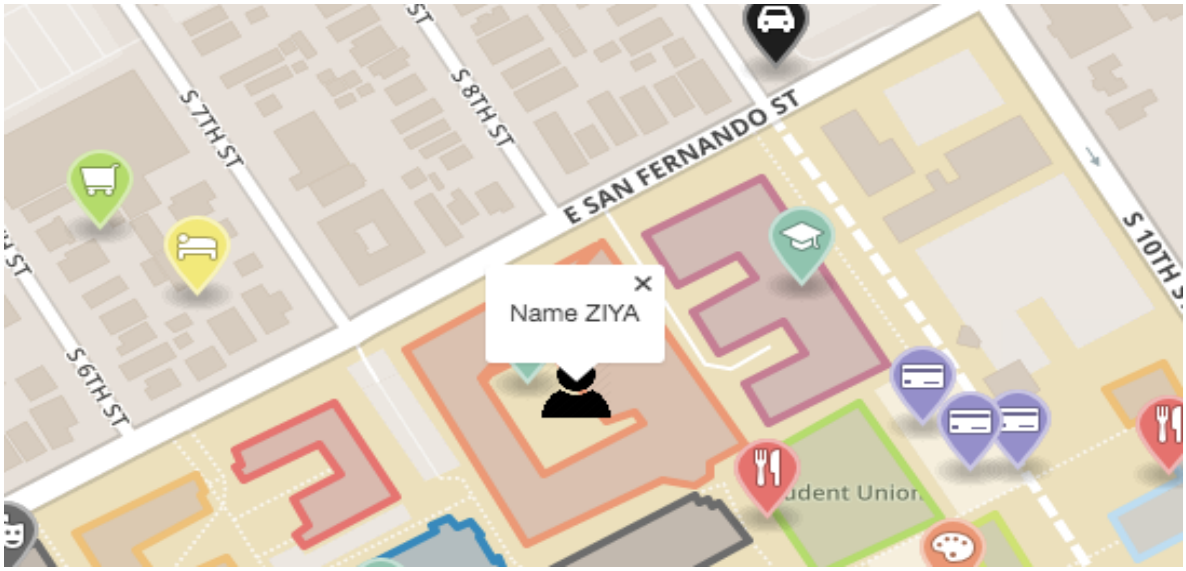

Fig 12: FriendsFinder feature of SpartaMaps

1.2 Requirements

The set of requirements planned and implemented for the SpartaMaps are as follows:

1) Design a navigation system, which would help the user to go swiftly from a source building to a destination building.

2) Design various Marker points to denote various interest places.

3) Design Markers clustering so that only when one attraction is checked, will the markers related to the attraction show up on the screen.

4) Design Polygon Markers such that each polygon gives a detailed analysis of the building contained in it.

5) Design source and destination markers such that the route gives a detailed list of the directions to be followed.

6) Analyze each building inside San Jose State University, giving social network reviews about the place.

7) Display a gallery of images related to each building including the 360-degree experience of each building along with a inner floor plan.

8) Track the steps travelled, distance covered by each user when travelling from one place to another and calculate the calories burnt.

9) Design a interactive dashboard to help the user change his personal details and the way he has to be displayed on the main page.

10) Enable a login feature to track individual users and a signup feature to add new users to access the application.

11) Enable a friend finder feature, which will display the location of the user using the GPS location.

12) Enable the user to view the school events, mark events on the calendar, and upload images in his personal space.

13) Track the path when a user is travelling from one place to another and also give the user options to choose his means of travel – Walking, Driving or Cycling.

14) Specify the clear directions for navigation.

15) Design a homepage to embed all these features as a complete package.

Chapter 2

Web UI Principles

2.1 Web UI Requirement Principles

2.1.1 Job Shadowing

Job shadowing is experiencing the software from the shoes of the user of the system.

Principal users

User 1 – Current Student on campus
User 2 – Prospective Student
User 3 - Current Student off campus

Intended Questions

1) How do you navigate inside your school campus?
2) What is the frequency of your travel between the buildings on SJSU campus?
3) Do you travel alone to your destination?
4) What application do you use to navigate oncampus or offcampus?
5) How often do you experience difficulty in finding a building?
6) How often are you likely to find places of interest in the application that you currently use?
7) Do you use wearable technology to track your steps and calories burnt?
8) How often have you found difficulty in navigating through a building in SJSU?
9) Do you think it would be helpful if you could find a friend near you to walk you to home?
10) Do you check Yelp or foursquare reviews about a place before going?

User 1

1) How do you navigate inside your school campus?
   I don't have a proper application for navigation. I just refer the map of SJSU given on the school's site to find directions

2) What is the frequency of your travel between the buildings on SJSU campus?
   I used to visit like 5 buildings per day since I have 2-3 classes per day on different buildings.

3) Do you travel alone to your destination?
   Yes, I travel alone and I prefer to do so in the mornings. When it is dark, I would love to get a company.

4) What application do you use to navigate oncampus or offcampus?
   I use Google Maps mostly for navigation, though it doesn't show the inner campus paths.

5) How often do you experience difficulty in finding a building?
   In the start of my school life, I found it very difficult to find my way to each building. I used to ask other people or at times call my friends to ask for directions to reach a place.

6) How often are you likely to find places of interest in the application that you currently use?
   It would be more efficient if I can just get all the places of my interest on one screen shot.

7) Do you use any wearable technology to track your steps and calories burnt?
   Oh. Yeah. I use Fitbit flex to track the number of steps travelled each day.

8) How often have you found difficulty in navigating through a building in SJSU?
   Very difficult at times, since many navigator systems don't show the paths inside the buildings.

User 2

1) How do you navigate inside your school campus?
   I haven't joined school yet. But, it would be better if there was a proper navigation system to save time.

2) What is the frequency of your travel between the buildings on SJSU campus?
   I haven't yet received the schedule of my classes. But, I do think that I might need to travel between buildings once my semester starts.

3) Do you travel alone to your destination?
   No. I always find some company.

4) What application do you use to navigate oncampus or offcampus?
   Google maps navigation would be my best bet.

5) How often do you experience difficulty in finding a building?
   Haven't experienced this kind of a situation before.

6) How often are you likely to find places of interest in the application that you currently use?
Haven't experienced this kind of a situation before.

7) Do you use any wearable technology to track your steps and calories burnt?
No. But I am planning to buy some wearable device to track my calories.

8) How often have you found difficulty in navigating through a building in SJSU?
Haven't experienced this kind of a situation before.

9) Do you think it would be helpful if you could find a friend near you to walk you to home?
Yes. I would prefer to walk with friends rather than alone.

10) Do you check Yelp or foursquare reviews about a place before going?
Yes. I do check Yelp reviews.

User 3

1) How do you navigate inside your school campus?
I am used to the surroundings of the school campus, since this is my third semester of school life.

2) What is the frequency of your travel between the buildings on SJSU campus?
I don't travel much between buildings as I have only one class per day.

3) Do you travel alone to your destination?
Yes. I travel alone

4) What application do you use to navigate on campus or off campus?
Sometimes I use the navigation system in my mobile phone.

5) How often do you experience difficulty in finding a building?
I experience difficulty during the start of my semester sometimes, when I have classes in a different building. I am still unaware of many buildings inside my school campus.

6) How often are you likely to find places of interest in the application that you currently use?
Rarely

7) Do you use any wearable technology to track your steps and calories burnt?
No. I am not a fitness freak.

8) How often have you found difficulty in navigating through a building in SJSU?
Many times inside library, when I have to find a specific study room.

9) Do you think it would be helpful if you could find a friend near you to walk you to home?
Yes. It would be absolutely stress bursting to talk to a friend while walking.

10) Do you check Yelp or foursquare reviews about a place before going?
Not much. Sometimes, I look onto tweets about the place.

2.1.2 Project Selection Tools

Project Selection Criteria	Very Low		Moderate		very High
	1	2	3	4	5
What is the potential for positive impact on your customers?				4	
What is the level of alignment to your organization's vision and strategy?				4	
To what degree will this make your work easier or more interesting?					5
How much control do you have over the process you want to address?				4	
What is the potential or cost reduction or cost avoidance?			3		
What is the potential for time savings?					5
What is the level of urgency related to this project?		2			
What is the likely hood that this effort could be completed in 3 months or less?			3		
what is the availability of existing data related to this process?				4	

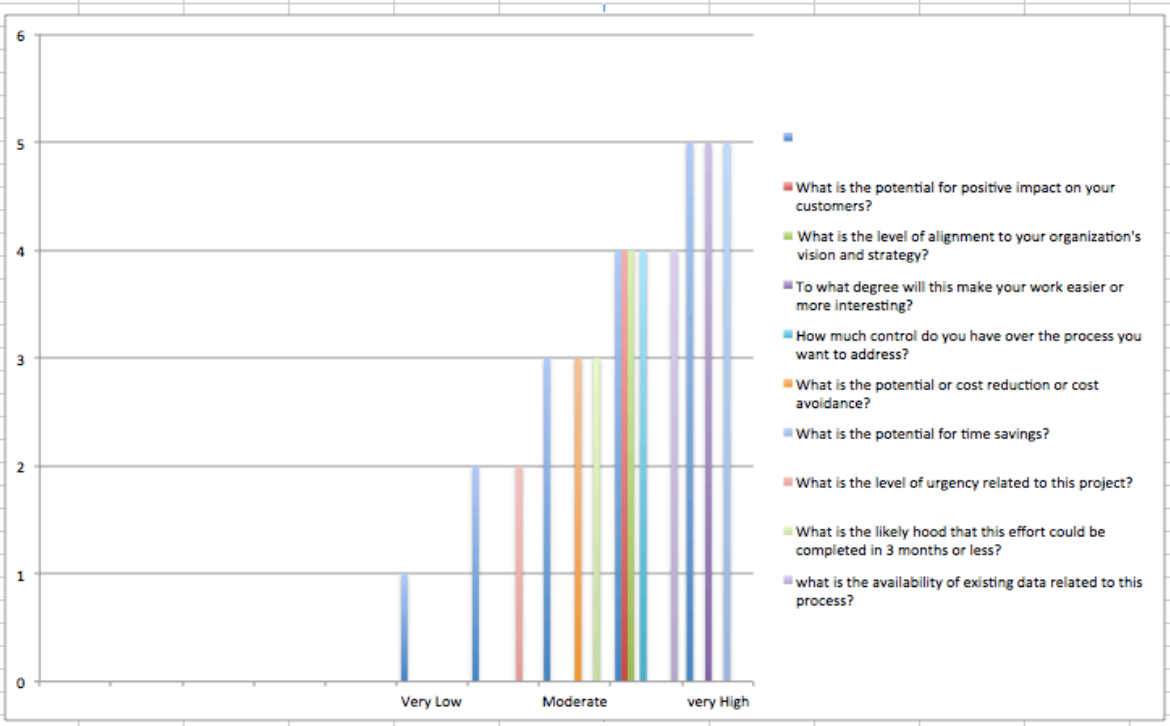

Fig 13: Project Selection Criteria

2.1.3 Voice of Customer

	Customer Requirements	Plan	Develop	Market	Deliver	Support
Better	Treat me like you want my business	2	3	3	2	5
	Deliver products that meet my needs	3	5	3	4	2
	Products/services that work right	4	4	4	3	3
	Be accurate, right the first time	3	5	4	4	3
	Ease of Use	5	3	5	3	4
	Fix it right the first time	2	5	4	3	2
Faster	I want it when I want it	4	3	1	2	3
	make commitments that meet my neec	3	4	3	5	1
	Meet your commitments	5	3	4	3	2
	I want fast,easy access to help	4	5	5	2	4
	Don't waste my time	3	3	2	3	5
	Timeliness	5	4	3	2	2
	If it breaks, fix it fast	2	3	3	4	3
Cheaper	Deliver irresistable value	3	4	4	3	4
	Help me save money	3	5	4	4	2
	Help me save time	3	3	4	5	1
	Reduce cost to own or use	3	2	4	5	5
	Total Weight	57	64	60	57	51
	Importance - (1-5)					

Fig 13: Voice of Customer

2.1.4 Personas

User 1

As a prospective student, I want to know about each building and its facilities so that I become accustomed with the college environment.

User 2

As a current student, I need to navigate around each building in San Jose State University environment with ease so that I can reach my destination quickly.

User 3

As a current student using the application, I need to know about the distance travelled and the calories burnt so that I can know about my energy expenditure.

User 4

As a current student using the findfriends feature in the application, I want to make friends with my fellow Spartans so that I get company and don't have to walk alone.

User 5

As a user of the application, I want to get social networking reviews about a place so that I can know about the place better.

2.2 Web UI Design Principles

2.2.1 Wireframes

Home page

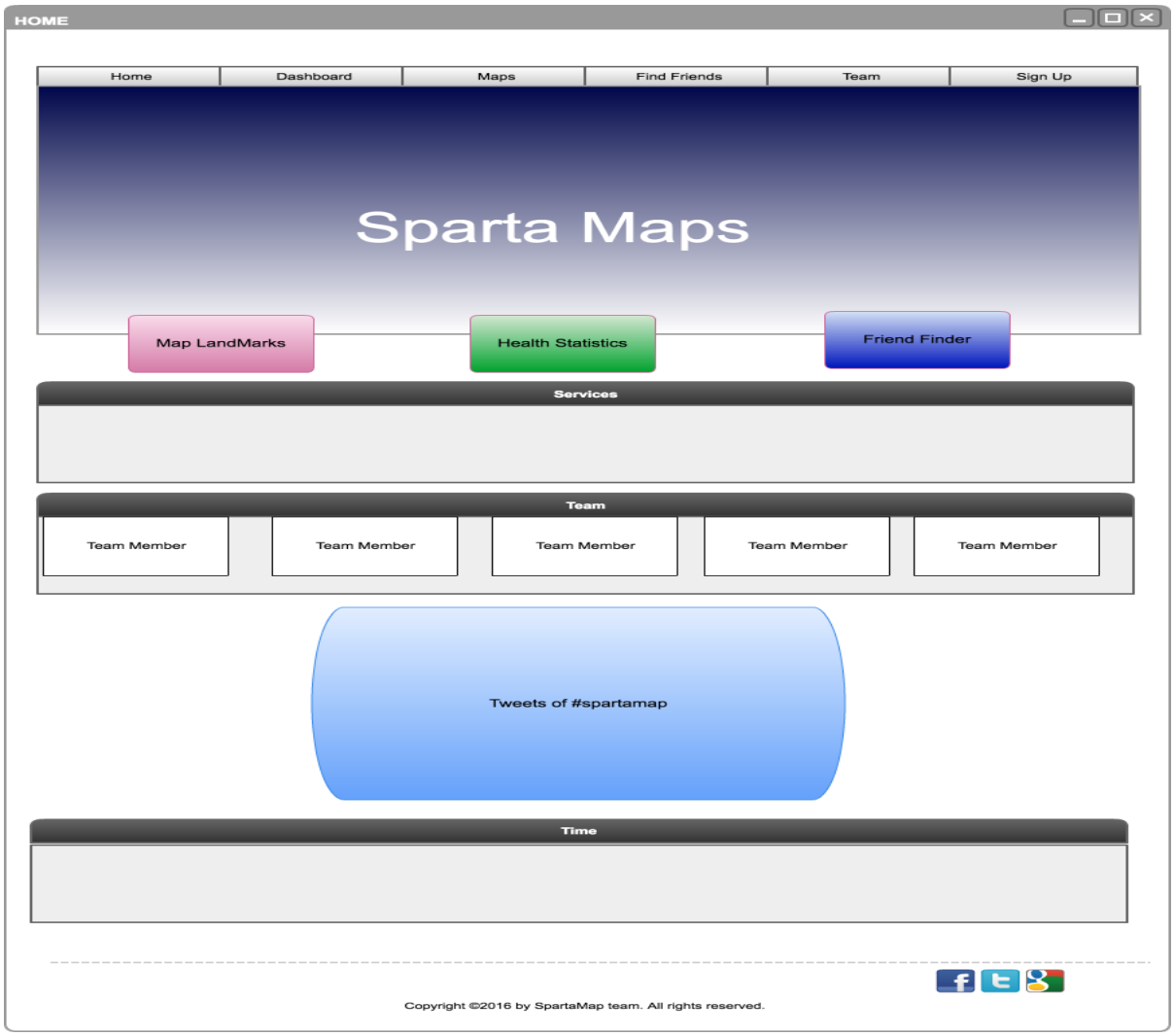

Fig 14: Homepage

This is the wireframe of the home page where User has plenty of things to navigate and use. This page is divided into 5 sections where first gives 3 options of Map Landmarks health statistics and Find friend. Next are the services section followed by the Team section where all the designers of this project are displayed. Next section is where the Social media comes in picture, this section shows the live data feed of tweets about #spartamap. Final section is of current time and some other details.

MAP page:

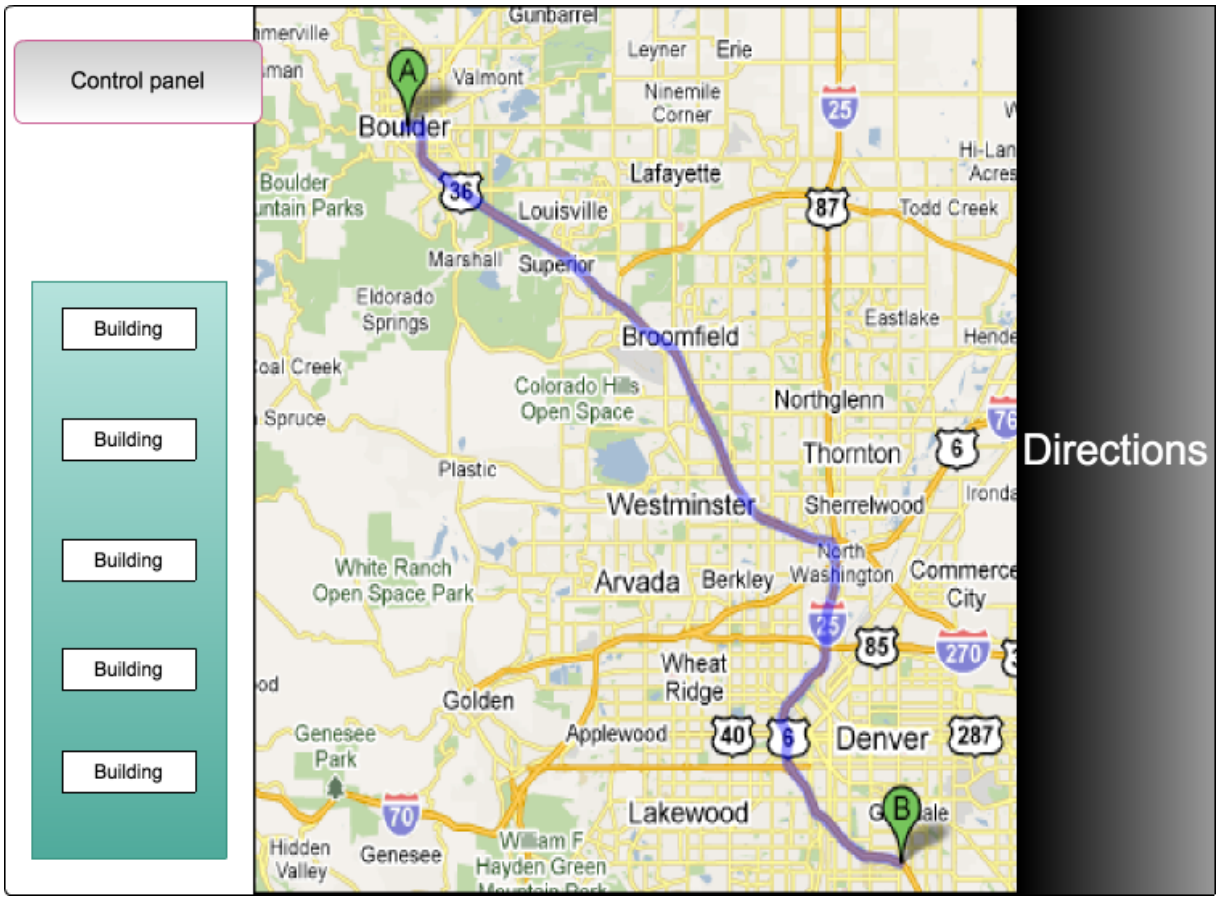

Fig 15: Navigation

This is the Map page where there are multiple options are provided to the user. Here first thing is control panel. In this there are Start and End positions provided. You can choose an option of walking or cycling in this section. Furthermore, there are options to zoom in zoom out and define position option. Below that there is a list of buildings in the campus you can click and choose any building. To the right of that is the direction section. Here you can follow the directions to reach the destination. It is one of the main page of the project and one of the main functionality of the project. In this page the default view is all the markers giving information about the each building. From control panel you have to choose your starting point and destination point.

Health Charts:

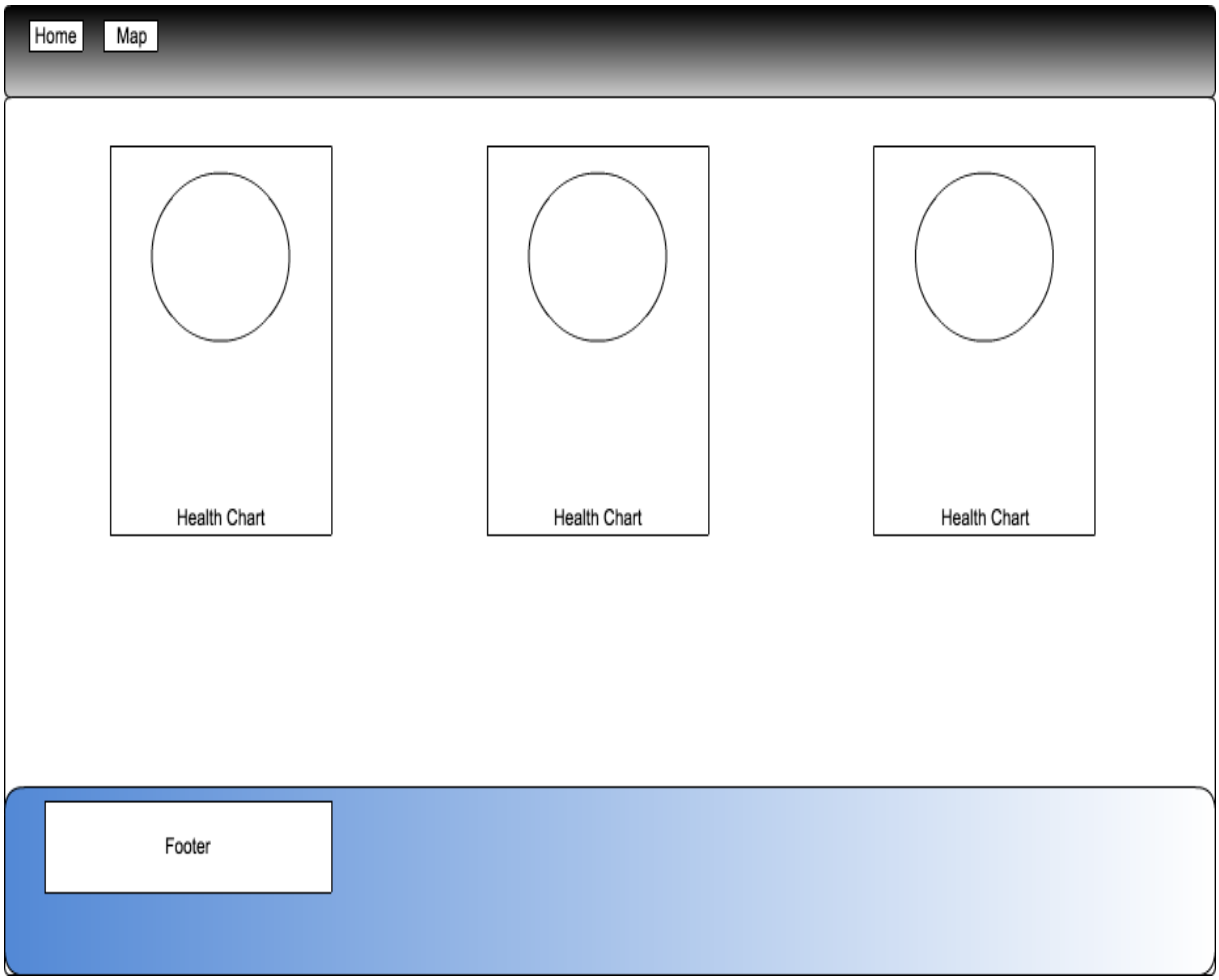

Fig 16: Health Statistics

Now this is another exiting feature of out web application where we will be providing the user with the Health statistics of the last tour. There are three columns here providing user with the 3 different statistics. The first is, Calories burned throughout the walking or throughout the tour. Next is the total distance that he has covered in the tour. And the last one is the steps he has walked during whole tour. All the charts are displaying the statistics of his last tour. There is a facility of navigating to the home page from the links provided at the top of the page. The home will take you to the main page or the first page.

SignUp Page:

This is the sign up page. Now, when you click on sign up page it will redirect you to login with your username and password. There are some other details he would have to enter in order to signup. The user also requires providing his email id as that will be considered as the unique id. After submitting button the user can see his section name and a personalize view for the web application.

Fig 17: SignUp Page

Dashboard Page:

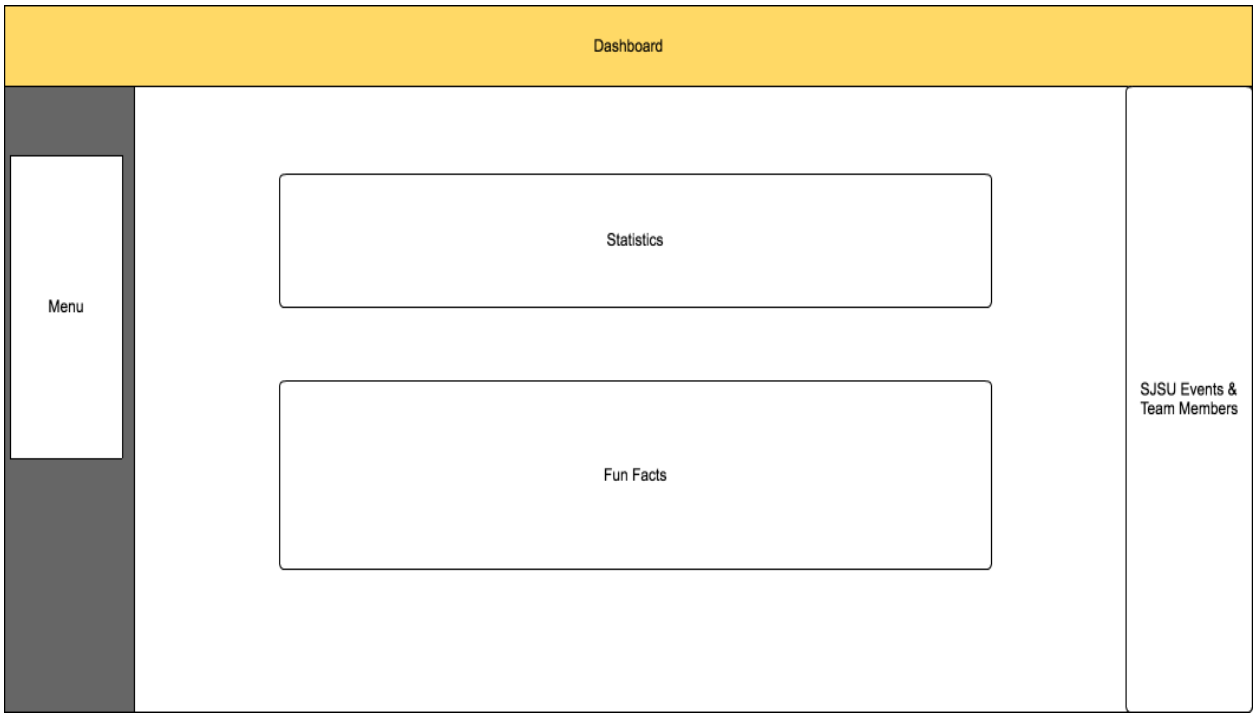

Fig 18: Dashboard

Now here is the Dashboard page. It provides user with the personalize touch to the web application. Now the screen is divided into many sections as you can see. The first is the header and it has a option of menu to hove over and navigate to different pages. Then there is a section of Menu items providing various other features. You will have to select the Health option here to go to health page of web application. Also there is a facility to see add and remove some images. This is totally users personal gallery which wont be accessible for other users as well as other pages. User then has some statistics in the center of the page. Here, he is provided with the numbers like how many friends he has, Calories burned, steps walked and

images in his gallery. The next part is Fun facts here he is provided with some of the facts about the web application such as top user, instagram updates etc. The last part is the section where he can see the current events happening in SJSU and the team members of the web application development team.

FindFriends

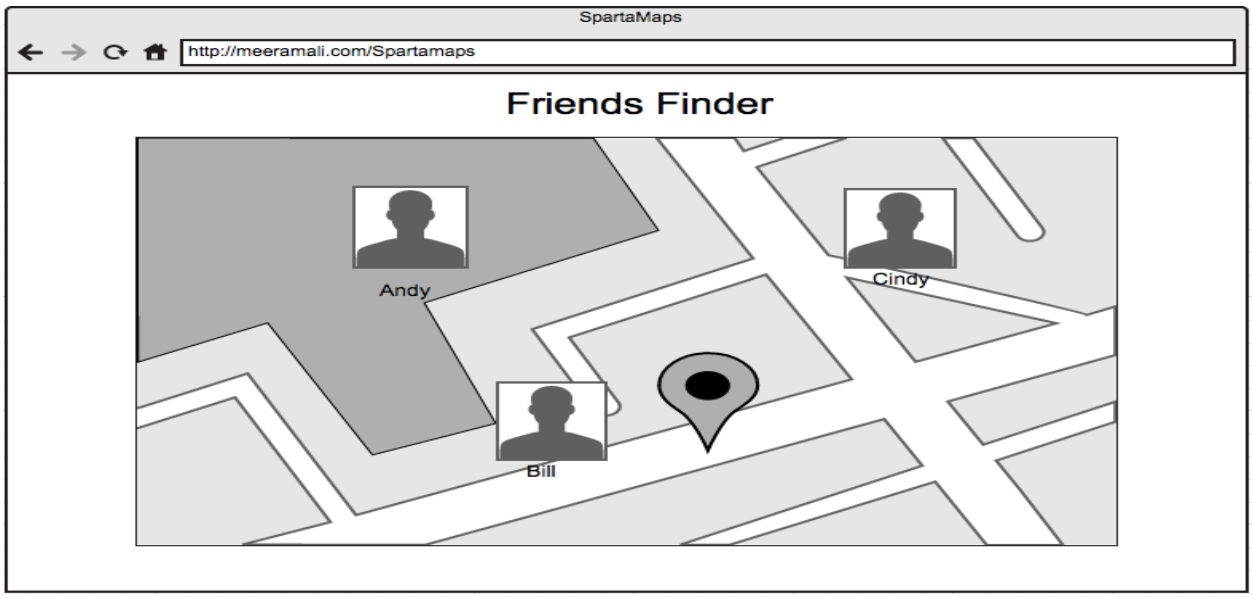

Fig 19: FindFriends

Gallery

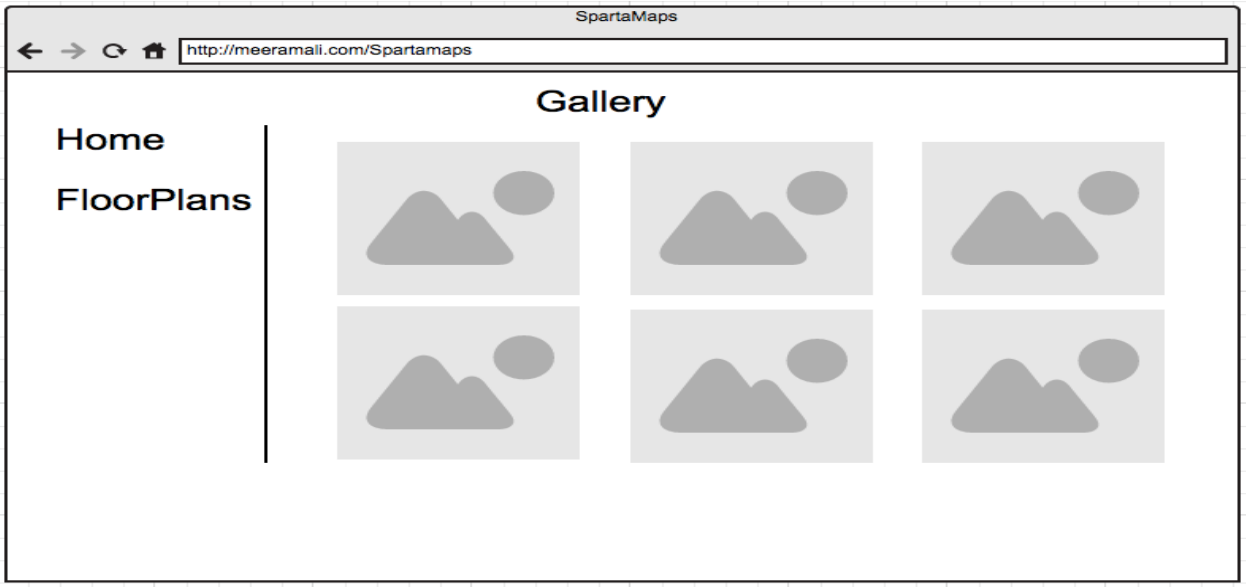

Fig 20: Gallery

Building Description

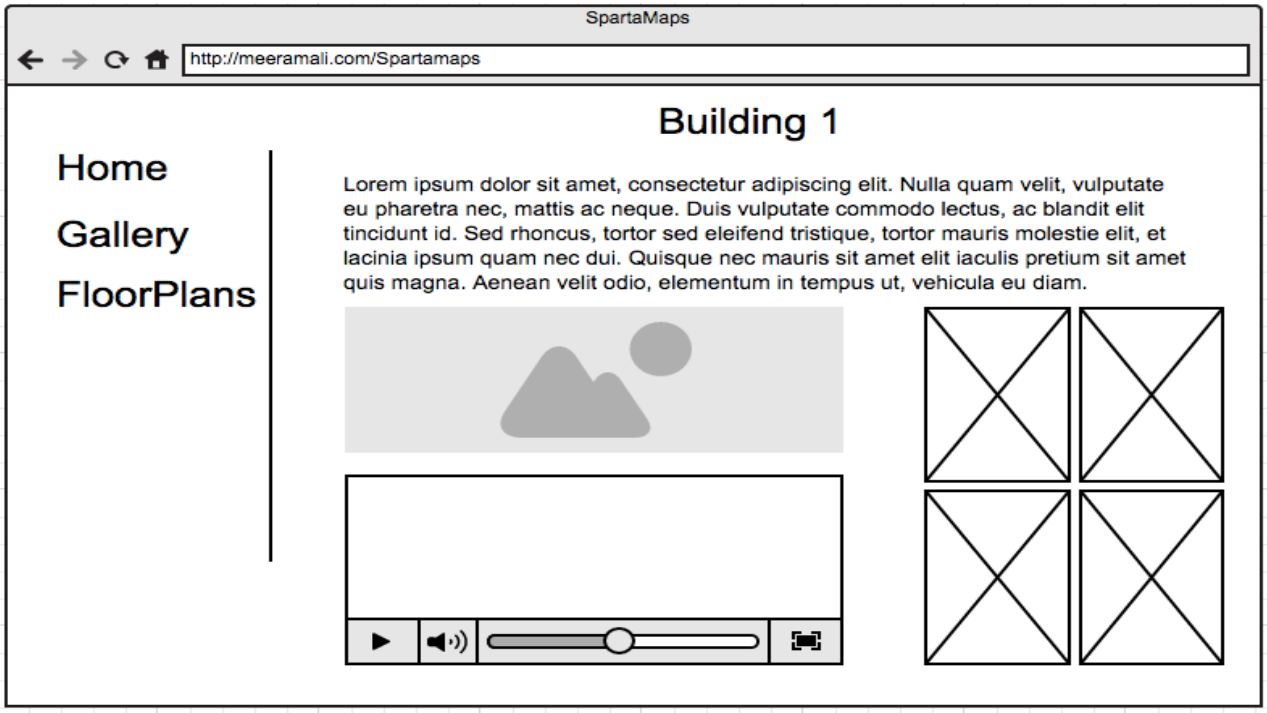

Fig 21: Description of Building

Full Web App Wireframe:

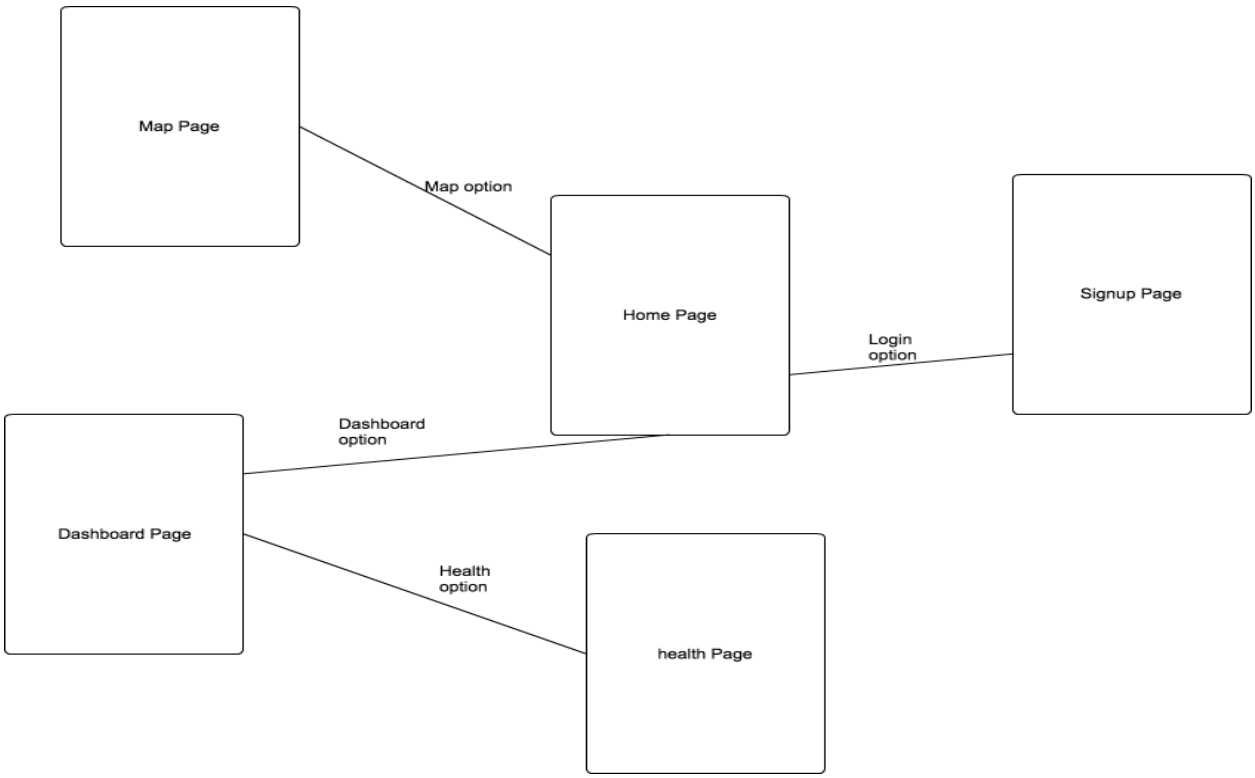

Fig 22: Full Wireframe Structure

This is the Full wireframe with each page in the Web application. Here it gives you the complete idea of the flow of web app. As explained earlier the main page here is the home page thorough which you can navigate and go to other pages. Although each page has their own structure this is an overview of all what's happening inside the web application.

2.2.2 Storyboard:

Scenario 1:

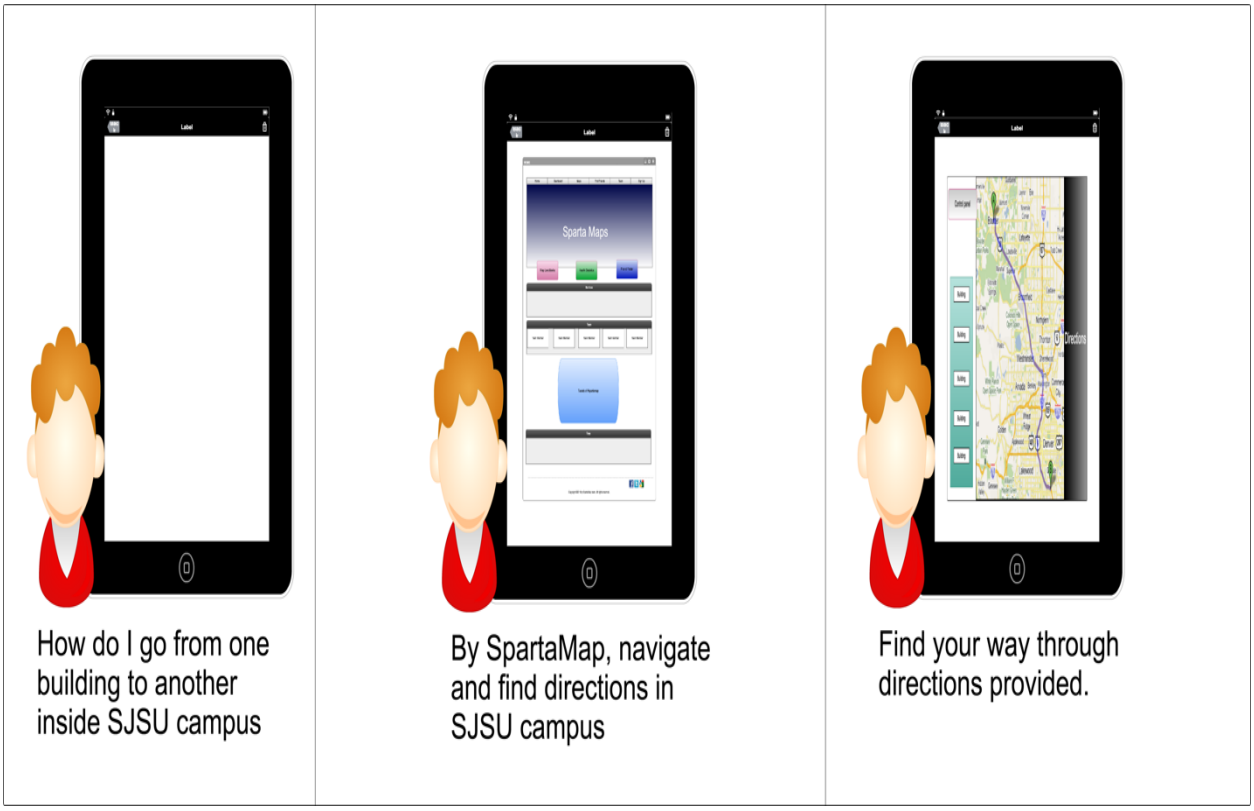

Fig 23: Scenario 1

As you can see the first scenario is a person or a student in our case who wants to find out his way to go from one building to another through SJSU campus. In the next step he joins and access the Sparta Map web application. Where he can find it very easily to go from one spot to another.

Scenario 2:

In this scenario a student is curious to find out who amongst her friend are there in the campus around him. So she can join and login to Sparta maps and navigate through it. It provides a facility to find friends around her in campus.

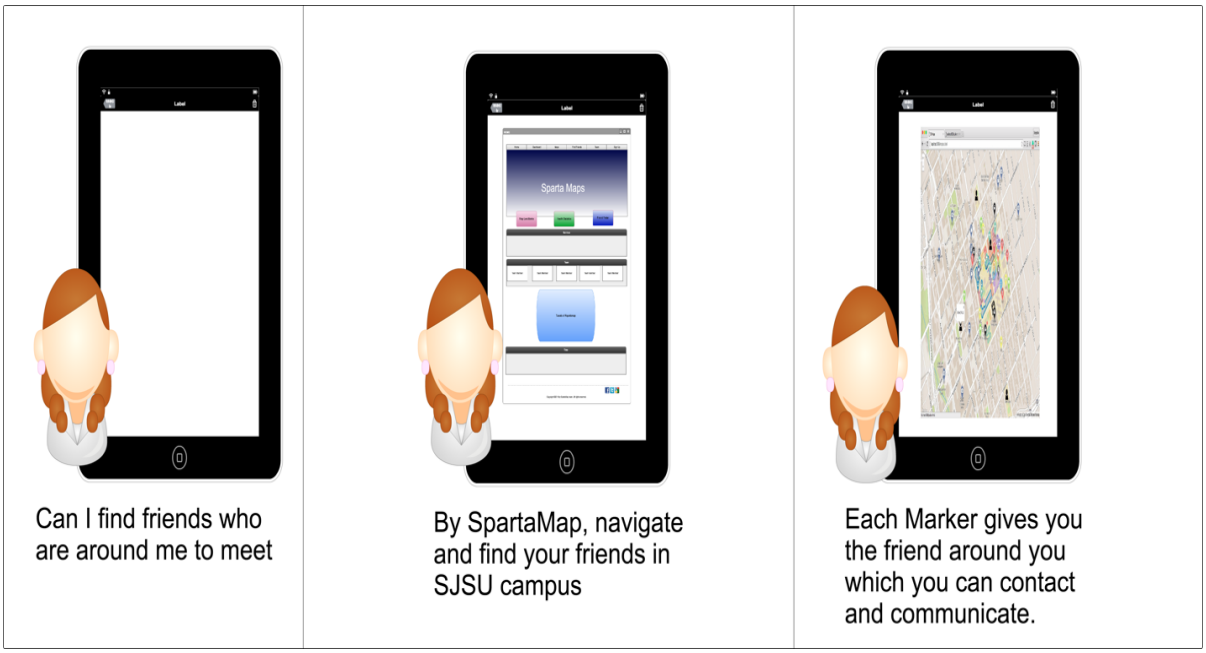

Fig 24: Scenario 2

Chapter 3

Application Design

3.1 High Level Architectural Design

The high-level architecture (HLA) is a general purpose architecture for distributed computer simulation systems. Using HLA, computer simulations can interact (that is, to communicate data, and to synchronize actions) with other computer simulations regardless of the computing platforms. The interaction between simulations is managed by a run-time infrastructure (RTI). HLA is an interoperability standard for distributed simulation used to support analysis, engineering and training in a number of different domains in both military and civilian applications and is the standard technical architecture for all US Department of Defense simulations.[5][6]

Now basically this takes the overall view of the full system. It gives you the insights of the system but still hides the details. It is concentrated to the customer and not the developer. Here each layer is showed and each layer has a separate responsibility.

High-level architecture of SpartaMap:

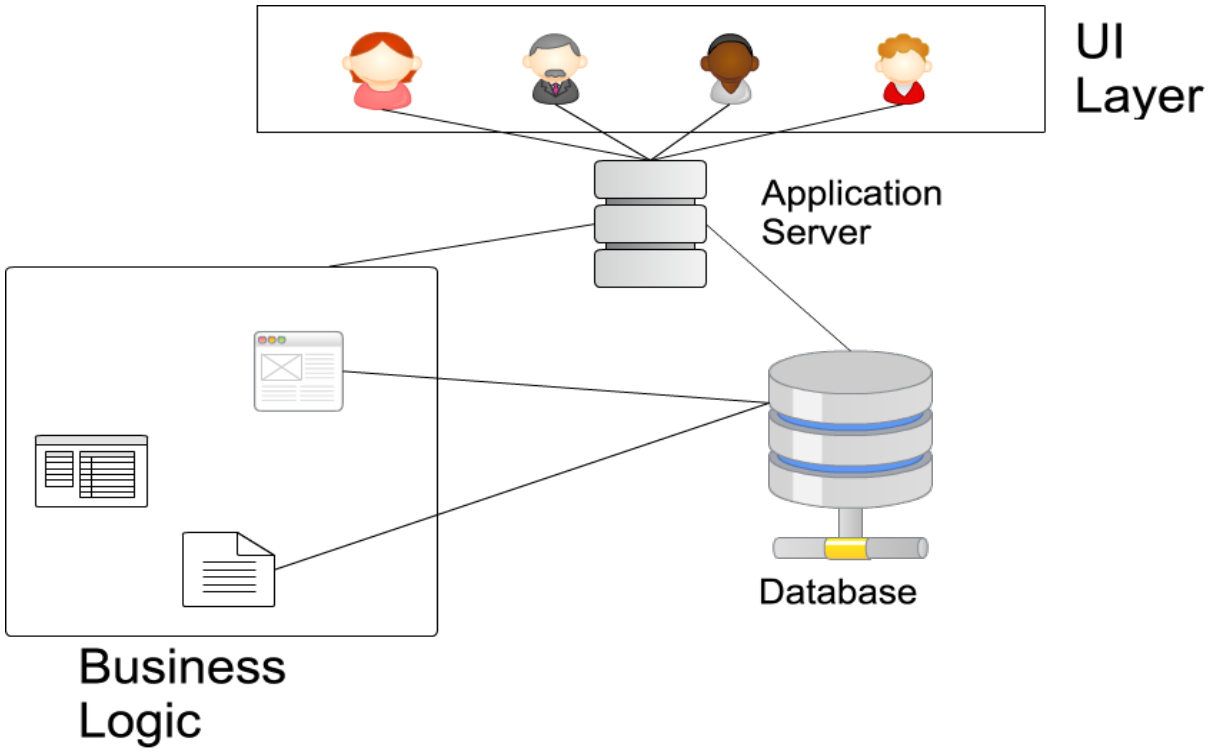

Fig 25: High Level Design

Now this is the high level architecture of SpartaMaps, as you can see this is divided into three layers with each having its own responsibility. The bottom layer is the Database and the

business logic. This resides on the Application server. The business logic saves data on to the database. It is accessible by the application server. This server decides weather to access database and modify it. Then is the UI layer, this includes all the customers of the web application. Now, user requests application server for the information now application server contacts the business logic and sees if the query that customer has fired can be executed or not. Then according to the business logic and rules defined in business logic the application server will access the database and find out the desire data set. It will then send the response to the customer with the view of his desired or requested dataset. To explain more, we can take a scenario of student requesting for a particular map. Customer will request the application server. It will then look into the business logic to see if with fired query does the map exists. If yes then Application server will go in the database seek the dataset for the map and will send the view of that map to the customer.

3.2 Data Flow Diagram

Context Level Diagram

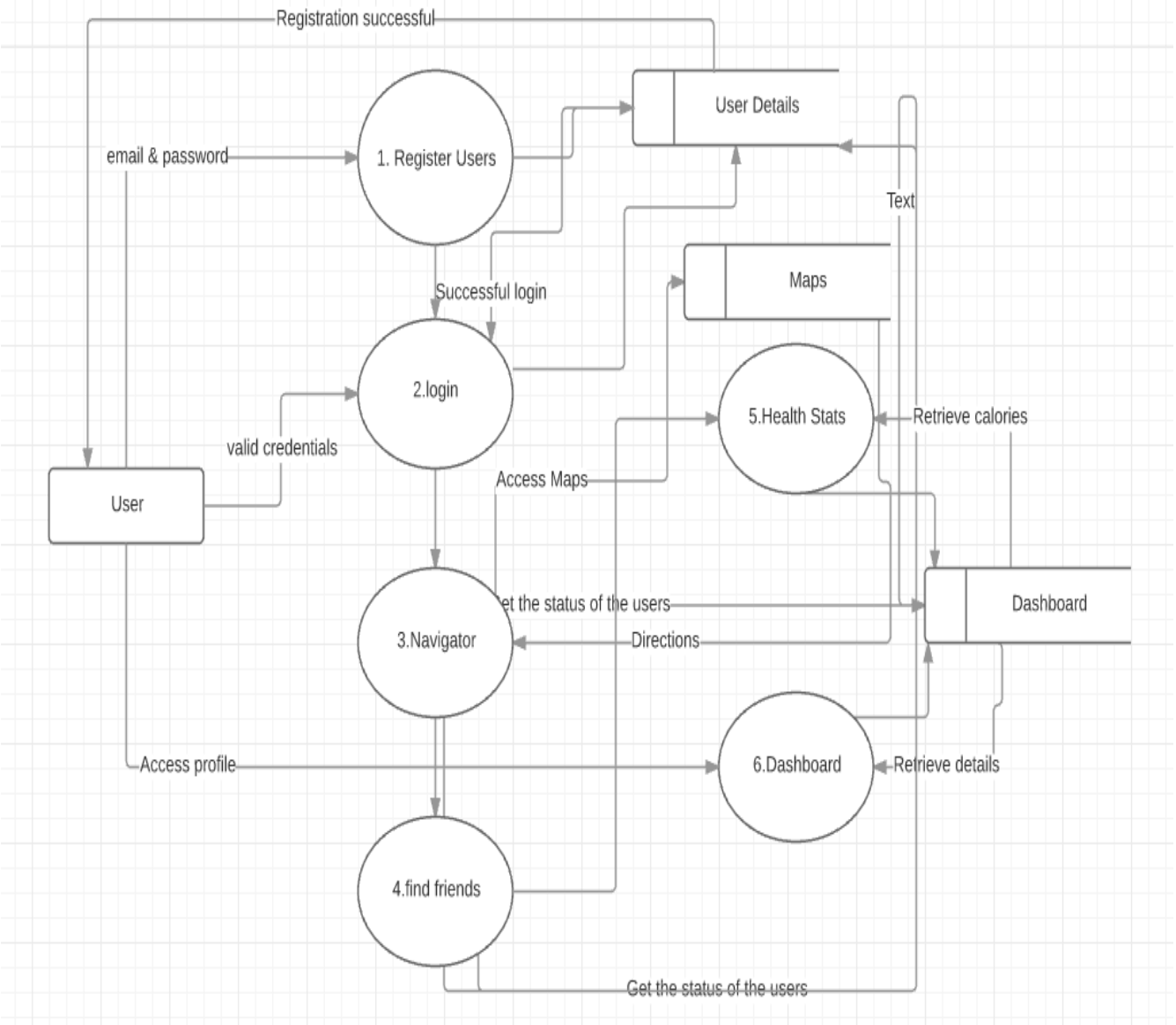

Fig 26: Context Level Diagram

1. Registration

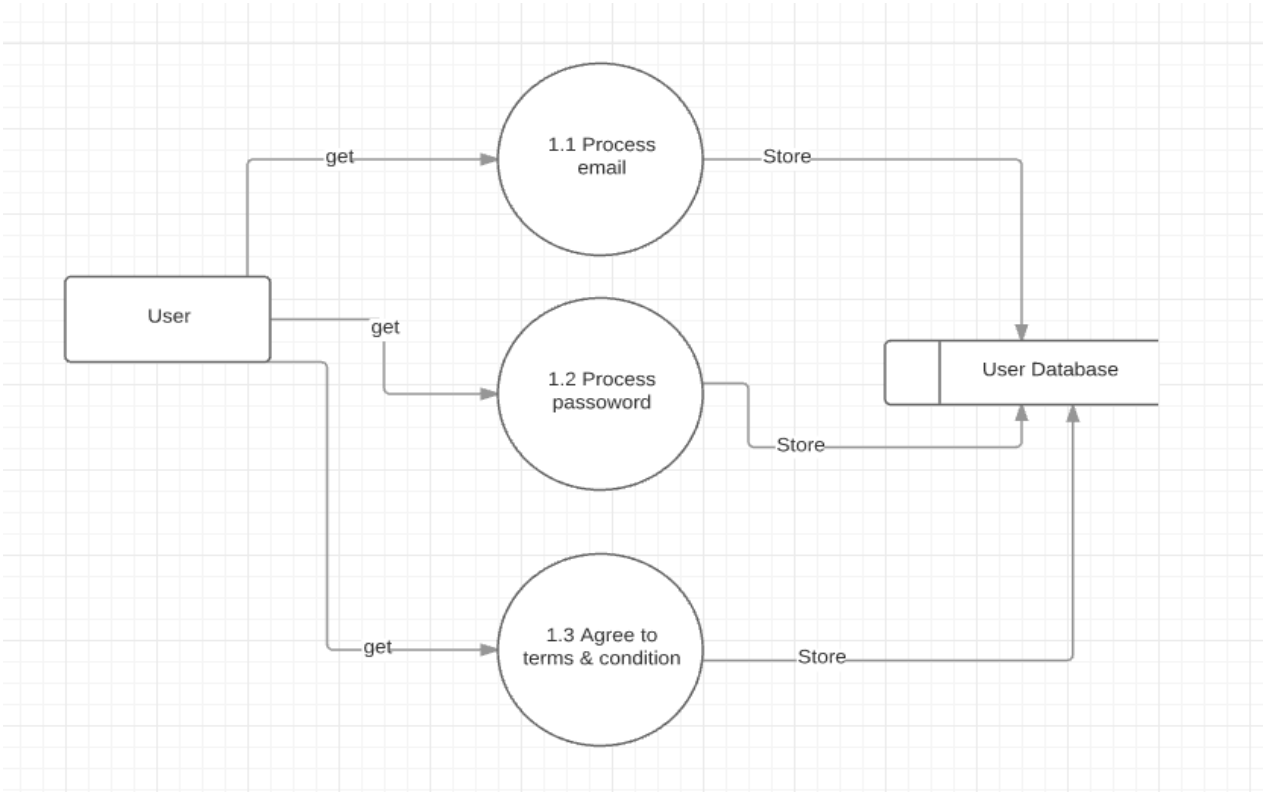

Fig 27: DFD 1

2. Login

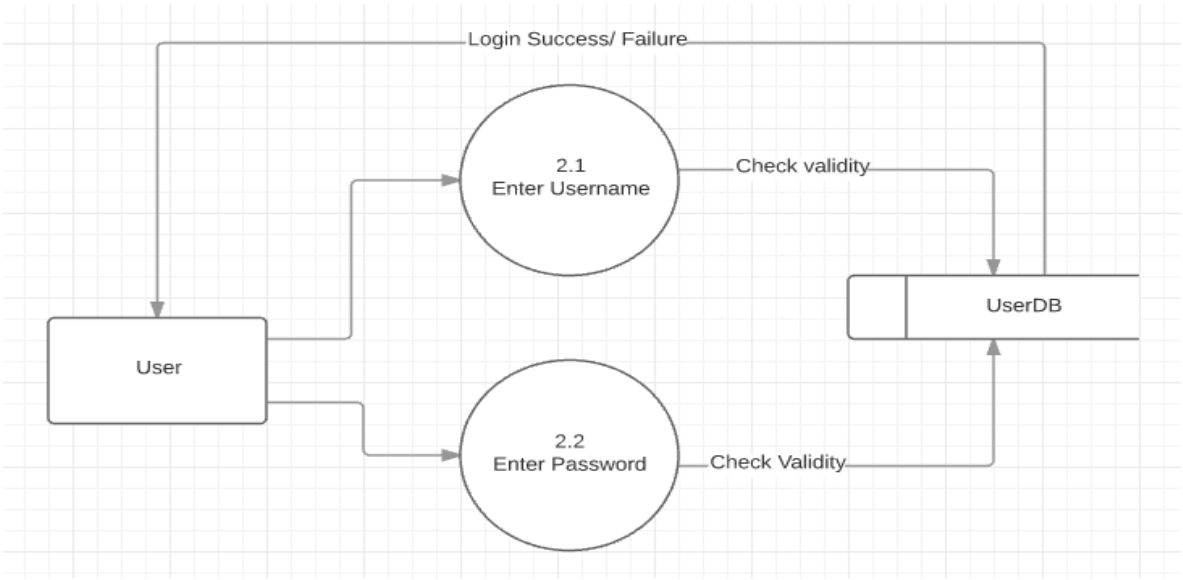

Fig 28: DFD 2

3. Navigator

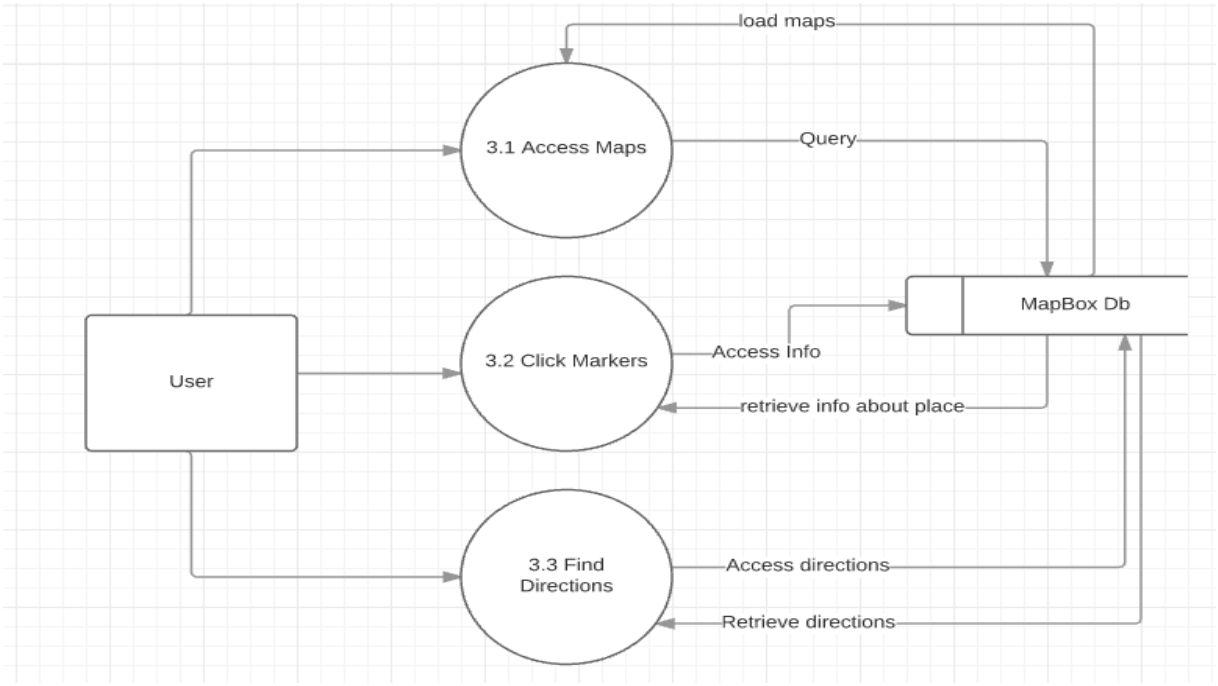

Fig 29: DFD 3

3.2. Click Markers

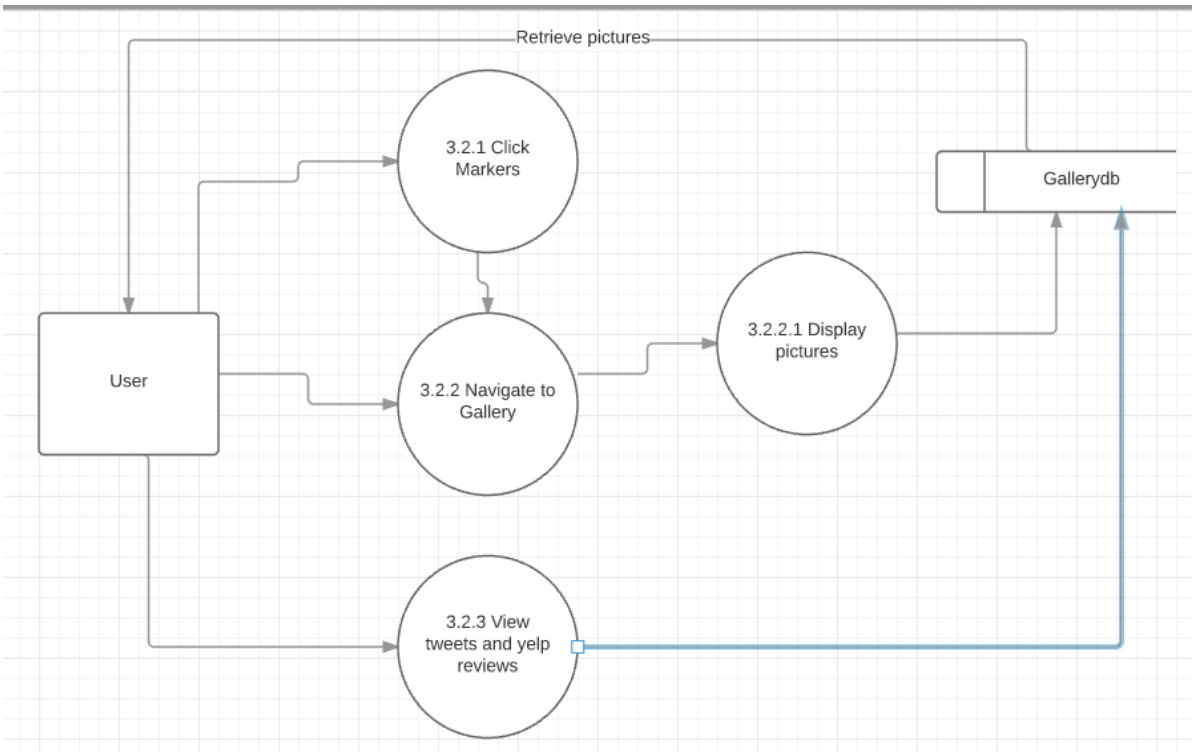

Fig 30: DFD 3.2

4. Find Friends

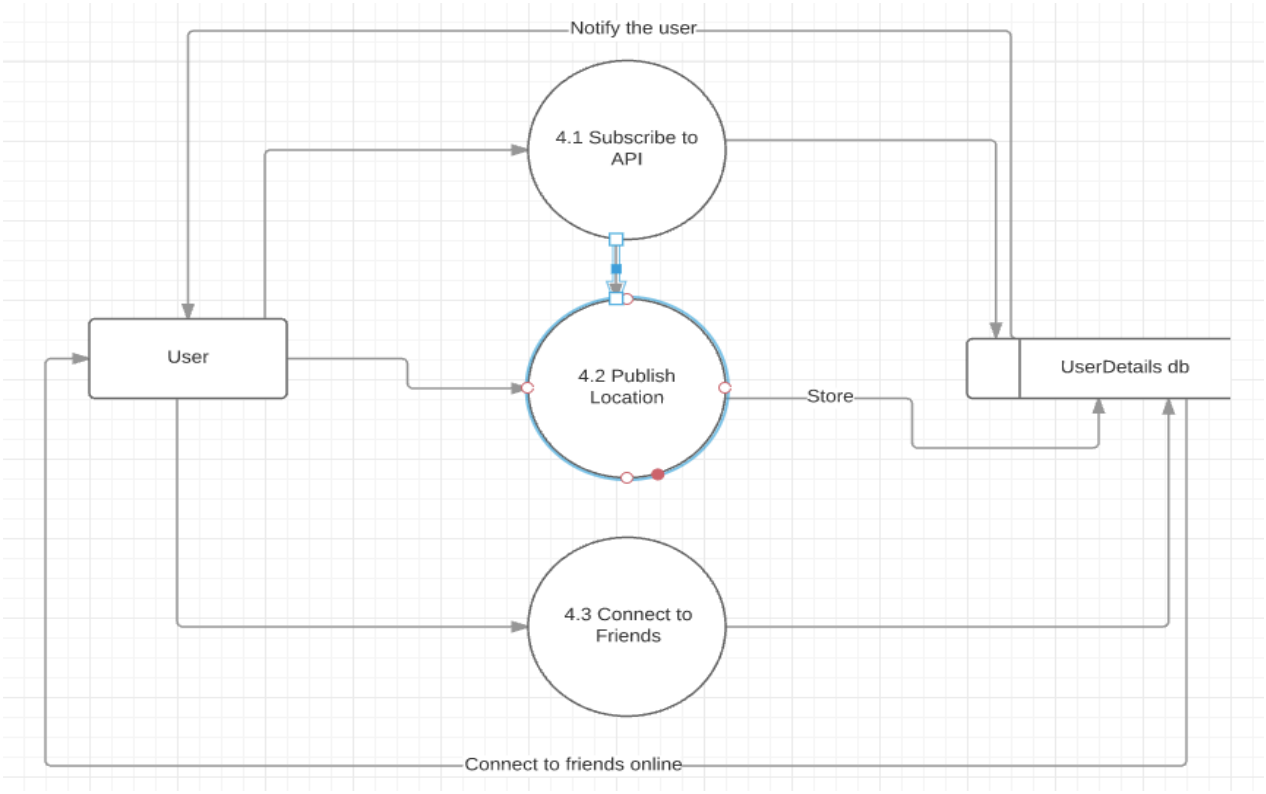

Fig 31: DFD 4

5. Health Statistics

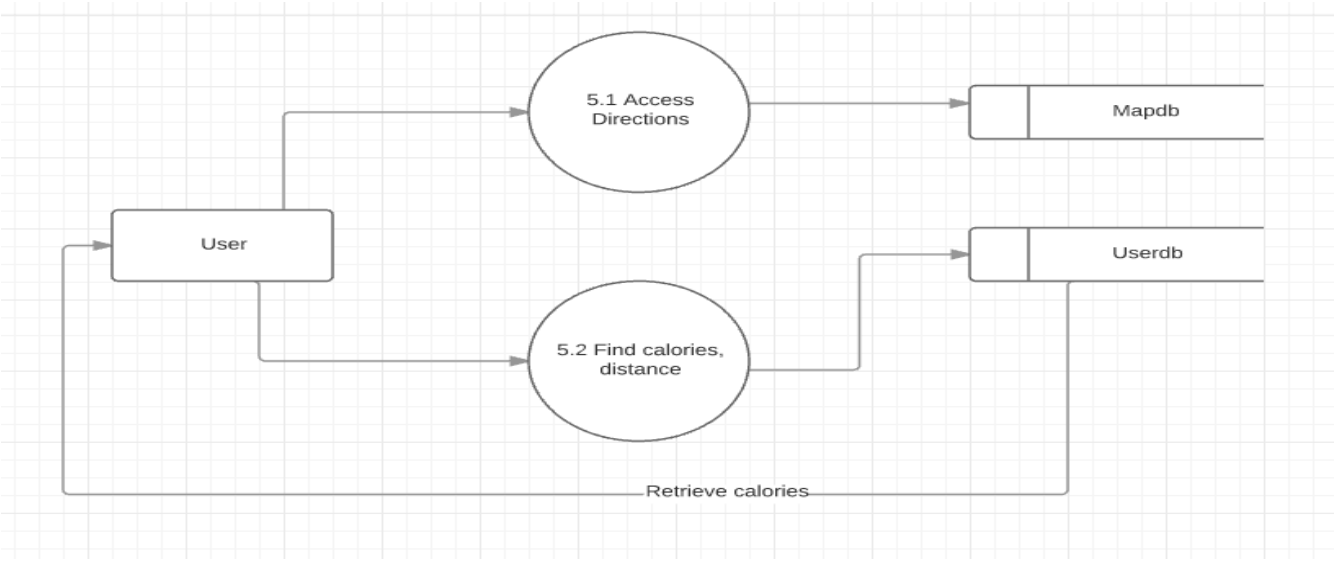

Fig 32: DFD 5

6. Dashboard

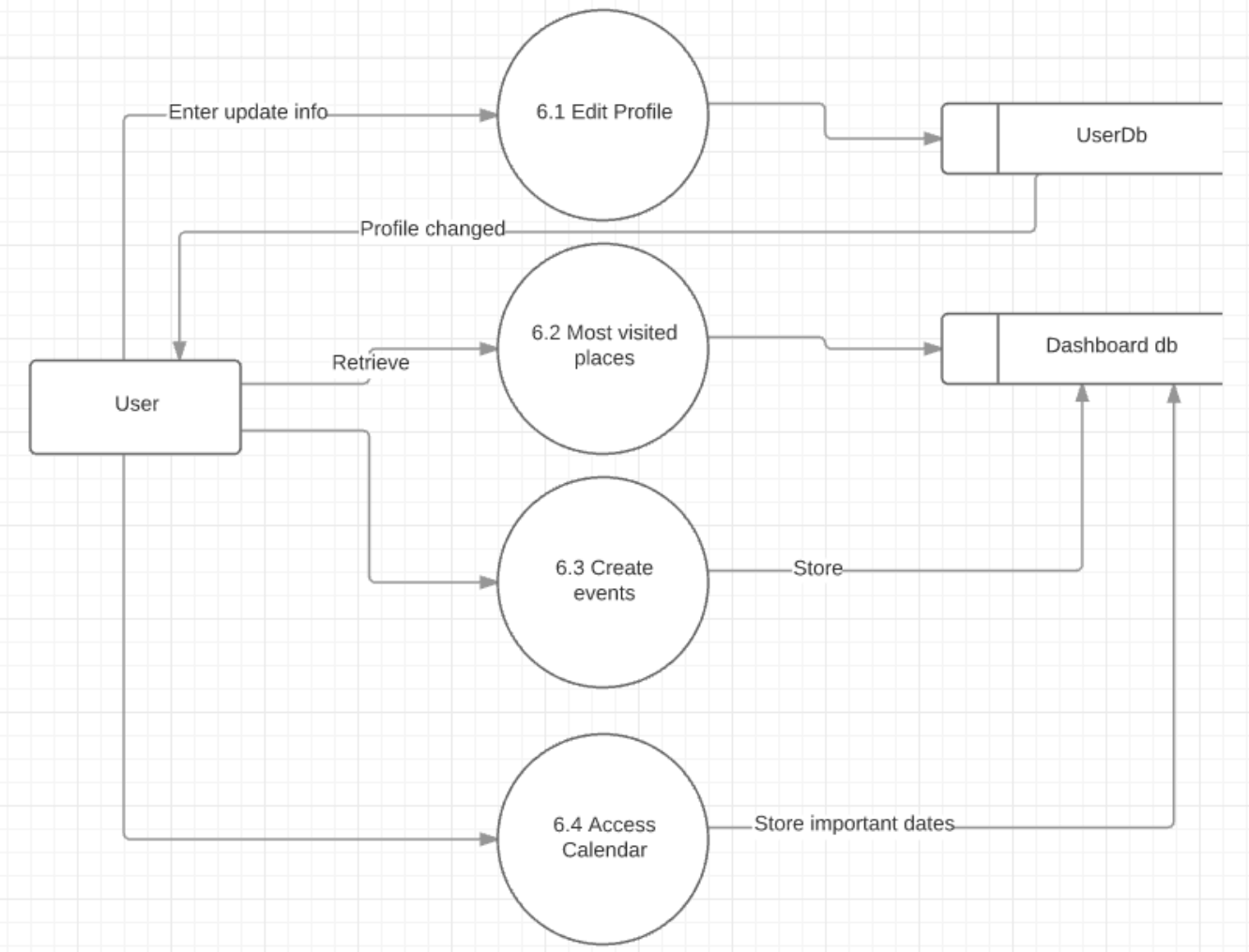

Fig 33: DFD 6

3.3 Sequence/ Workflow

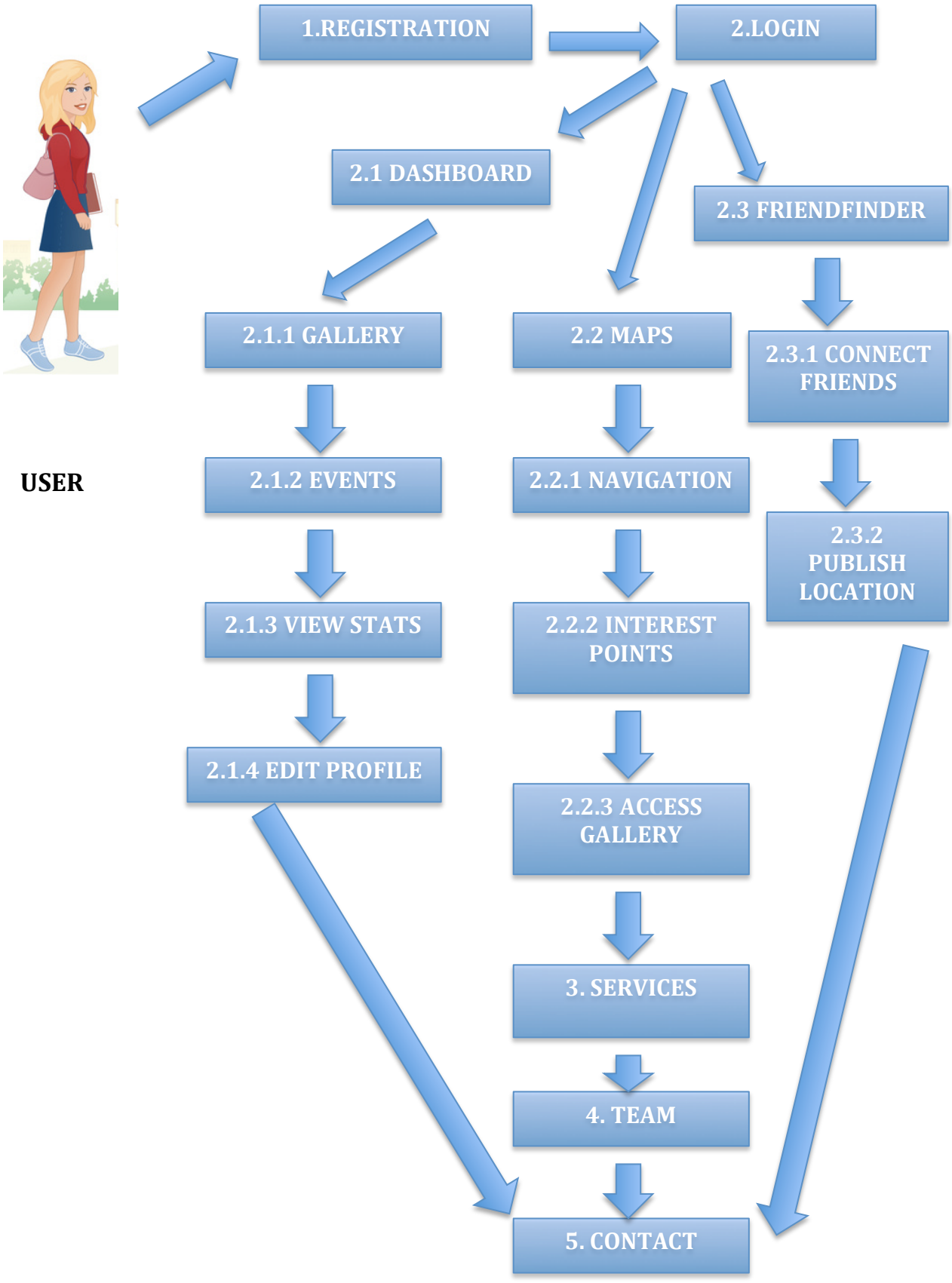

Fig 34: Workflow of SpartaMaps

3.4 Client Side Design

In this chapter we will be discussing the client side design of the SpartaMap web application. There are number of technologies used at the Client side which we will elaborate in this chapter.

MapBox API

The mapbox web services allow for easy map designing using mapbox tools and services. The mapbox webservices use the mapbox.js, mapboxgl.js and leaflet.js to create markers and polygons to mark the different places on the map. There are four techniques for accessing mapbox services – cURL, Mapbox CLI, Mapbox Javascript SDK, Mapbox Python SDK. Basically, it is a opensource tool to create interactive maps and has wide range of formats such as GeoJSON, JSON, JSONP, KML and REST to embed the maps on the HTML and the map can be shared with a wider audience. All the maps created with mapbox are compatible with Android and IOS devices.

For the Spartamaps, we used different methods of the MapBox API such as marker clustering, geocoding, custom tooltips, zooming, panning to make our maps look interactive in nature.

Fig 35: MapBox API

PubNub API

PubNub api establishes the publish-subscribe design pattern for streaming live data and signaling for device which establish and maintain persistent socket connections to any device and send data to audiences around the globe. The pubnub contains the components such as API keys, Messages and Channels. For a PubNub stream to start, you have to establish API keys. A minimum of a publish/subscribe key is needed to establish the connection. If a client

has only to subscribe and not subscribe, a publish key is just enough. If a client has to publish, you need to have subscribe and publish key. The pubnub payloads can contain JSON data including Booleans, Numbers, and Array. The channels can be unicast or multicast but there is a limit on the messages being sent.

In SpartaMaps, we used the PubNub API for the find friend feature. The SJSU people registered to the application has to send their location using the publish and subscribe method for the other person to know the number of persons around him and connect with them.

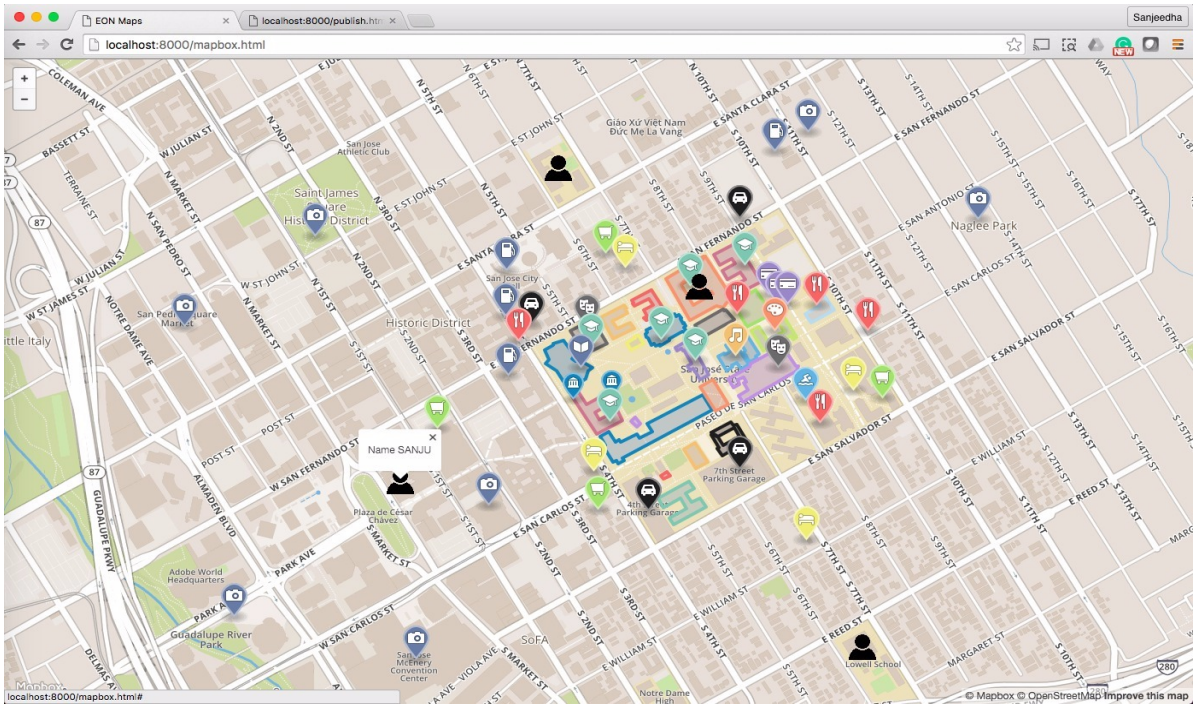

Fig 36: PubNub API

Geolocation API

The geolocation API of the Google maps returns the location and accuracy radius based on the user's latitude and longitude positions. It takes the information from the GPS location given by the mobile for a particular person. Geolocation requests are sent using POST and it requires an API key to be specified. The key identifies the application. The Geolocation API is used with MapBox API in SpartaMaps to track the user's current position. The navigator.geolocation method is used to get the value of the latitude and longitude. The showPosition() outputs the latitude and longitude coordinates of a particular device. WatchPosition() method and ClearPosition() methods are used to keep the trail of the user and clear the trail.

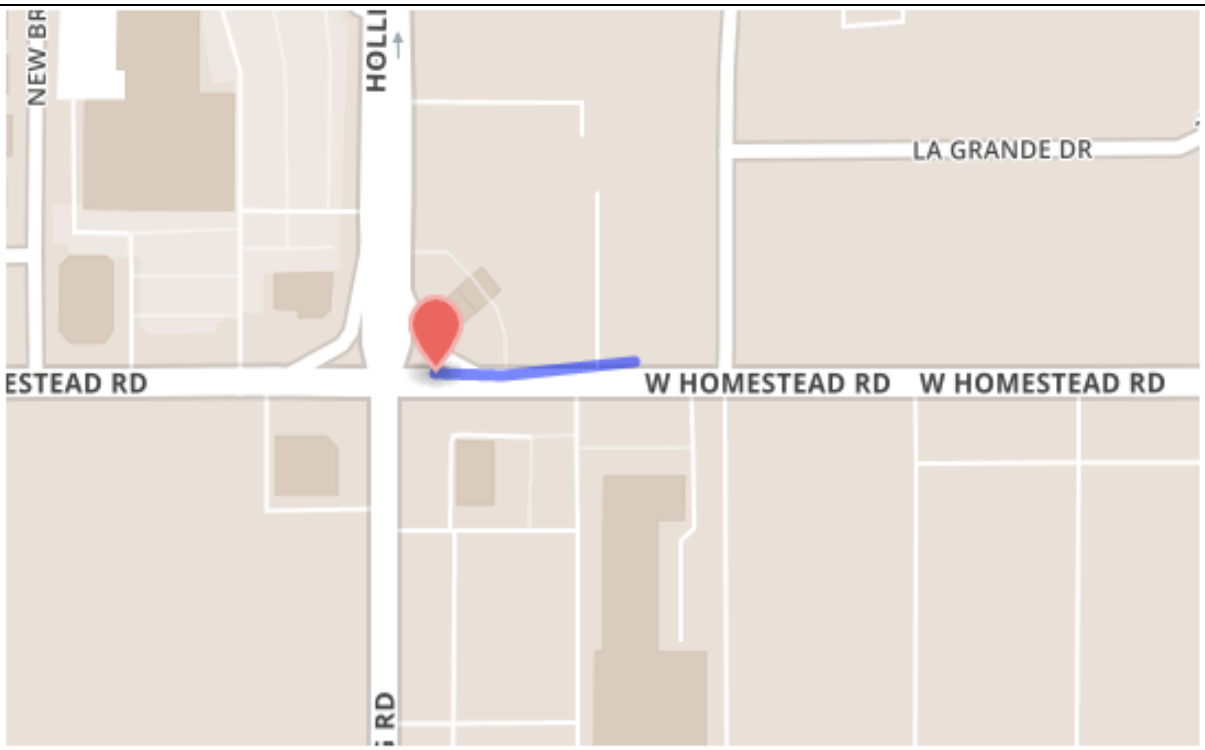

Fig 37: Geolocation API

JavaScript Libraries

A JavaScript library is a library of pre-written javascript code, which allows for easier modeling of javascript applications, especially AJAX and other web-centric applications. These libraries are dynamic and easier means of the programing interactivity and basically reduce the lines of code.

In SpartaMaps, we have used the javascript libraries such as mapbox.js, mapboxGL.js for navigator feature, d3.js for graph display and the like.

Twitter API

The twitter API connects the application with the global conversations happening in Twitter. It is easy to embed a piece of code to enable twitter feeds onto the page. The streaming API's in twitter enable live data to be updated and stay in sync with the user profile updates.

In Spartamaps Application, we have embed twitter feeds codes with the respective hash tags to display the top tweets about the application and also the buildings inside SJSU.

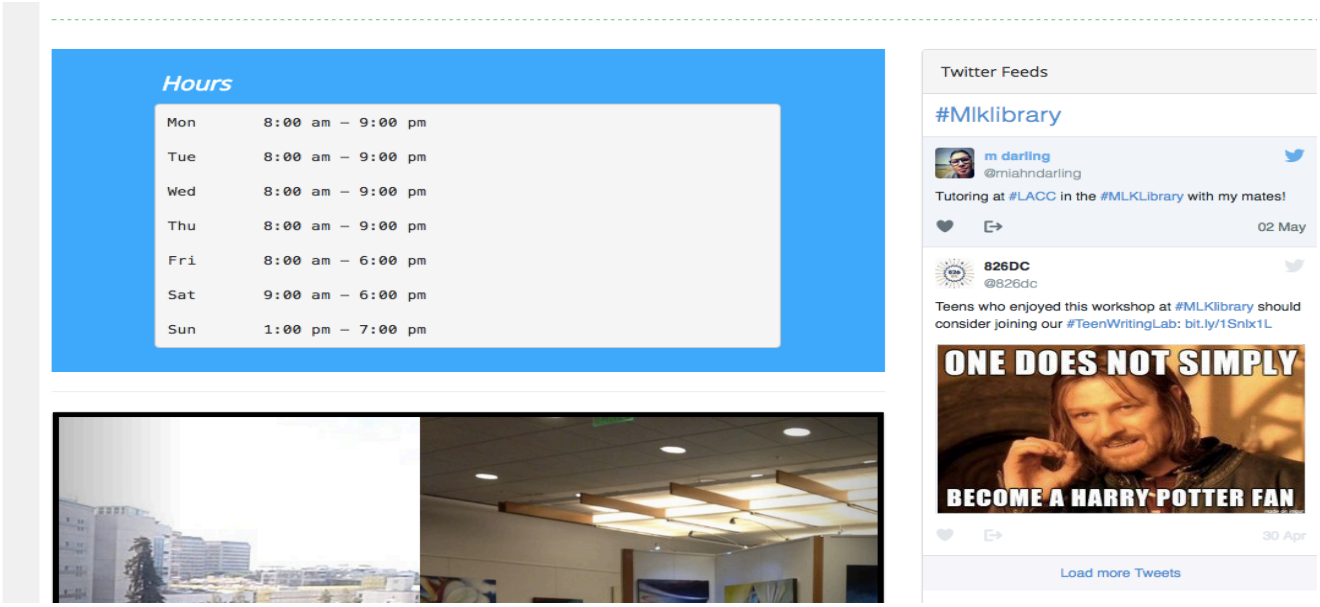

Fig 38: Twitter API

Yelp Embeddable Widget

Any business page on yelp can be embed in a website using the embeddable yelp widget. All starts with entering a BusinessID in the text field intended to do so. The BusinessID is available on the yelp profile page and then by copying the text in the URL the yelp widget become available.

In SpartaMaps, we implemented the feature for some buildings so that the user can see the yelp reviews about the particular place or the restaurant and decide to go to it.

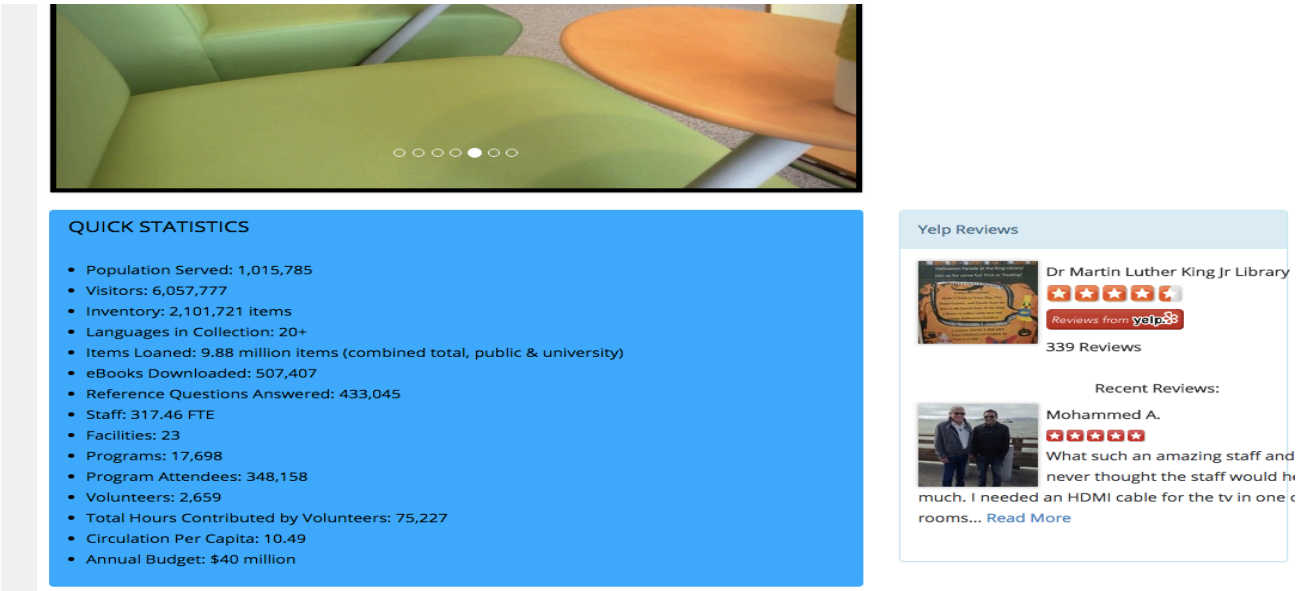

Fig 39: Yelp Widget

FourSquare API

FourSquare widget in SpartaMaps is also similar to other social networking widgets. Each marker is marked foursquare and when clicked on the marker can direct to the foursquare page of the building.

Fig 40: Foursquare API

HTML

HTML is an abbreviation for Hyper text Markup Language. It is a core technology generally used to create web pages as well as for the mobile client. HTML describes the web semantics of the web page for client side.

For the “SpartaMaps” The HTML is the base language for development of almost each page. With starting page of Home page. There are many other libraries and Language snippets like CSS, JavaScript. That are supporting the HTML to work. We have used vast number of tags that HTML5 provides through out the project.

Bootstrap:

Bootstrap is a free and open-source front-end library for creating websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications.

Bootstrap is a front-end web framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

Bootstrap is the second most-starred project on GitHub, with over 95K stars and more than 40K forks.

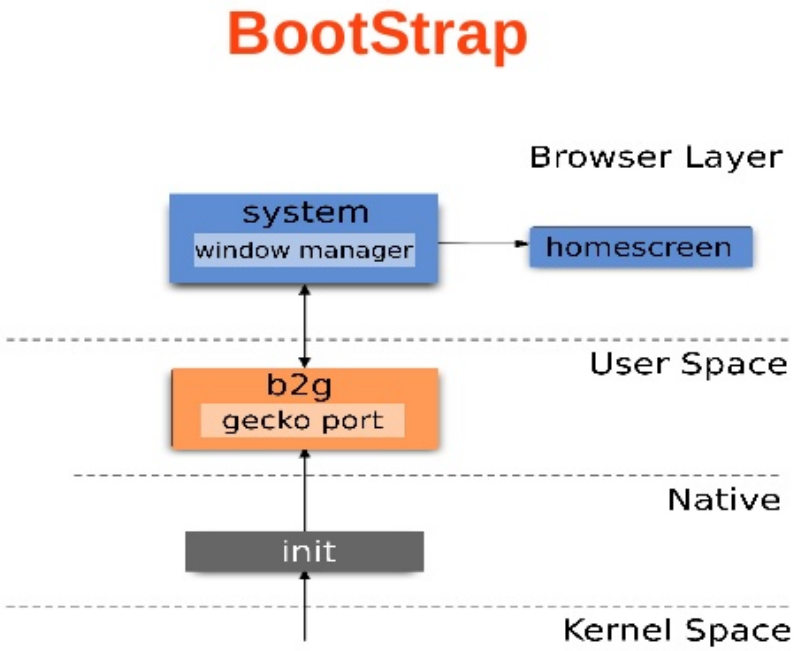

Fig 41: Bootstrap Architecture

For the SpartaMaps We have used bootstrap libraries on top of the HTML base. The main purpose we had to use this library is to make the web page responsive and mobile first. The library leverages the grid structure to divide the screen in 12 columns. This gives the great advantage to distribute the content evenly across the web page. Another very big advantage of bootstrap is once you load the page unless there is a change the bootstrap will load faster next time as it caches some data on to the client side which helps client to have more quick service than normal HTML page would have.

CSS

An abbreviation for the Cascaded style sheet. This technology is mainly used to give a style to the HTML content. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone

technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

In SpartaMaps we are using CSS to style contents and make it more beautiful and appealing for the customer. The CSS is used in almost every development stage of SpartaMaps.

Owl-Carousal

The Owl Carousel Template for FooGallery allows you to create a simple, elegant, and responsive carousel of your gallery images. This can include from 1 to 6 images at a time. It supports the same default styles and hover effects as the FooGallery Responsive template but also has previous/next buttons and pagination bullets. Just like the Responsive and Masonry templates, Owl Carousel supports on-the-fly Thumbnail sizing (with or without cropping), a default "Thumbnail Link", and your choice of Lightbox.

Thumbnail Size and Styles - This section determines how the thumbnail images will appear in your carousel. Here you can dynamically change the sizing of the images, set the margin between images, the border and hover styles and more.

Carousel Stage - This is one of the most distinctive features of Owl Carousel. Here you can set how many images you see at a time in your carousel. Most "sliders" just show one image at a time. But with Owl, you can show how ever many you can comfortably fit inside the "stage".

Carousel Navigation - This section is for determining how your users will navigate through your images. For example, "Navigation" represents the "Prev/Next" buttons you can use to click through one image at a time (NOTE: Mobile users can always swipe through the carousel as well). But "Pagination" gives you "dots" to represent a set of images based on how many you put in your "Stage". This section also has settings for autoplay, and slide animations.

JQuery

JQuery depends on DOM. The DOM is a tree-structure representation of all the elements of a Web page and jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. JQuery, at its core, is a DOM (Document Object Model) manipulation library. For

example, jQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag), changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

JQuery also provides a beautiful feature support for event handling that goes beyond basic DOM element selection and manipulation. The callback function & event assignment and the event definition are done in a one step in a one location in the code. jQuery also aims to leverage some other highly used JavaScript functionality.

For The Sparta Maps we used JQuery in the finding the maps for the customer. The jQuery is used to make an AJAX request to foursquare to load markers data. Along with this prominent use, We have also used JQuery in other pages to help and retrieve data from the application server.

JavaScript

The HTML DOM is a standard object model and programming interface for HTML. With the HTML DOM, JavaScript is able to access and change, get, add or delete the elements of an HTML Document. The W3C Document Object Model (DOM) is a platform and language-independent interface that allows any program or a script to dynamically access and update the structure, content and overall layout or style of the document. The programming interface is the collection of all the properties and the methods of each object.

- A property can be defined as a value that we can get or set as in changing the content of an HTML element.
- A method can be defined as an action that we can perform or do on an HTML element.

Following things can be done by JavaScript using HTML DOM are:-

- It defines all HTML elements as objects.
- It defines all the events for all the HTML elements.
- It can change all the HTML attributes present in the page.
- It can change all the layout and styling of the page.
- It can remove all the existing elements and attributes.

To access an HTML element, widely used way to do it is finding the id of any element by using method getElementById. [1]

In our project, we have used this method to track the navigation of a user by finding the element with id=”navigation”.

E.g.: *document.getElementById (“navigation”).innerHTML*

cURL - Client URL Library

It is a library created for PHP, that allows us to connect to different types of servers and also communicate with them to get data via different protocols. It supports the following protocols:

- Http, Https
- Ftp
- Gopher
- Telnet
- Dict
- File
- ldap

It also supports:
HTTPS certificates, HTTP POST, HTTP PUT, HTTP form based upload, FTP uploading , Proxies, cookies, and user & password authentication.

PHP

PHP stands for Hypertext Preprocessor. It is a server side scripting language designed for web development, which allows web developers to create dynamic content that can interact with databases as well. PHP is basically used for developing web based software applications, and is also used as a general-purpose programming language. It is a widely used, free and efficient language which can be embedded into HTML code. It can also be used in combination with different web template systems, web content management systems and web frameworks. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

We have used PHP in our project to interact with databases, and also to maintain sessions in order to make certain data accessible across various pages of our website. A session creates a temporary directory on the server where the registered session variables and their values are stored. When the user closes the browser or leaves the website, the server terminates the session after a predetermined period of time (generally 30 mins). Session is started by making a call to **session_start()** function on the beginning of the page which checks if there is an existing session. If not, it starts one. The session variables get stored in an associative array called $_SESSION, and can be accessed till the particular session is active. Session is destroyed by making a call to the function session_destroy(). This function doesn't need an argument and a call to this can destroy all the existing session variables for that session. To delete only one particular session variable, unset() function is used.

Google Maps API

The google maps API offers rich map visualization with accurate directions and Street View through Google Street View Image API. This API allows embedding a static panorama or a thumbnail into the webpage using javascript. Google Street View API uses a specific panorama ID called pano. The output size of the image is specified using the size parameter. The fov describes the field of view of the image. The pitch attribute specifies the agle of the camera. The API also contains the sensor parameter to indicate whether the application uses the sensor to determine the user's location.

Fig 42: panoramic view of the MLK Library

Local Storage

With local storage, the web applications can store the data locally within the browser. This local storage help the application to function even when there is no network connection. The page won't display an offline status. Instead of that it would display the home page with offline support. SpartaMaps contains local storage mechanism and functions even when the application goes offline.

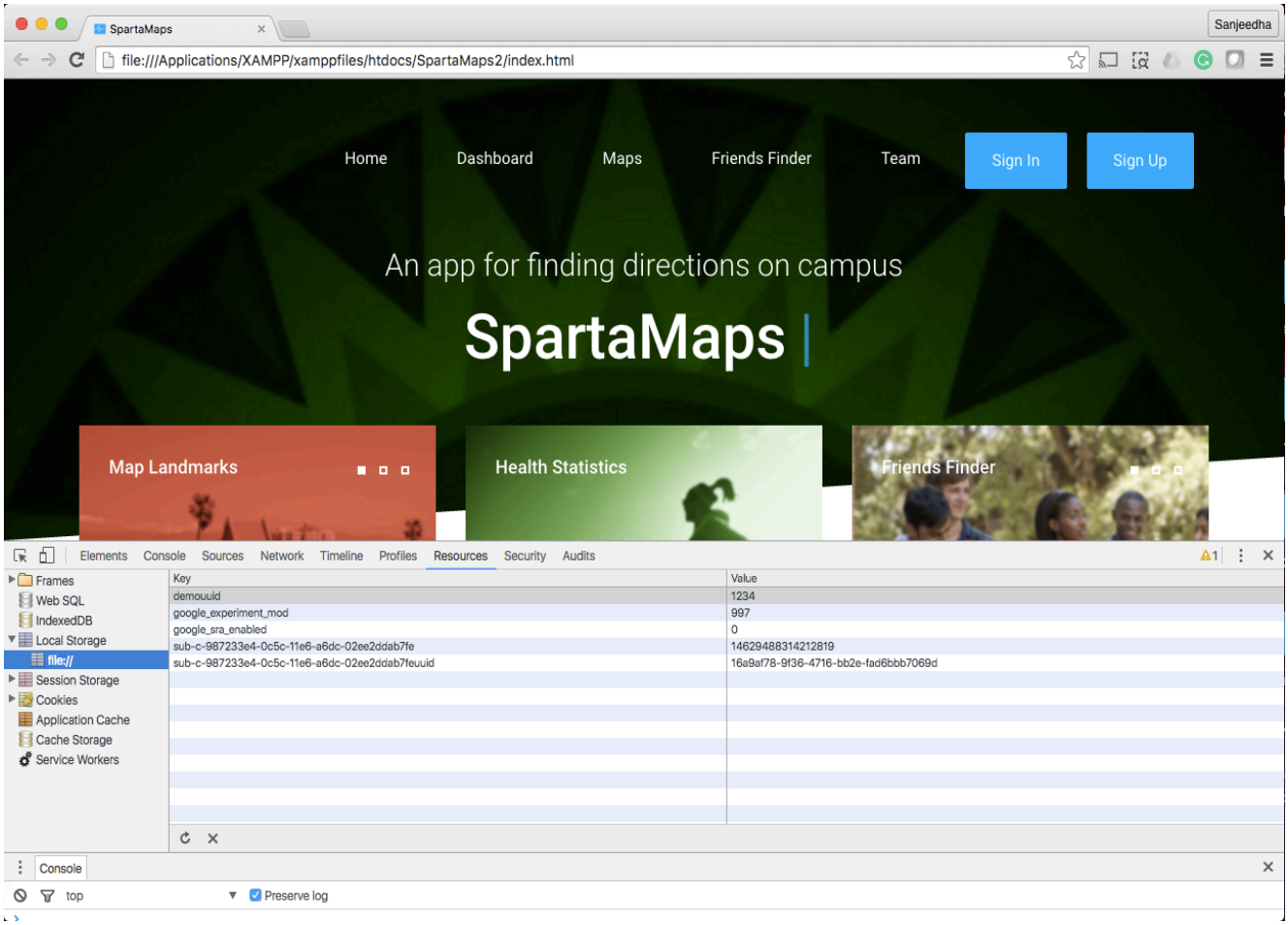

Fig 43: SpartaMaps with Local Storage

Whole front-Side Design

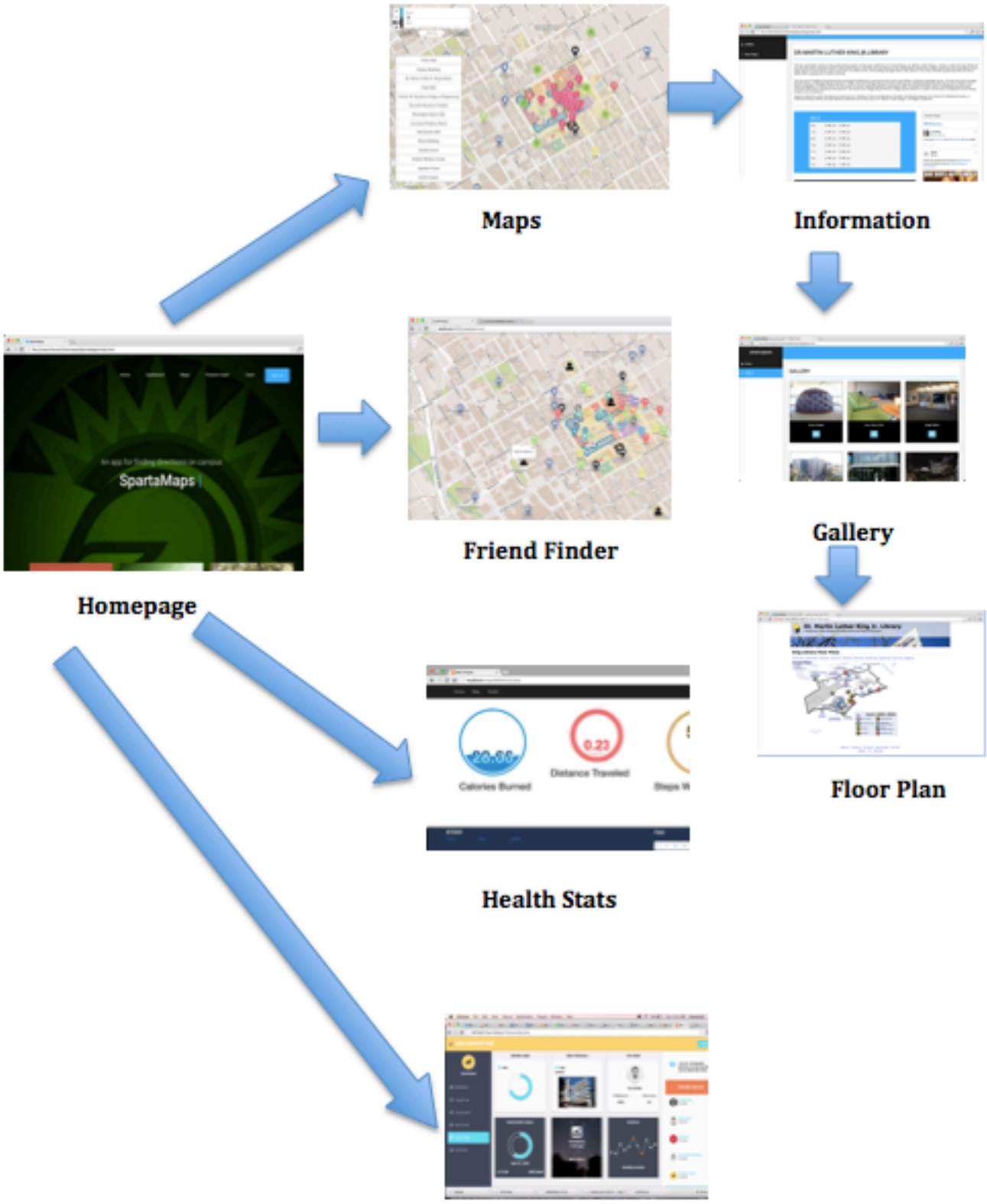

Fig 44: Whole Front-end UI

3.5 Server Side Design (Interfaces & RESTful API)

REST - Representational state transfer is a software architectural style of the web.
By using REST APIs, we can induce architectural properties such as

- Performance
- Scalability
- Simplicity
- Modifiability
- Visibility
- Portability
- Reliability

HTTP Methods

RESTful Web services explicitly use HTTP methods that follows the protocol defined by RFC 2616. REST developers use HTTP methods explicitly and also maintain consistency with protocol definitions. REST design principle has a one-to-one mapping between HTTP methods and create, read, update, and delete (CRUD) operations:
- Creating a resource on the server - POST
- Retrieving resource - GET
- Changing the state of a resource or updating it - PUT
- Deleting resource - DELETE

Stateless: The absence of state on the server improves the performance and simplifies the design as it removes the need to synchronize session data with external applications.

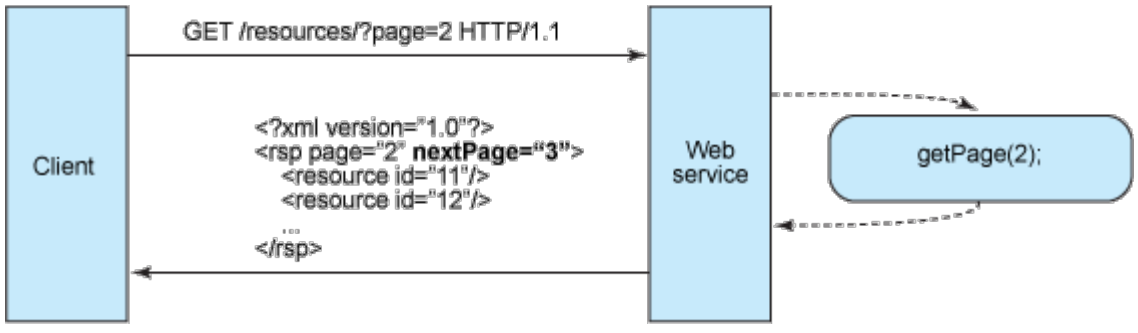

Fig 45: HTTP Methods

REST APIs and Services provided by our Sparta Maps:

http://www.meeramali.com/SpartaMaps/

- This is our home page from where you can Sign Up, Sign In, and accesses the menu to utilize other services provided by us.

http://www.meeramali.com/SpartaMaps/Theme/index.php

- This is the Dashboard service provided for the particular user. It is his/her dashboard , where a lot of details and visualizations are provided. The user can navigate to other services from here.

http://www.meeramali.com/SpartaMaps/Maps.php

- This provides the main service of interactive navigation and finding directions inside the San Jose State University campus.

http://www.meeramali.com/SpartaMaps/friends.php

- This provides the Friends Finder services for finding a friend on campus by knowing his/her location with PubNub integration.

http://www.meeramali.com/SpartaMaps/Theme/HealthStats/index.php

- This service provides the health data information like calories burnt etc. for the user according to his/her travel statistics (visits completed etc.)

http://www.meeramali.com/SpartaMaps/Maps.php/{destination}

- {destination} - According to the destination inside the San Jose State University Campus, the details about the location, with various features, and a virtual tour of the routes, and panoramic views etc. are provided .

Chapter 4

Application Development

4.1 HTML5 Features

Modern browsers are more advanced and provide advanced features. HTML5 was introduced to meet the modern day needs. It supports backward compatibility, and the base to it is that no major changes were made to the programming model while making a transition from HTML to HTML5. It mainly aims at improving the support for multimedia, and is supported by web browsers like Google Chrome 10.0, Firefox 4.0, Internet Explorer 8.0 and provides a simple, better looking and easy to use interface for application users. Following are some of the HTML5 features we have used in our project:

1. **DOCTYPE:**
   The <!DOCTYPE> declaration is the first thing in the HTML file. It tells the browser about the Html version of the page.

Fig 46: SpartaMaps Doctype

2. **Email:**
   It has been a task to perform validations over input fields using Javascript. HTML5 has solved this problem for us, the validations is performed right when the user inputs the value instead of checking it through javascript when the user submits the form.

```
</section>
<div class="modal fade" id="modal1" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content modal-popup">
            <a href="#" class="close-link"><i class="icon_close_alt2"></i></a>
            <h3 class="white">Sign Up</h3>
            <form action="" class="popup-form">
                <input type="text" class="form-control form-white" placeholder="Full Name">
                <input type="text" class="form-control form-white" placeholder="Email Address">
```

Fig 47: SpartaMaps Email field

3. **Password**

A lot of websites require registration these days. HTML5 provides input type as password, which doesn't let the input being displayed as and when the user types. You can also define a pattern attribute to it where you provide a regular expression to restrict the user's input to demand a strong password for security reasons.

4. **Required**

Almost all websites these days either want you to register with them or give them some kind of information about you. This is generally achieved using forms. There are places where the information asked is mandatory to fill. Instead of checking and applying validations through javascript codes to check if the user has provided an input, HTML5 gave a feature called required, which gives an error if the input field is blank.

Fig 48: SpartaMaps Password & Required tags

5. **Placeholders**

There are some places in the form where in a user can give his input in different formats, or maybe requires a guideline over what you should fill up. Also, to make the code compact and avoid the use of labels, a feature called placeholders is provided by HTML5. Eg: if there is an input field and you give it a placeholder value as 'Enter email', it will show a grey text on input email as enter email, which disappears as soon as you start to give an input.

```
</section>
<div class="modal fade" id="modal1" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content modal-popup">
            <a href="#" class="close-link"><i class="icon_close_alt2"></i></a>
            <h3 class="white">Sign Up</h3>
            <form action="" class="popup-form">
                <input type="text" class="form-control form-white" placeholder="Full Name">
                <input type="text" class="form-control form-white" placeholder="Email Address">
```

Fig 49: SpartaMaps Placeholder tag

6. **Nav:**

HTML5 has introduced a special tag called nav, which defines a group of links for navigation bar. It is only the links for navigation that should be added to this tag. This makes it easy for a developer to track where the element navbar is defined as it is displayed as a separate tag.

Fig 50: SpartaMaps Nav tag

7. **Figure**

The figure element is introduced in HTML5 to include self-contained content like illustrations, diagrams, and photos and code listings. This element is related to the main flow, so the position for it is independent. Which means if it is removed, it will not affect the flow of the document in any manner.

Fig 51: SpartaMaps Figure tag

8. **Header**

Header element is also newly introduced in HTML5. It is set up to define intoductury content of the HTML page. It generaly contains the HTML mark-up for navigation links or navigation-bar. However, a page can have multiple header tags also. Typically header tag has the website logo, home, navigation links etc

9. **Footer**

Footer tag of HTML5 defines the footer part of the document. This generally contains a sitemap, authorship information, contact information, other relevant links, copyright information, social links to follow, and other related documents. As header elements, even footer elements can be multiple inside a document.

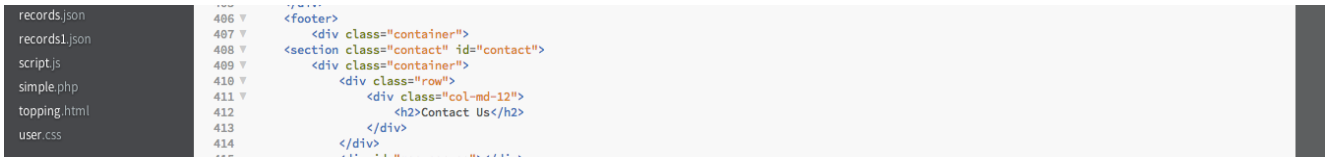

Fig 52: SpartaMaps Footer tag

10. Section

The section tag in HTML5 has been introduced to create different sections in the document to separate contents. Each <section> tag defines sections in a document, which can contain content such as chapters, headers, footers, or any other sections of the document. It is supported in all the popular browsers.

Fig 53: SpartaMaps Section tag

4.2 Design Patterns

We have implemented the MVC (Model View Control) pattern; Topic based publish-subscribe pattern, Client-Server pattern, AJAX pattern as part of our project.

MVC (Model View Controller) design pattern

MVC is a software architecture or the structure of the framework that separates the application logic or business logic from the rest of the client or end-user interface. It does this by dividing the application into three main sections: the model, the view, and the controller.

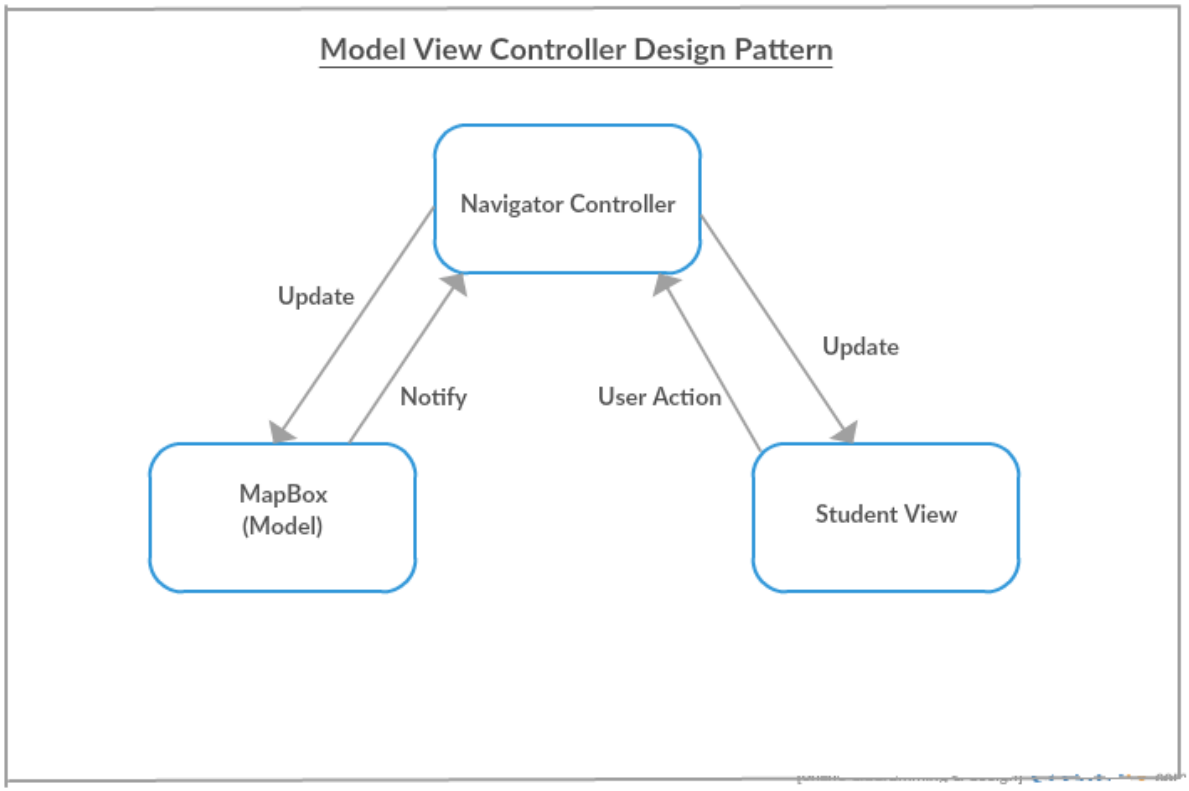

Fig 54: Model View Controller Design pattern

The model oversees the essential features and information of the application. It can respond to demands for data, respond for changing the state of the information and even notify the observer when any information changes on the occurrence of any event. The information can be present in any form like any data structure or any storage system. Basically, the model in MVC design pattern can be termed as a data and data administration of the application.

The view provides the client interface component of the application. It will extract the desired information from the model of the application and provides it into a well-defined structure. This structure is needed for designing the user interface part of the application.

The controller is responsible for receiving input from the end-user and making calls to model objects and the view of the application to perform suitable activities.

We have used MapBox as a Model in our application which is providing useful information of the SJSU Maps to SJSU students. SJSU students are considered as view of the application. Navigator controller is used to help them to search another students which are currently displaying as Online on the map.

Topic based Publish-Subscribe design pattern

In a Topic based publish-subscribe design pattern, Sender applications broadcast each message with a topic as a tag without referencing any particular group of receivers. The messaging system then forwards the same message to all the desired applications that have asked to receive messages on that topic.

Publishers of the message are concerned only with the sending of the message with their desired topic irrespective of the task of providing service to the concerned. Messaging system is responsible for sending message to the desired audience and contains information of all the subscribers and publishers.

Publishers and subscribers can communicate with each other only via messages without getting noticed of all the available subscribers and publishers respectively. Both parties are loosely coupled with each other such that publishers are not aware how many subscribers are there for their messages and also subscribers do not have any knowledge of the no. of publishers.

Subscriber applications can subscribe in more than one area of interest or topics and can receive only those messages, which matches with their area of interest.

In Spartamaps, Publisher will be the user who wants to find friends to accompany him/her to the desired location and sends message which contains the information of the current location of the user as well the destination information. Message Broker will receive message from the publisher and broadcast this message to all the intended subscribers and in return it will let the publisher know about the response from any of the subscribed entity.

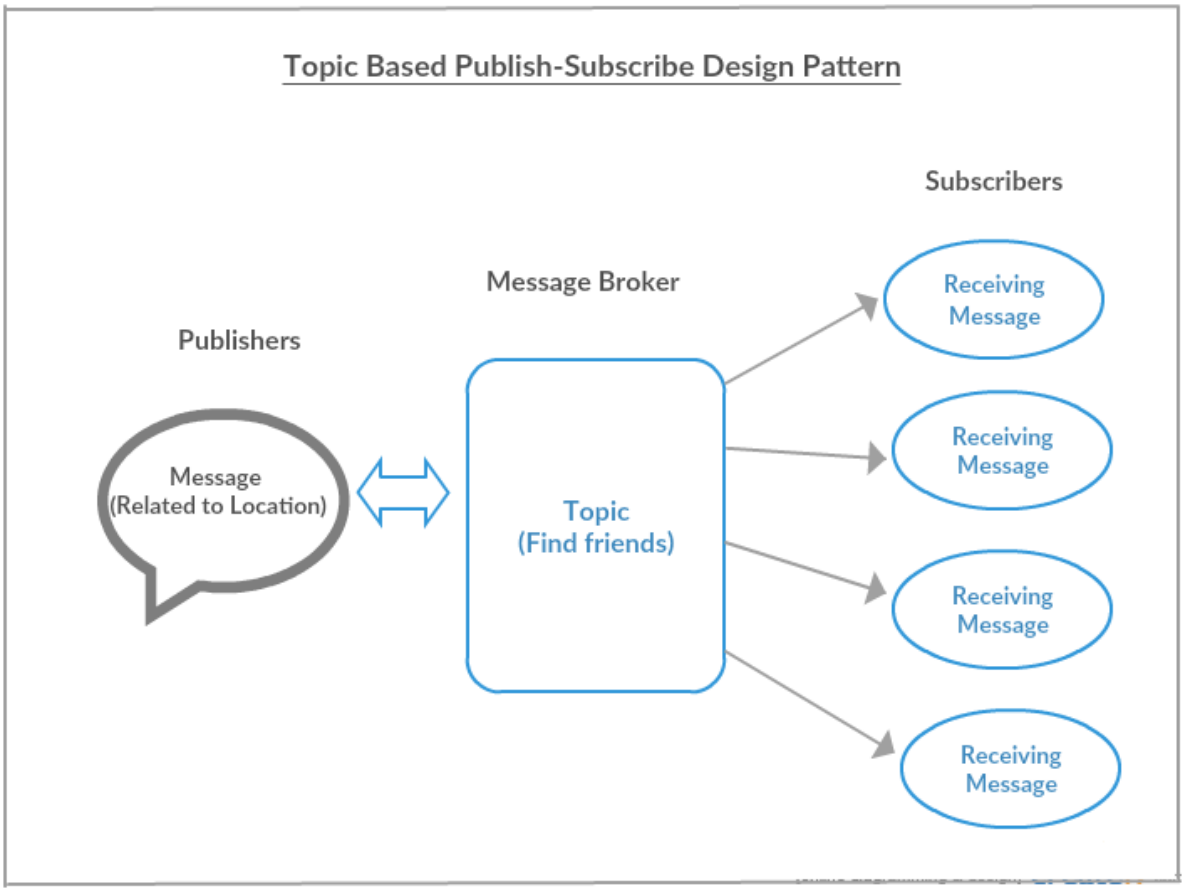

Fig 55: Topic based Publisher-Subscriber Design pattern

Client-server design pattern

Client-server system is responsible for providing access to a central application from one or more remote clients. The server should be designed in such a way that it should be able to dynamically accept and provide service to any number of incoming connections without causing any high-level of interruption. The server should also properly manage all the incoming requests from the clients by continuously monitoring each of the client and thus should be able to serve each of them individually.

The client should be able to connects, disconnects and again reconnects to the server n no. of times. The client should be able to receive the data form the server.
This pattern involves check connection service which is used to check the status of each connection as it is serviced. If an error occurs, the service should be able to take desired action, either providing the recovering options or by providing the appropriate error message.

We have used multi-client server overview in our project which can be well understood by the following figure.

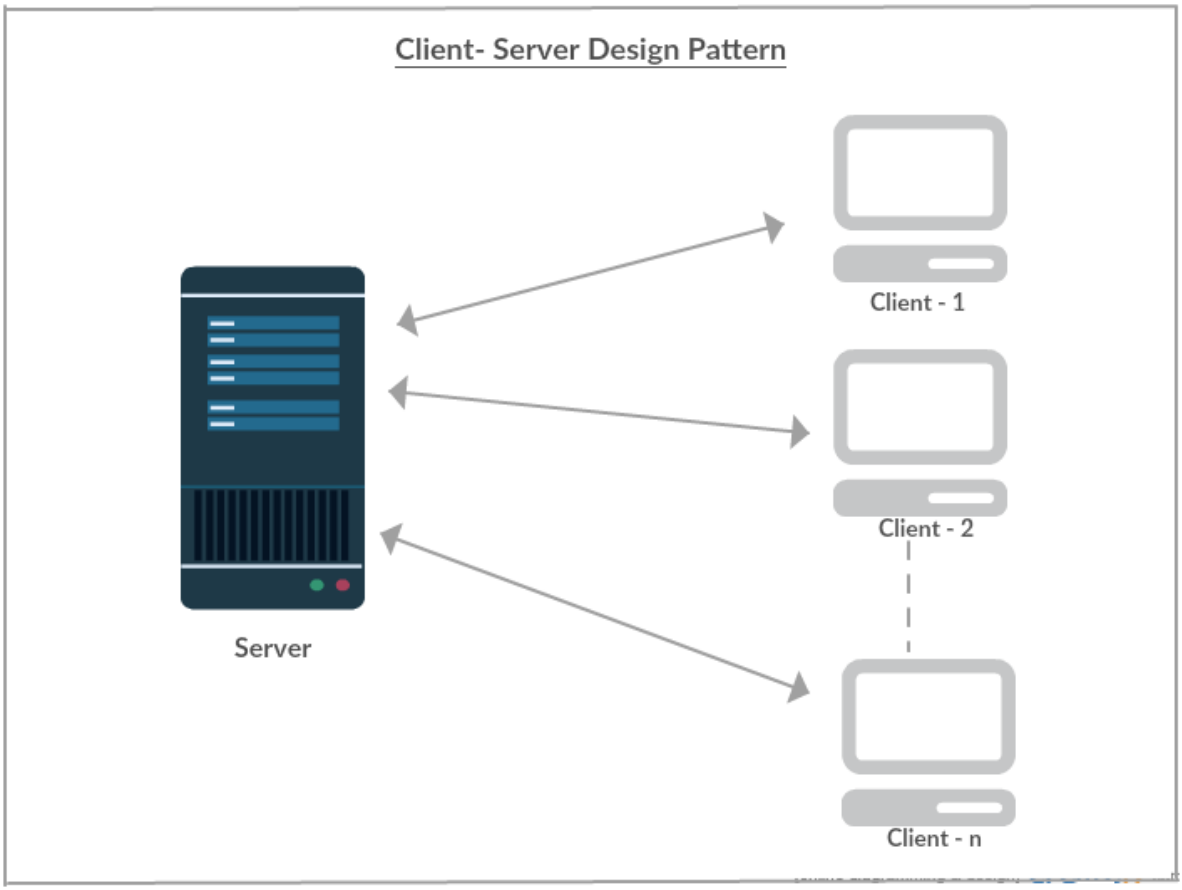

Fig 56: Client-Server Design pattern

AJAX pattern

AJAX stands for Asynchronous JavaScript and XML. It is responsible for reloading only that data which undergoes changes in spite of loading the whole page. It allows partial rendering or fetching of the data.

This pattern is based on AJAX Request (XMLHttpRequest) and AJAX Response (XMLHttpResponse) model. AJAX request involves two methods: open ( ) for opening the desired URL and Send ( ) to send the request to the server.

The work flow of this pattern will be like this. The client requests for a particular object and thus AJAX request has been made and send to the server. The server will receive the request, reacts to it as per the current state. If the current state of the server is in ready state, then the request will get proceed and the server will create the AJAX response with the desired information seek by the client. This response will send back to the client's end. The client side will then proceed this response and gets updated at end-user's interface.

The working of this design pattern can be well understood by the figure mentioned below.

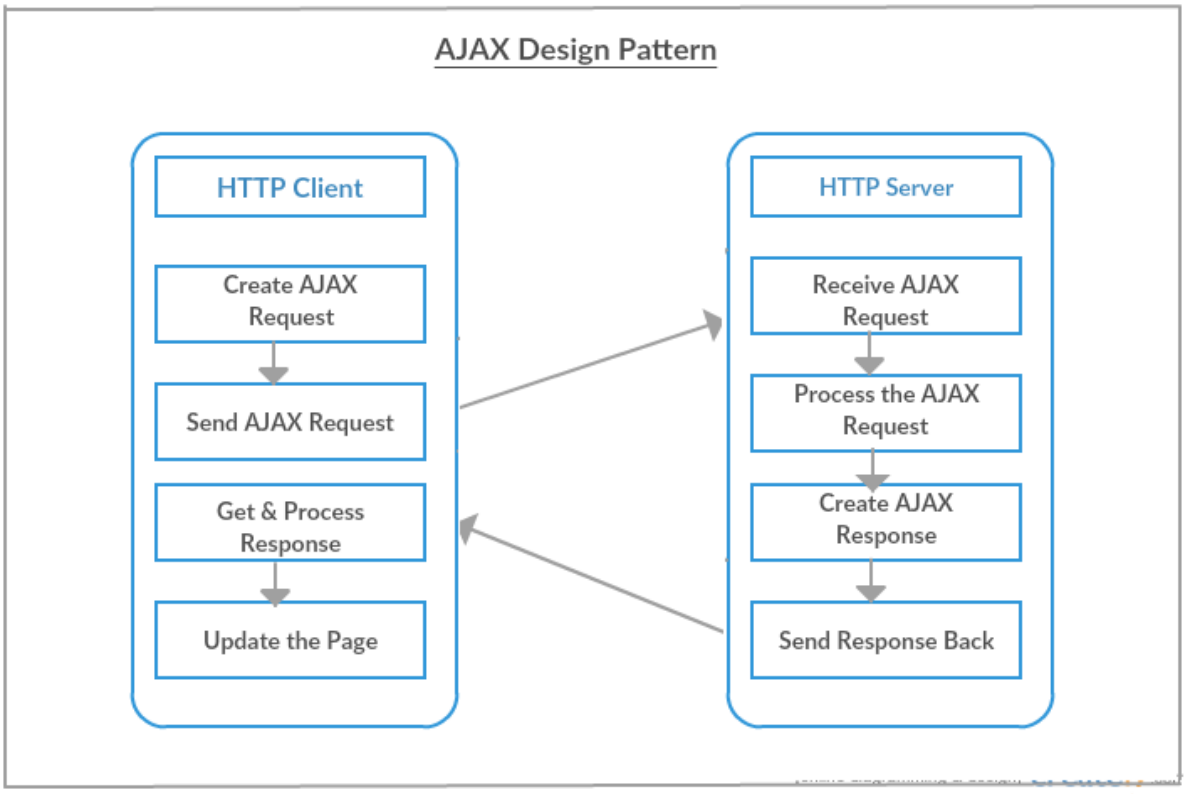

Fig 57: AJAX Design pattern

4.3 JavaScript Libraries

MapBox.js

Mapbox has a javascript API, which makes map building faster and more repeatable. It combines opensource projects such as Wax, Easey, and markers.js to develop interactive maps. It is built upon vector maps, which renders data in the real-time. The rendering is super-high and fluid in nature responding quickly to user feedback. The sizes can be controlled to nearly 75 percent more or less zoom in and out.

More functions on the map include changes on the fly, smoothing transitions, icons and labels adjustments. The maps can also instantly queried and the information can be retrieved fast since the data is present mostly on the client rather than the server.

The javascript API highlights upon stability, consistency and performance. MapBox.js is lightweight and can be cached in browser. The tile layers are set with mapbox.layer under the mapbox interface. The maps are set with mapbox.map controls.

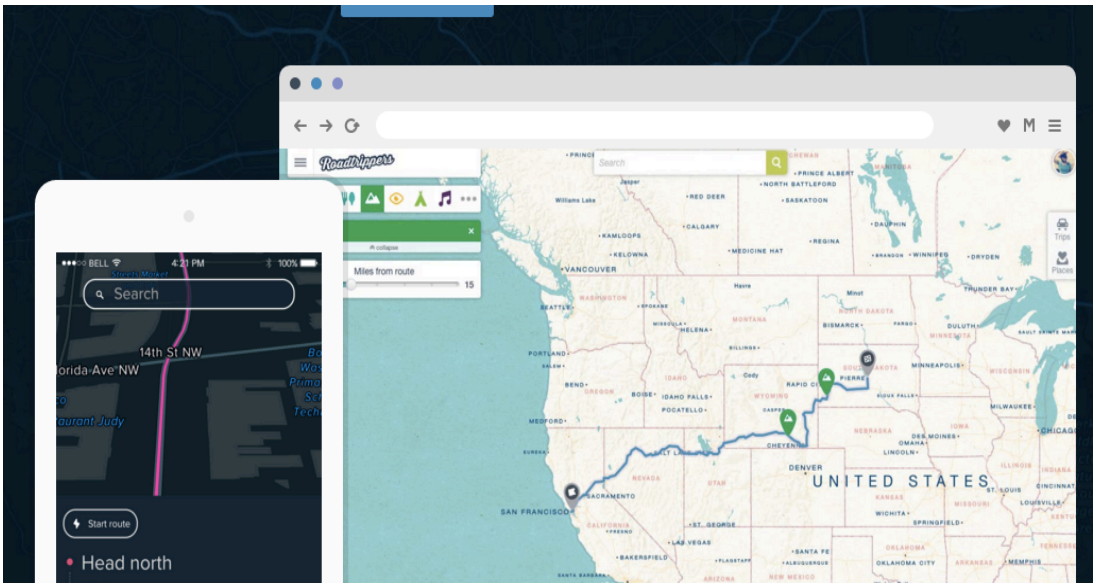

Fig 58: MapBox js

SpartaMaps utilizes all the cool features of the MapBox.js to render the map. The javascript API allows for polygon specification, Marker drawing and navigation through Mapbox Studio. It also helps to draw the shortest path between two places and also specifies the current location of the user.

MapboxGL.js

MapboxGL.js is a javascript library that uses WebGL and mapbox.js to render maps. The maps are vector based and is also a part of the mapBox ecosystem. It is written for both mobile and desktop applications.

Fig 59: MapBoxGL.js

SpartaMaps uses mapboxGL.js to render interactive maps. With just a simple Map instance, the javascript library creates a map and offers many methods to increase the interactivity. Some of the methods used are fitBounds, getStyle, getZoom, which are used for bounding, styling and zooming the map areas. The navigation is achieved by defining the latitude and longitude position of the marker.

D3.js

D3.js is a javascript library for bringing the DOM manipulated objects to life. It is basically used for drawing graphs and charts which showcase the data dynamically in real-time. It uses SVG (Support Vector Graphics), HTML and CSS to denote the real-time data. It allows binding data to the Document Object Model. After binding, dynamic transformations can be applied upon the data. The transformations are also smooth and interactive. D3 supports big datasets. It is extremely flexible and has minimal overhead capabilities. D3 also allows for component reusability through a variety of plugins and features. The elements can be selected using predicates like ID, class attributes etc. D3 also allows for backward compatibility using Sizzle. One unique feature of D3.js is that it also allows for geographical path function modeling using geo coordinates and SVG path data.

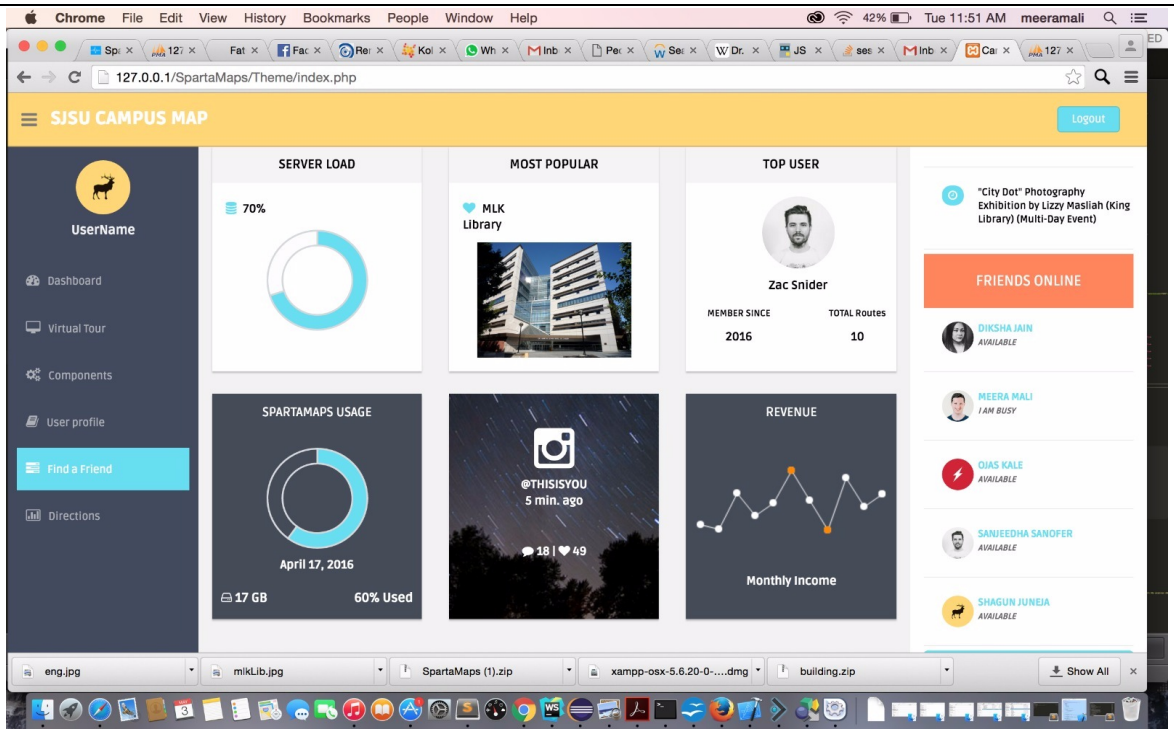

Fig 60: d3.js

In SpartaMaps, D3.js is used to get the real time data for the number of visits the user makes to the application. The calorie counter is also dynamically modeled using d3.js.

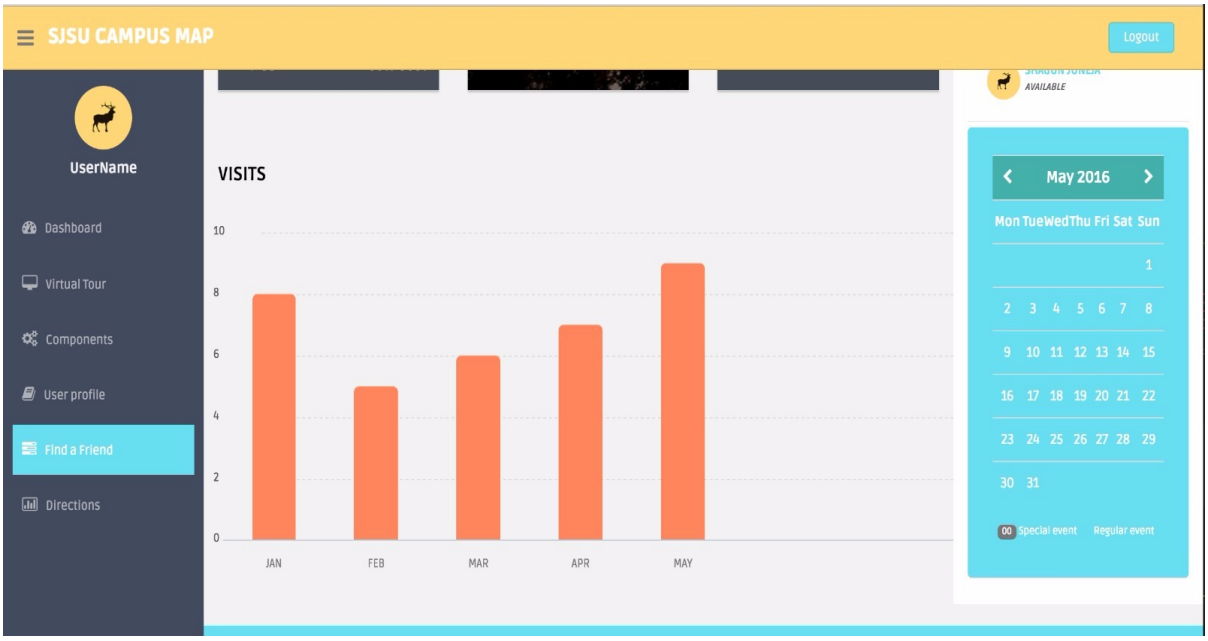

Fig 61: d3.js in SpartaMaps

Leaflet.js

Leaflet.js is completely opensource javascript library for drawing mobile intereractive maps. It gives the ability to implement more good features on the map. It is designed with simplicity, performance and usability all hallmark traits and the code with which the script is defined is also very simple making it easy to change only certain behavior to run the desired characteristic in the maps. The whole process starts with the creation of a simple div for the maps and tiles, markers can be added at the top of it.

SpartaMaps uses this property of leaflet.js to design interactive maps and to add the desired mapping functionality such as creation of markers, establishing route from one point to another, creation of polygons for each building and much more.

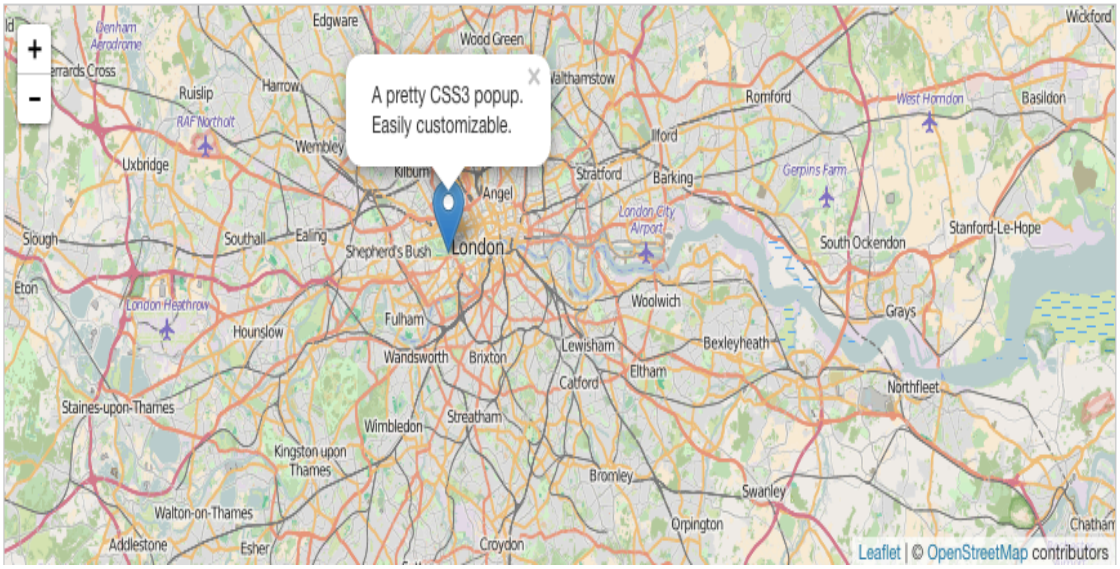

Fig 62: leaflet.js in SpartaMaps

Node.js

Node.js is an opensource library for creation of web servers and networking tools using javascript. It was originally designed to reduce the complexity of developing server applications. The characteristic of node.js is that it operates upon a single-thread, using non-blocking I/O calls allowing it to perform concurrent operations. Any function, which performs node.js involves a function called as callback to make it highly concurrent.

In SpartaMaps, node.js is used to create a server for find friends application. For a publish and subscribe scheme to work, there has to be a server connection which has to take all the subscribe requests from the clients connecting with it.

Typewriter.js

Typewriter.js is used to get a typewriter like font effects to the HTML elements. In SpartaMaps this effect is used to create a typewriter effect for the homepage design. It creates an actionable element with the title and helps to increase the interactivity of the page.

Bootstrap.min.js

Bootstrap.min.js is a minified and compiled version of CSS and Javascript for Bootstrap architecture. This minified version has to be downloaded which will open up into a set of hierarchies to enable bootstrap onto the page. SpartaMaps uses this js framework for creating bootstrap on the Homepage. Bootstrap is mainly for responsive design.

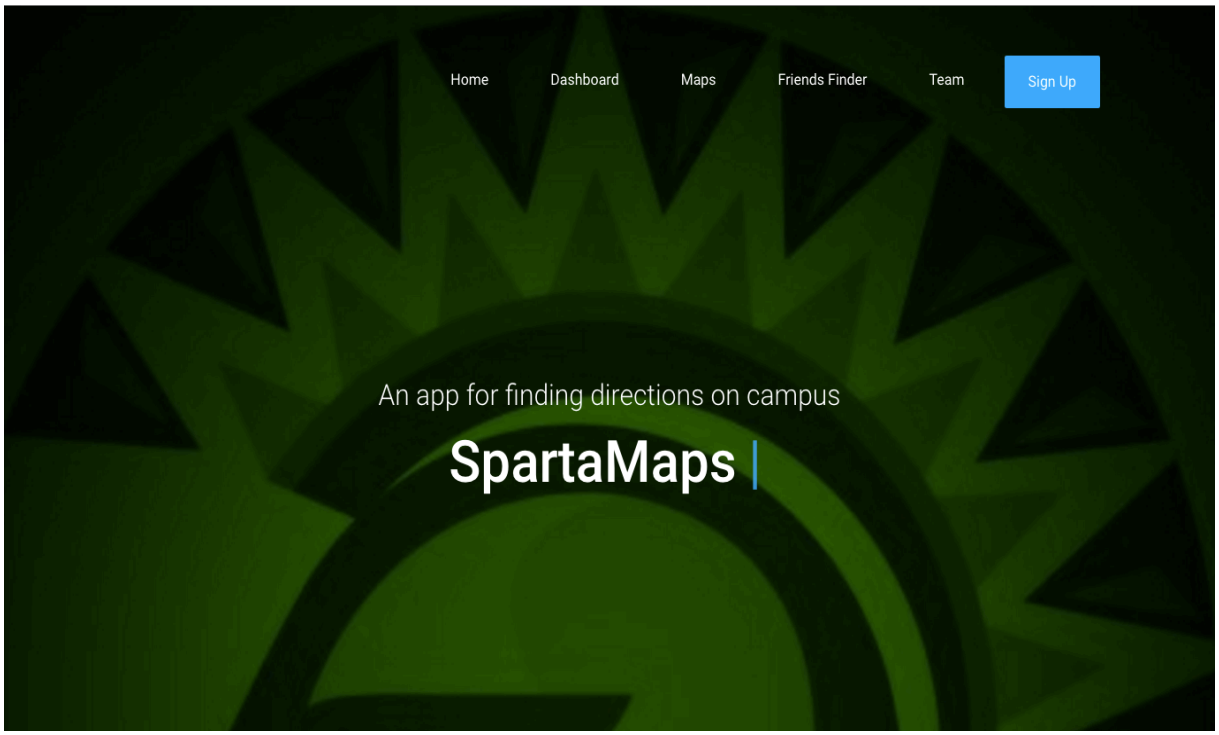

Fig 63: typewriter.js & bootstrap.js in SpartaMaps

4.4 Hardware Devices

Ricoh Theta S

The Ricoh Theta S was first announced in October 2013. With the wide spread use of Virtual reality headsets and other 360 degree photo & video capture devices, Ricoh started to evolve as a primary 360 camera. The camera uses two ultra-wide angle lenses to capture a fully spherical image when stitched together in the camera. For Ricoh Theta S, there are twin lenses with folded optics allowing for longer optical path while the build is small and easily accessible. The Ricoh Theta S can also record a spherical video with stereo sound in full High Definition at 30 frames per second for upto 25 minutes. The camera originally has a 8 GB storage and can hold up to 1500 photos but there is a strain on the battery power. Taking a picture or the video with the camera is very simple and the camera need not be moved while taking pictures.

SpartaMaps uses Ricoh Theta S to capture 360- degree spherical photo and video of each building specified to capture the liveliness of the building and the environment.

Fig 64: Ricoh theta S

Google Cardboard

Google Cardboard is used to develop accessible virtual reality to create liveliness of the scene and to be simple in a natural way. The cardboard SDK's available for the cardboard helps to start creating VR applications or adapt an existing application for virtual reality.

The VR view of the google cardboard allows embedding 360 degree VR media into websites on desktop and mobile. This technology is designed for the developers of the traditional applications to immerse them into the virtual environment. The VR view supports mono and stereo 360 images and videos across the web, android and iOS.

Images captured from the Ricoh Theta can be converted into VR view using Ricoh theta application and can be viewed through the google cardboard to provide an immersive environment

In SpartaMaps, the images and videos of the buildings captured from Ricoh Theta are converted into VR view using two-lens mode in the google cardboard and the images can be viewed. The path from MLK library to Clark's Hall was captured on Ricoh Theta S and viewed using Google Cardboard.

Fig 65: Google Cardboard

Fig 66: VR Content

4.5 Pagination

Pagination is the process of dividing a document into discrete pages, either electronic pages or printed pages.

Today outputting an electronic file to a printing device, such as a desktop printer or a modern printing press, usually produces printed pages. These electronic files may for example be Microsoft Word, PDF or QXD files. They will usually already incorporate the instructions for pagination, among other formatting instructions. Pagination encompasses rules and algorithms for deciding where page breaks will fall, which depend partly on cultural considerations about which content belongs on the same page: for example one may try to avoid widows and orphans. Some systems are more sophisticated than others in this respect. Before the rise of information technology (IT), pagination was a manual process: a human decided pagination. Today, machines perform most pagination, although humans often override particular decisions (e.g. by inserting a hard page break).

In reference to books produced without a computer, pagination can mean the consecutive page numbering to indicate the proper order of the pages, which was rarely found in documents pre-dating 1500, and only became common practice c. 1550, when it replaced foliation, which numbered only the front sides of folios.

On the Internet, pagination is used for such things as displaying a limited number of results on search engine results pages, or showing a limited number of posts when viewing a forum thread. Pagination is used in some form in almost every web application to divide returned data and display it on multiple pages. Pagination also includes the logic of preparing and displaying the links to the various pages.

Pagination can be handled client-side or server-side. Server-side pagination is more common. Client-side pagination can be used when there are very few records to be accessed, in which case all records can be returned, and the client can use JavaScript to view the separate pages. By using AJAX, hybrid server/client-side pagination can be used, in which Javascript is used to request the subsequent page, which is loaded and inserted into the Document Object Model via AJAX.

Server-side pagination is appropriate for large data sets providing faster initial page load, accessibility for those not running Javascript, and complex view business logic.

Correctly implementing pagination can be difficult. There are many different usability questions such as should "previous" and "next" links be included, how many links to pages should be displayed, and should there be a link to the first and last pages. Also ability to define the number of records displayed in a single page is useful.

Pagination in SpartaMap.

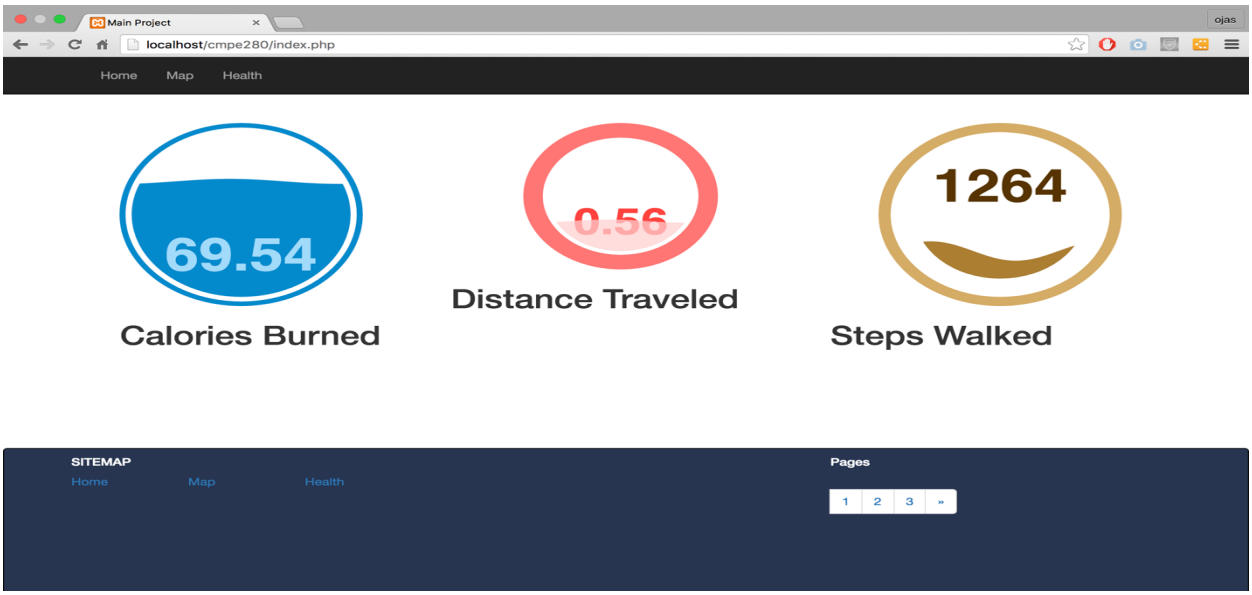

Fig 67: Pagination in Spartmap

This is the first page in pagination where the user is provided with the statistics of his last tour. Now pagination comes in picture here. The next pages provide the user with statistics of the last but one tour. On clicking the next button or it gives following screen.

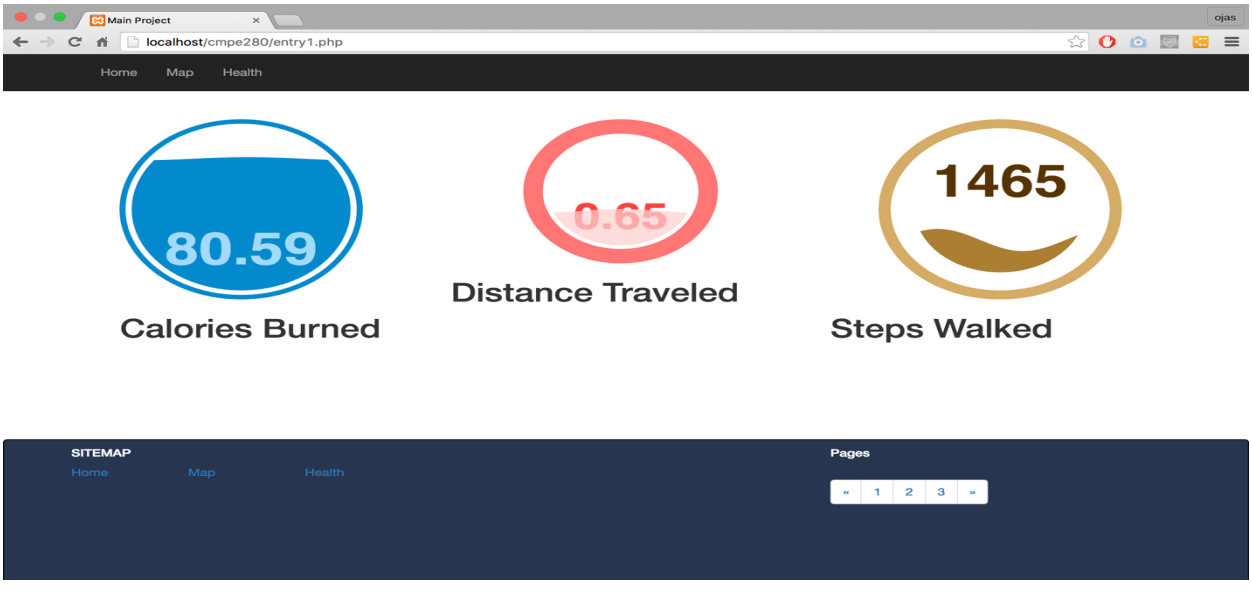

Fig 68: Pagination in Spartamap

As you can see the statistics are changes and as explained earlier it is showing the the stats of the second last tour of that user. The distance traveled and is more and thus for this longer tour of course the user has burned more calories as well as more steps walked.

Some of the next pages:

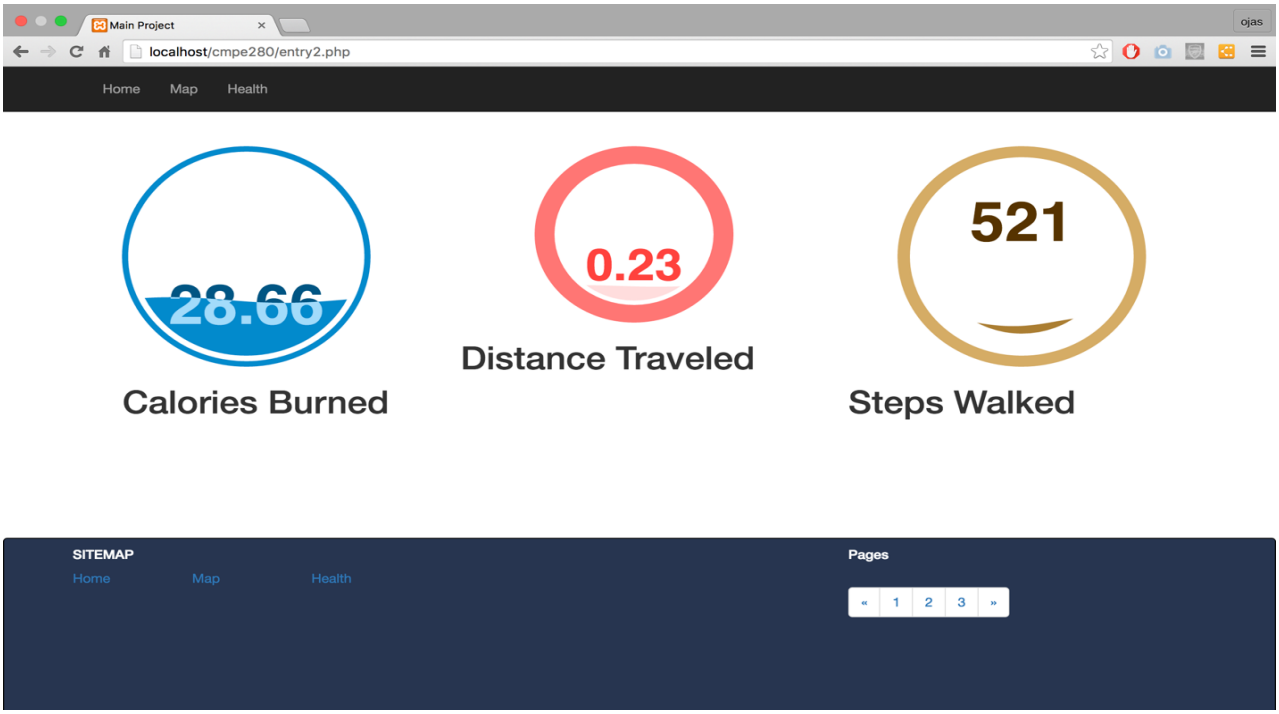

Fig 69: Pagination in Spartamap

This is the third last tour. With less distance travelled and less calories burned with few steps walked.

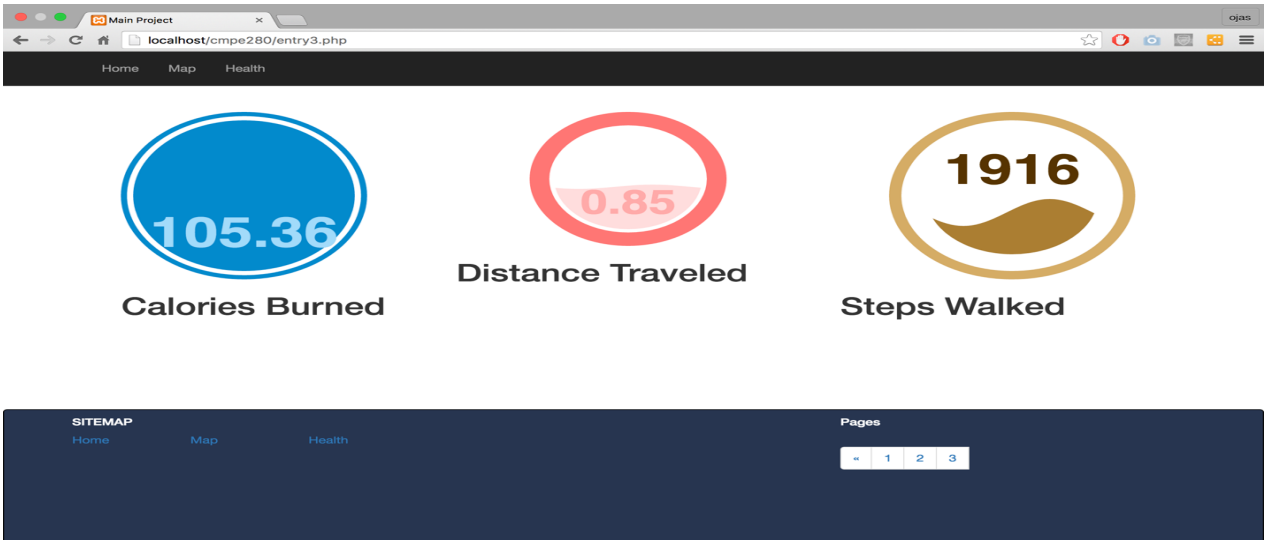

Fig 70: Pagination in Spartamap

This is fourth last tour he has done. It seems that as of now he has done only four tours and that's the reason the pagination index shows the 3 page of pagination. With more tours the user can see and navigate through his statistics of the last tours.

4.6 Localization

Localization is the process of adapting a product or content to a specific market or specified region or locale as we call it. Locale comprises of specific components and translating text. It includes adaptation of many aspects of the product like language for example.
Some elements used to achieve localization:

- Translation - To change the language of the content
- Graphics - To target specific markets

Website Localization

The process of adapting a website into different linguistic and cultural context is known as Website Localization. It involves a lot more than simple text translation.
The components of the website which are affected

- Content
- Images
- Overall design
- Requirements

Though they reflect the language and cultural specific changes, they still maintain the integrity of the website.

Language Codes

Language codes indicate the locales involved in the adaptation and translation of the product. They are used in various contexts. Language can be represented by two- and three-letter codes which have been specified by the International Organization for Standardization (ISO). For example:

Sr.No	Language	Variant	Code
1.	English	British American Canadian Indian	en-GB en-US en-CA en-IN
2.	Spanish	Mexican Spanish Colombian Spanish	es-MX es-CO
3.	Tamil	Indian Tamil SriLankan Tamil	ta-IN ta-LK

Localization in SpartaMaps:

Technology used

In our Website, we have provided localization with PHP scripts.

Design and Code

1. A PHP file is made for each language separately , with the language content defined inside as PHP constants.

2. PHP Constants: Their value cannot be changed in the script. They are automatically global across the entire script, unlike variables.

   Syntax: define(*name*, *value*, *case-insensitive*)

   Parameters: *name*: name of the constant
   value: value of the constant
   case-insensitive: Should constant name should be case insensitive. default:false

Languages provided:

- English
- Spanish
- Tamil

The php files for each language contains the constants defined for the front page of our website.

English (en.php)

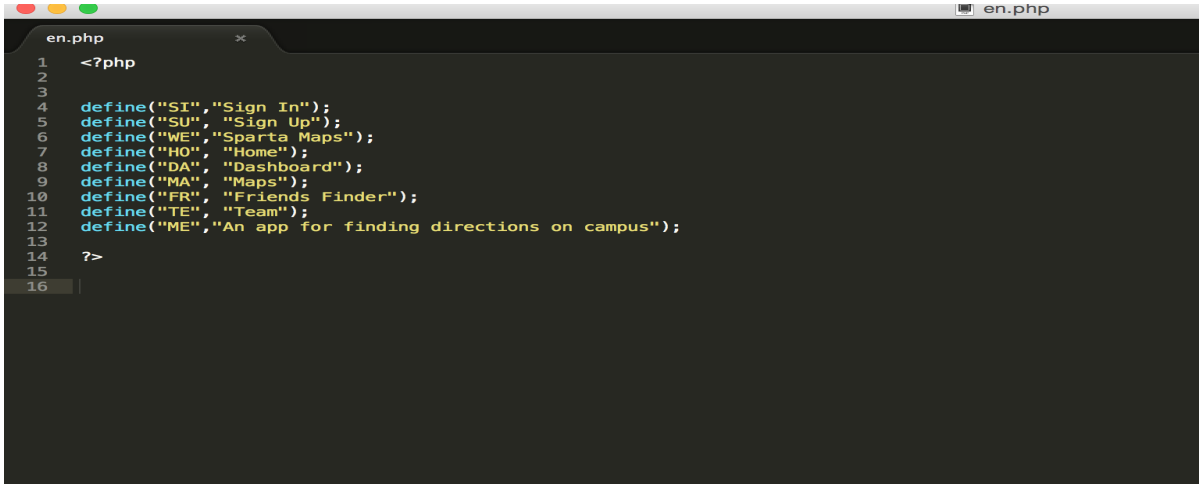

Fig 71: English code

Spanish (es.php)

es.php

```php
<?php


define("SI","Registrarse");
define("SU", "Regístrate");
define("WE","Sparta mapas");
define("HO", "Casa");
define("DA", "Tablero");
define("MA", "mapas");
define("FR", "Amigos Descubridor");
define("TE", "Equipo");
define("ME","Una aplicación para encontrar direcciones en el campus");

?>
```

Fig 72: English code

Tamil (ta.php)

ta.php

```php
<?php


define("SI","உள்நுழைய");
define("SU", "பதிவு");
define("WE","ஸ்பார்டா வரைபடங்கள்");
define("HO", "வீட்டிலல்");
define("DA", "கட்டுப்பாட்டு அறை");
define("MA", "வரைபடங்கள்");
define("FR", "நண்பர்கள் தேடுங்கள்");
define("TE", "அணி");
define("ME","வளாகத்திலல் திசைகளிலல் கண்டுபிடிதத்து ஒரு பயனல்பாட்டை");

?>
```

Fig 73: Tamil code

3. The appropriate language file is included in the main file for localization.

- *include 'ta.php';*

  These constants are used in the main file to show data.

  - Example:

  *<?php echo '
'.HO.'
' ?>*

  *<?php echo '
'.DA.'
' ?>*

  *<?php echo '
'.MA.'
' ?>*

  HO , DA, MA - They are the php constants defined in the language file.

4. To provide localization completely for the website , we can store the language preference of the user in php sessions.

   - *$_SESSION["lang"]="en";*

This session variable is set according to the user's language preference, and is accessed in all the web pages. Content will be displayed accordingly in the preferred language.

English(en)

index.php

```php
<?php
session_start();

$_SESSION["lang"]="en";
include 'en.php';
if (isset($_SESSION["login"]))
{
    $flag = "true";
} else {
    $flag= "false";
}
?>



<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

Fig 74: Tamil code

Spanish (es)

index.php

```php
<?php
session_start();

$_SESSION["lang"]="es";
include 'es.php';
if (isset($_SESSION["login"]))
{
    $flag = "true";
} else {
    $flag= "false";
}
?>



<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

Fig 75: Spanish code

OUTPUT Pages:

English

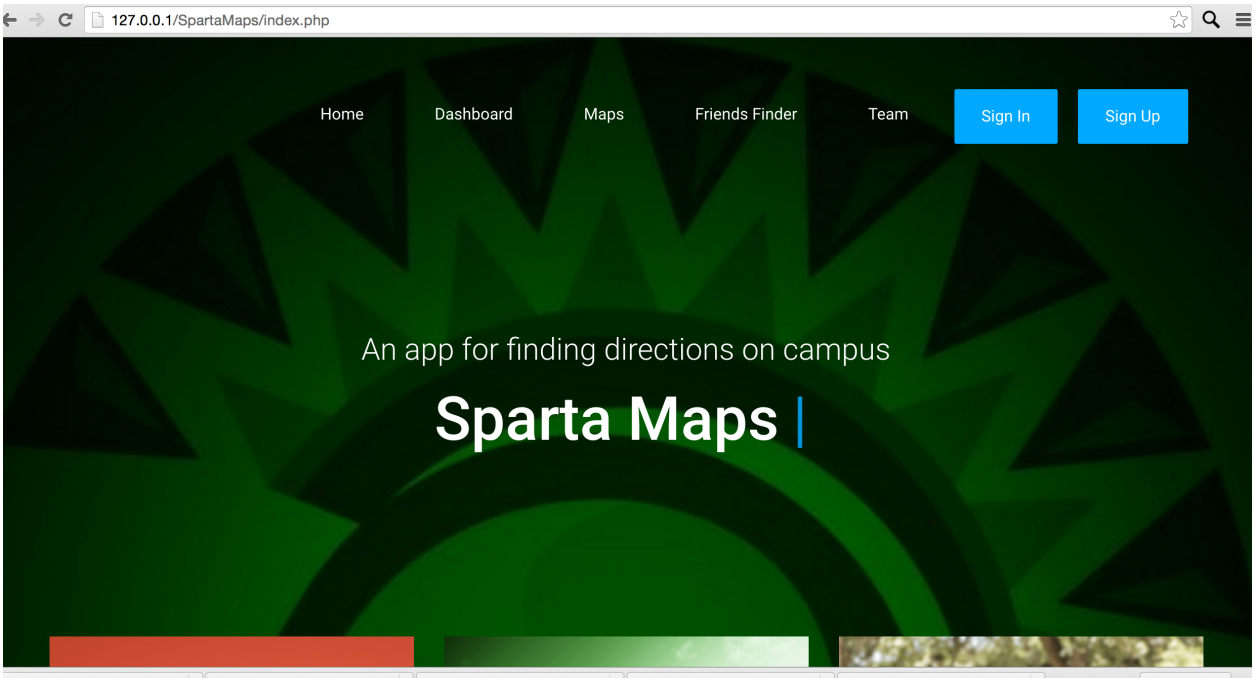

Fig 76: Homepage

Spanish

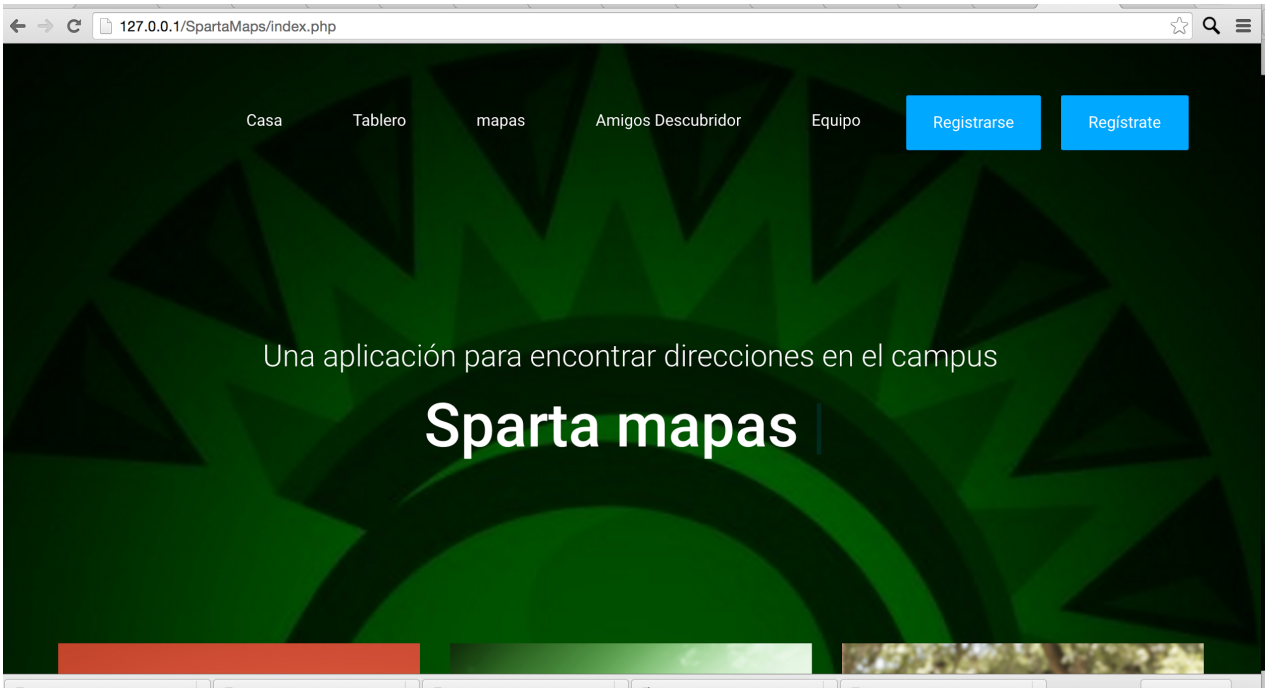

Fig 77: Spanish Translation

Tamil

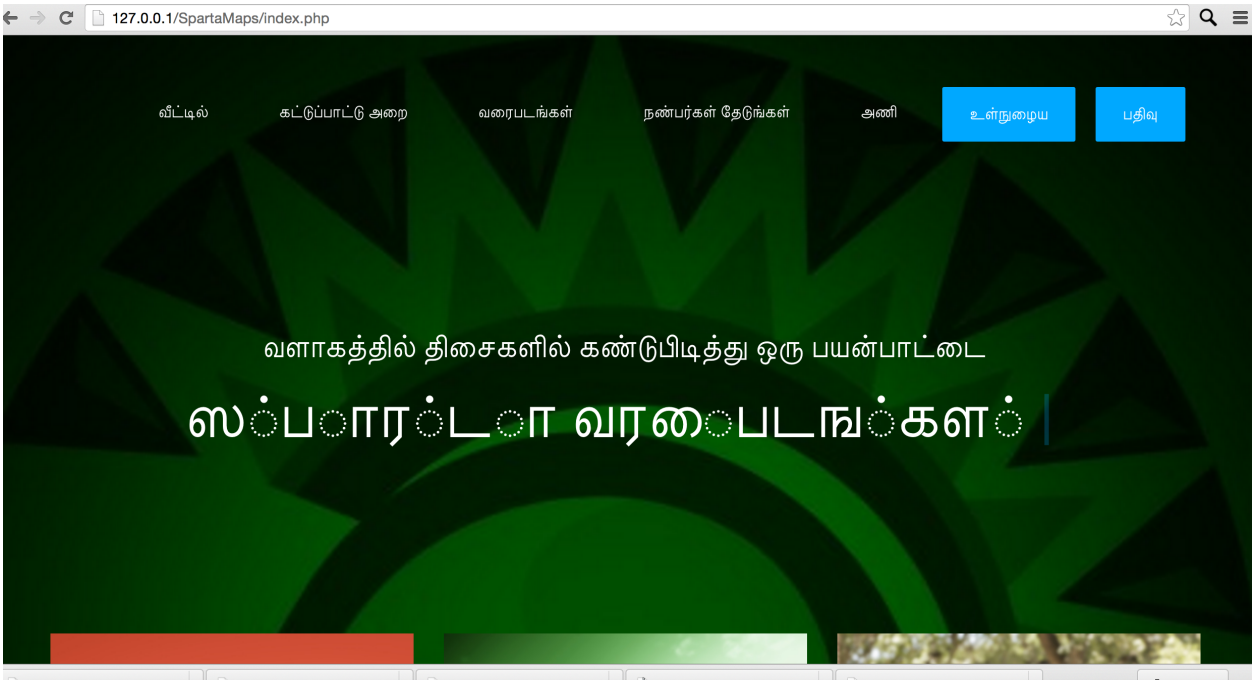

Fig 78: Tamil Translation

Scalability:

The localization can be provided on a large scale for all aspects of the website in various ways. **Google Translator API** can be used to translate the content to be locale specific. Some main languages translations provided by it:

Language	Code
Punjabi	ma
Romanian	ro
Russian	ru
Serbian	sr
Sesotho	st
Sinhala	si
Slovak	sk
Slovenian	sl
Somali	so
Spanish	es
Sudanese	su
Swahili	sw

Swedish	sv
Tajik	tg
Tamil	ta
Telugu	te
Thai	th
Turkish	tr
Ukrainian	uk
Urdu	ur
Uzbek	uz
Vietnamese	vi

4.7 Search Engine Optimization

A website is successful only when it gets noticed. To achieve this, we need to implement Search Engine Optimization techniques. Promoting a website is necessary to increase the number of visitors of the website. The process of improving the website in order to appear in the search results of search engines is known as Search Engine Optimization.

There are a lot of ways to achieve Search Engine Optimization. It can be using certain words on pages, or it can be taking care of the way your website is linked to other websites. The motive behind search engine optimization can also be to get your website built in a structured manner that makes it easy for the search engines to reach it and understand its structure. If your website is relevant to the content searched for, search engine optimization would benefit you as well as the users who are looking for information.

Search engine optimization involves finding out the words or keywords that are able to generate traffic, make it easily available to search engines and aid in marketing the website's unique offerings. The most important content which helps your website to be listed in the results is HTML format. Non-text content like images or flash files are given less preference or are sometimes even ignored by the search engines. The websites where image content is necessary, you can make use of alt attribute and give it a name, so as to give a text description in HTML format, so that it is visible to search engines. Added to this, audio or video content can be given with a transcript for the search engines to pick up and index.

Once a website is optimized, it can't appear in results forever despite of better relevant content being available in the search results. To ensure this, search engine optimization practices keep changing over the time to get the best results in search. One should be updated about the change in search engine rules to get the most out of it. The most common techniques to achieve search engine optimization are:

1. Your website pages must contain title tags and meta descriptions. Use of meta keywords can make sure search engine talks specifically to that page and that it is also formatted correctly. Also, the meta descriptions should be unique for every page of your website. Title tags should also be between four to eight words, and should be unique. These words should be selected in such a way that they grab the user's attention, and makes him want to click and read more.
2. As compared to plain text, it is much easier for media content for example videos, to rank on the first page. A developer needs to focus on mobile and tablet optimization as well as other media. HTML content is noticed the most easily by search engines, but content like videos can also be optimized by following the best practices for this content so that search engines can crawl that content and get credit to your website.

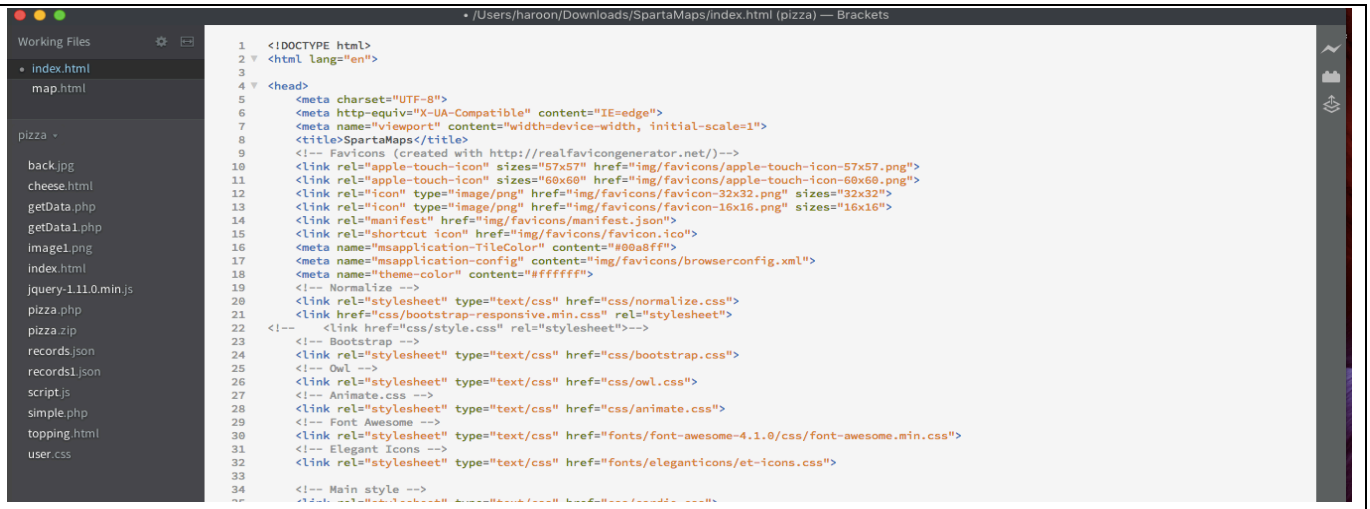

Fig 79: Use of Meta Tags in SpartaMaps

3. The domain name for your website is the base for searches, and should be consistent throughout the website. Use of sub-directory root domains (spartamaps/health) is better than subdomains (health.spartamaps.com). It gets you in a better position from search engine point of view. Also, the domain name should be consistent throughout your website. If you type www.spartamaps.com or spartamaps.com, it should redirect to the same website. If it doesn't, search engine may be looking at two different websites then. In such a case, the overall search engine optimization techniques you used will not have the desired effect.

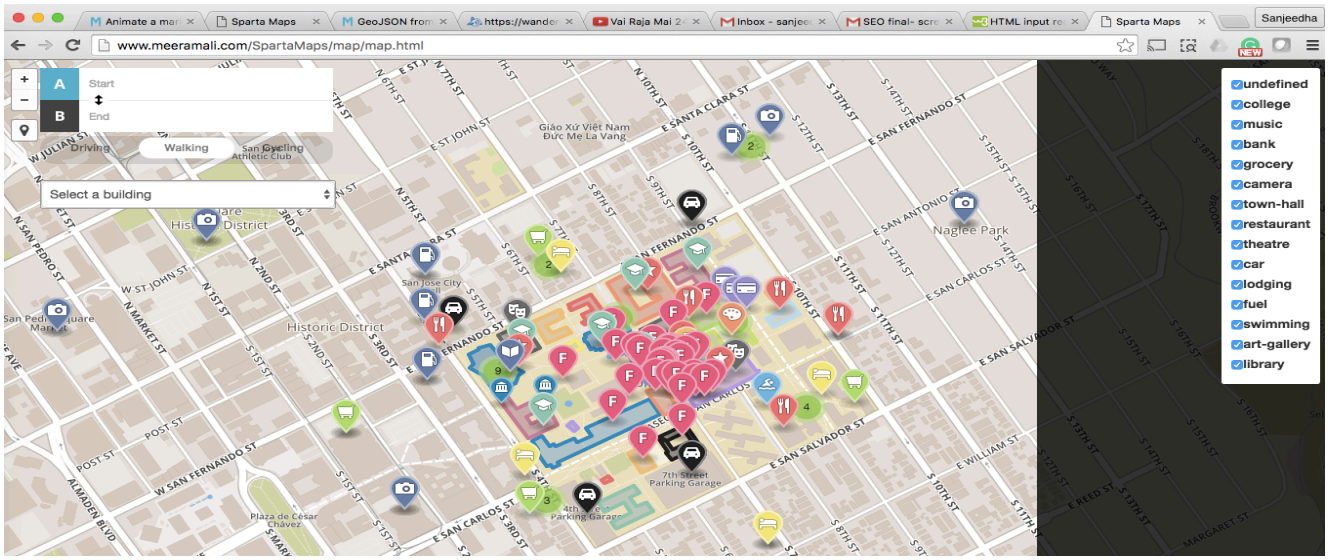

Fig 80: Use of route directory in SpartaMaps

4. Adding keywords to your website helps, but if the scope of your website is extended to other platforms using multi-channel optimization, i.e. giving links to platforms like Facebook, Twitter, LinkedIn, Email, ads, makes your website extend to other offline platforms as well. So, maintaining consistency along with extending to other platforms

will add to the branding of your website, and train the users to use specific phrases you are optimizing for.

Fig 81: Use of alt in SpartaMap images.

5. Search engine spiders only have a certain amount of data storage, so if you're performing shady tactics or trying to trick them, chances are you're going to hurt yourself in the long run. Search engine optimization techniques like adding keywords, ads and links to other websites prove to be beneficial but overdoing it will have a counter effect. Overusing keywords on pages, purchasing links, or providing a poor user experience including so many ads that the user can't find appropriate content will affect your optimization and increase your bounce rate.

To sum it up, the best practices to achieve search engine optimization will be to focus on the website theme, text on website pages, titles and descriptions on your website, performance of your website, providing rich content for other websites to link to as a reference or to cite some information and ensure that the bounce rate for your website is low.

Chapter 5

Application Testing

Software testing can be stated as a process of validating and verifying the software application such that it meets the business requirements and thus comes as a highly functional and robust application and also free from defects or bugs. Testing of an application is necessary to ensure the quality of the product, overall working of the application when deployed on different platforms.

We have defined the scope of our application while specifying the project requirements and thus we concluded that a good test coverage is required to test the project completely and make sure that it is performing well as per the requirements. So, in order to meet our scope we decided to test various modules of our project. Our Testing plan comprises of unit testing, integration testing, and regression testing and overall system testing while developing various modules of our project. We formed different test cases to test the application manually and have written test automation scripts by making use of various automation tool such as selenium, LoadRunner etc to check if the deployed application is ready for use.
Following are the various testing types that we have used to test the different modules of the application after or during the implementation of the code:

- *Unit Testing:* Tested all the small units of the project while development to make sure that the code is working properly.[1]

- *Component Testing*: Tested all the components or each module of the application individually to make sure each of them working properly as per expected.

- *Integration Testing*: Tested the effective functioning of the components when integrated with one another.

- *User Interface Testing:* We have tested the overall layout, proper images, logo or video fetching and high quality of the different screens in the application.

- *Back-end service Testing*: Tested for the back-end data storage, its security and reliability. Checked the proper retrieval of the data without getting it manipulated or updated. Tested the validity of the data entered by the user by using negative test scenarios also. We also checked for the proper create, read, update and delete operations.

- *Web Services Testing:* Tested for secure, uninterrupted, asynchronous and timely communication between client and server while making AJAX calls for request and response.

- *Documentation Testing:* Tested the documentation to check whether it is providing necessary information to be understand about the functionality of our application by a non-technical or technical person. We also tested the strings or texts which we have provided in our application whether if they are conveying proper information or not.

- *Cross browser compatibility Testing :* Tested the proper working of all the functionalities of the application on all the web as well as mobile platforms such as IE, Safari, Chrome, Mozilla and iOS.

- *Scalability Testing:* Scalability of each module is tested to ensure improved customer satisfaction.

- *Performance Testing:* Tested the performance of the system i.e. how fast application is responding when simultaneously many users are accessing the application.

- *Usability Testing:* Tested the usability of the application such that it should be user-friendly and easy to utilized and customized by the end-user.

- *Reliability Testing:* Tests are performed to make sure the application is from any failures or interruptions.

- *Load Testing:* LoadRunner is used as an automation tool to check how application behaves when desired load is given on the application.

- *Regression Testing:* After changing the behavior of one component, stable components within the same modules are also tested to ensure such that new functionality doesn't affect the existing feature.

Following figure shows the steps we followed while testing the entire application:

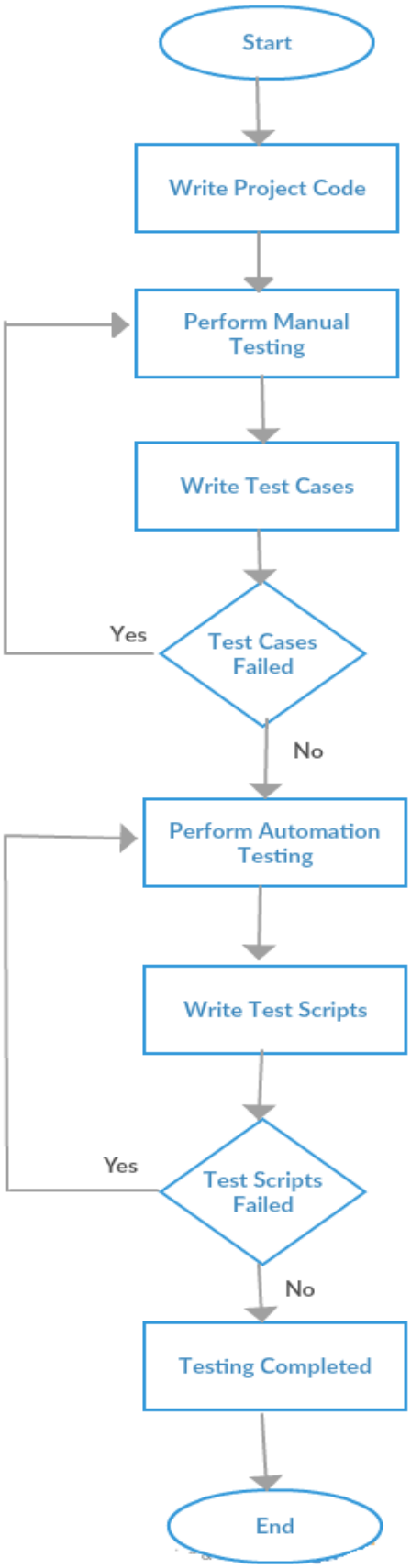

Fig 82: Testing Cycle

5.1 Usability Testing

Some of the test cases that we have considered while testing our application:

Test case ID	Test case Name	Test case Description	Expected Outcome	Actual Outcome	Result
01	Login	To check user can login.	The authenticate user should be able to Login into the application.	The user is able to login properly and is able to see the dashboard.	Pass
02	Edit Profile	To check user can edit the profile.	The user should be able to update his/her profile.	The user is able to update the profile.	Pass
03	Health Check-up	To check user can check his/her health.	The user should be able to check the steps count and calorie burnt.	The user is able to check all the health statistics.	Pass
04	Find friend	To check user can track other users on Map.	The user should be able to track the friends on the Map.	The user is able to track the friends on Map.	Pass
05	Find SJSU buildings	To check user can see the details of the buildings.	The user should be able to see details of the buildings while clicking on the Map.	The user is able to see the description, Gallery and Floor maps of the selected buildings.	Pass
06	Navigation	To check	The user	The user is	Pass

	on Map	user can see the navigation on the Map.	should be able to see the navigation on the Map.	able to see the navigation as per the current location.	
07	Virtual Reality	To check user can see the Video in 360’ view	The user should be able to see the video in 360’ view.	The user is able to see the video in 360’ manner.	Pass
08	Routing the path	To check user can get the path from source to destination.	The user should be able to get the shortest path between the two locations.	The user is getting the shortest path as per the type selected for the path.	Pass
09	Dashboard	To check user can see the usage details.	The user should be able to see the usage details on the dashboard.	The user is able to see the usage details on the dashboard.	Pass
10	Calendar Events	To check user can view the calendar events.	The user should be able to see the upcoming events on the calendar.	The user is able to see the upcoming events on the calendar.	Pass

5.2 Automation Testing

Testing the software with automation tools increases efficiency, effectiveness and overall coverage of the software testing. Automation testing helps in faster execution of the test case as per compare to the manual testing and also can be extended to perform tasks which are impossible with manual testing. It is very helpful when there is a repetitive execution of the same test cases with different scenarios.

Thus, in order to enhance the overall quality of our application, we have used Selenium and LoadRunner as automation Tools.

Selenium:

It is used to automate various web applications for testing purpose. We have used Selenium IDE (Integrated Development Environment) to test the various UI scenarios and important features of our application. It supports HTML, DOMs and JavaScript to create various test scripts. It supports 4 programming languages, which are: Java, Ruby, and Python and C #. It provides the facility to run a single test case to run entire test suites. We have recorded the test cases and later on we have played them in order to check the results.

LoadRunner:

LoadRunner is an automation tool to test the load on the application to determine the overall performance of the application. It determines the end-to-end behavior of the application when desired load is applied on the application and also to check the efficiency of the application, how application will behave when more load is there than the desired load.
It supports a wide range of web applications, mobile applications and useful in cloud testing also.

It simulates thousands of user simultaneously using the application, recording them by logging HTTP request between the server and the client and later it analyze the performance of the major components of the application. It simulates the user activity by generating messages from application components or by simulating interactions with client's interaction from the interface such as mouse movements or key pressing. [2]

Test case: 1
Test case name: Login
Description: To check user is able to login the application.
Result: Pass

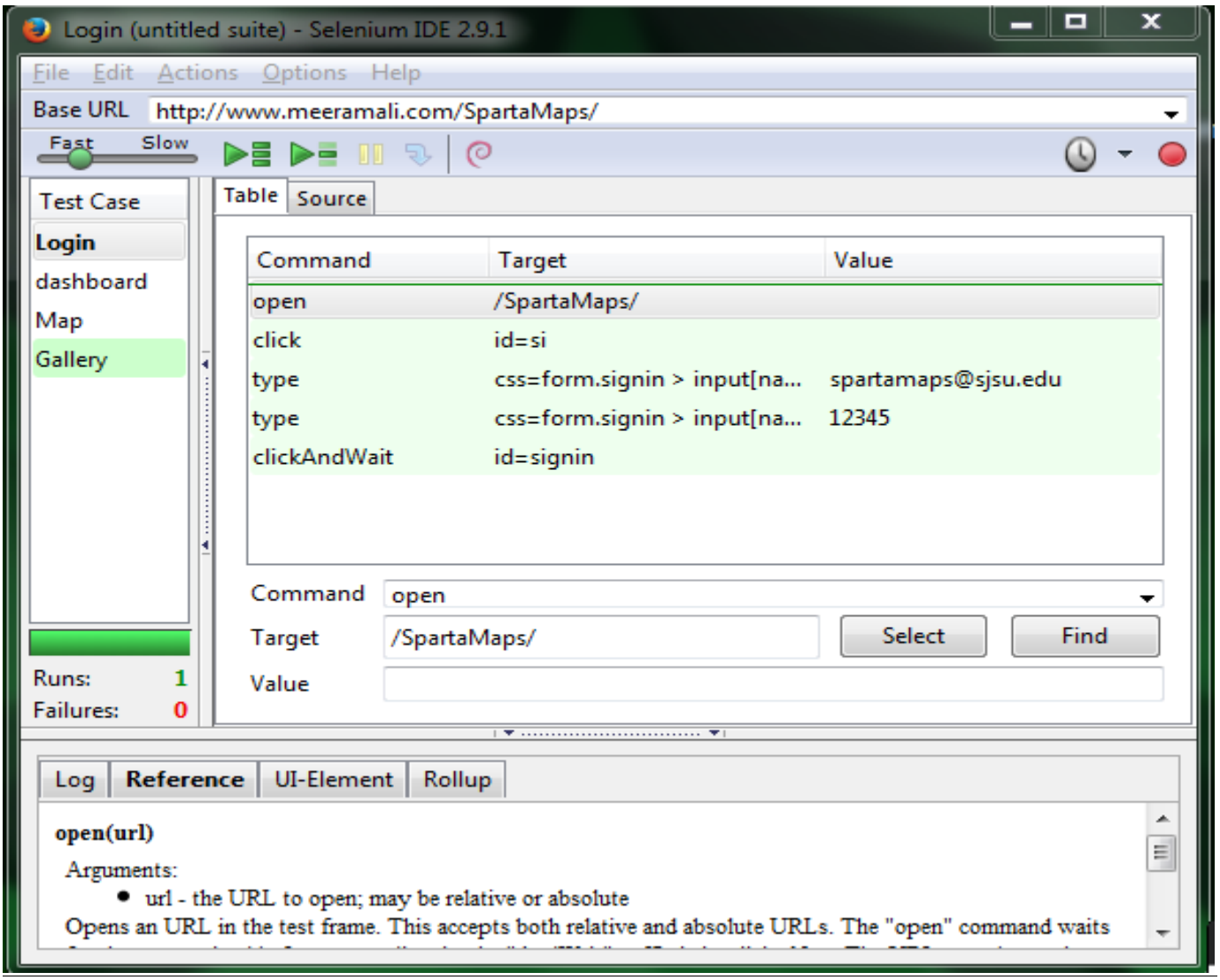

Fig 83: Test-Case 1

Test case: 2
Test case name: Dashboard
Description: To check if user is able to access the dashboard after successful login.
Result: Pass

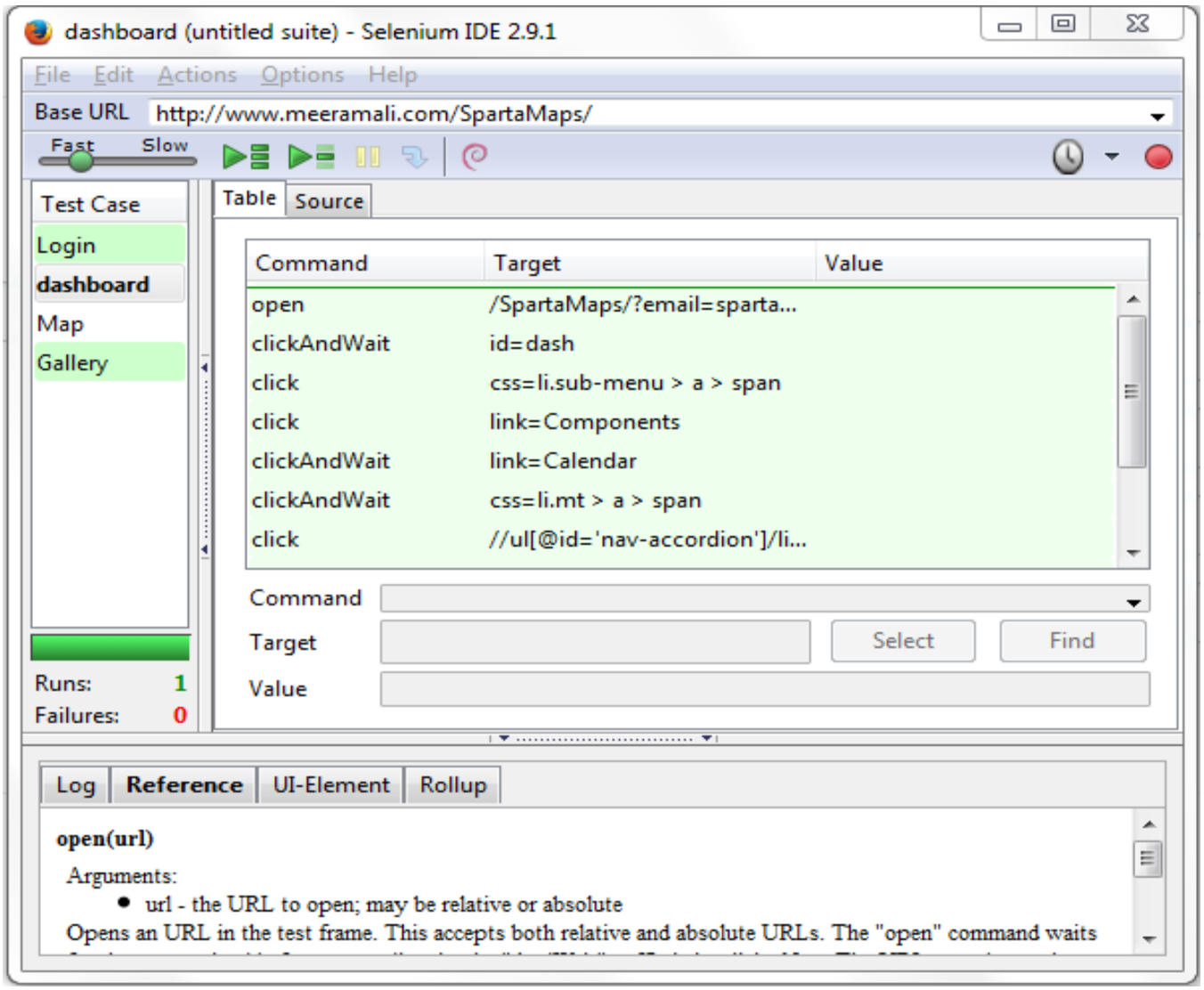

Fig 84: Test-Case 2

Test case: 3

Test case name: Maps

Description: To check if user is able to view the location of a building on the Map.

Result: Pass

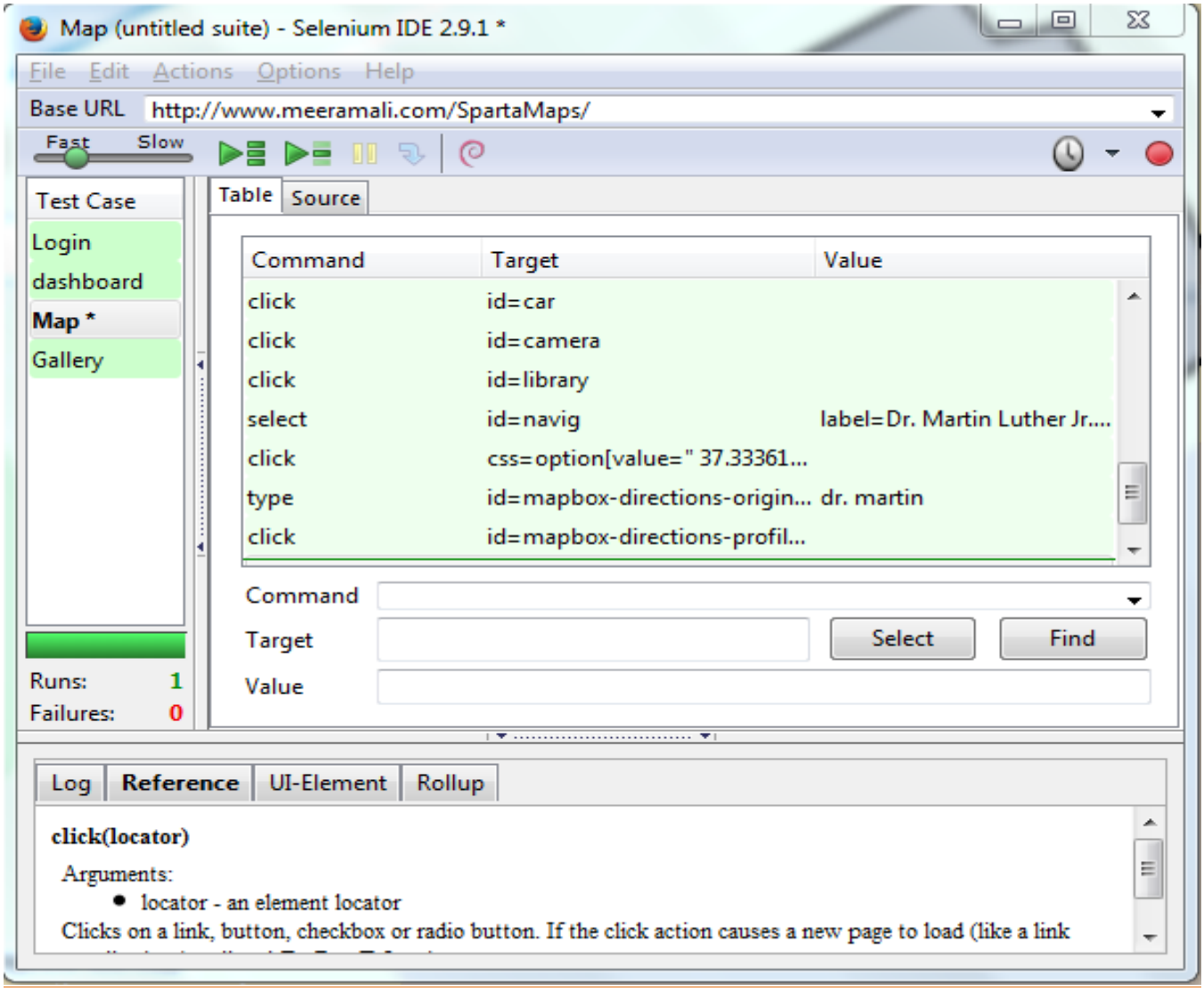

Fig 85: Test-Case 3

Test case: 4
Test case name: Gallery
Description: To check if user is able to view the gallery and related details.
Result: Pass

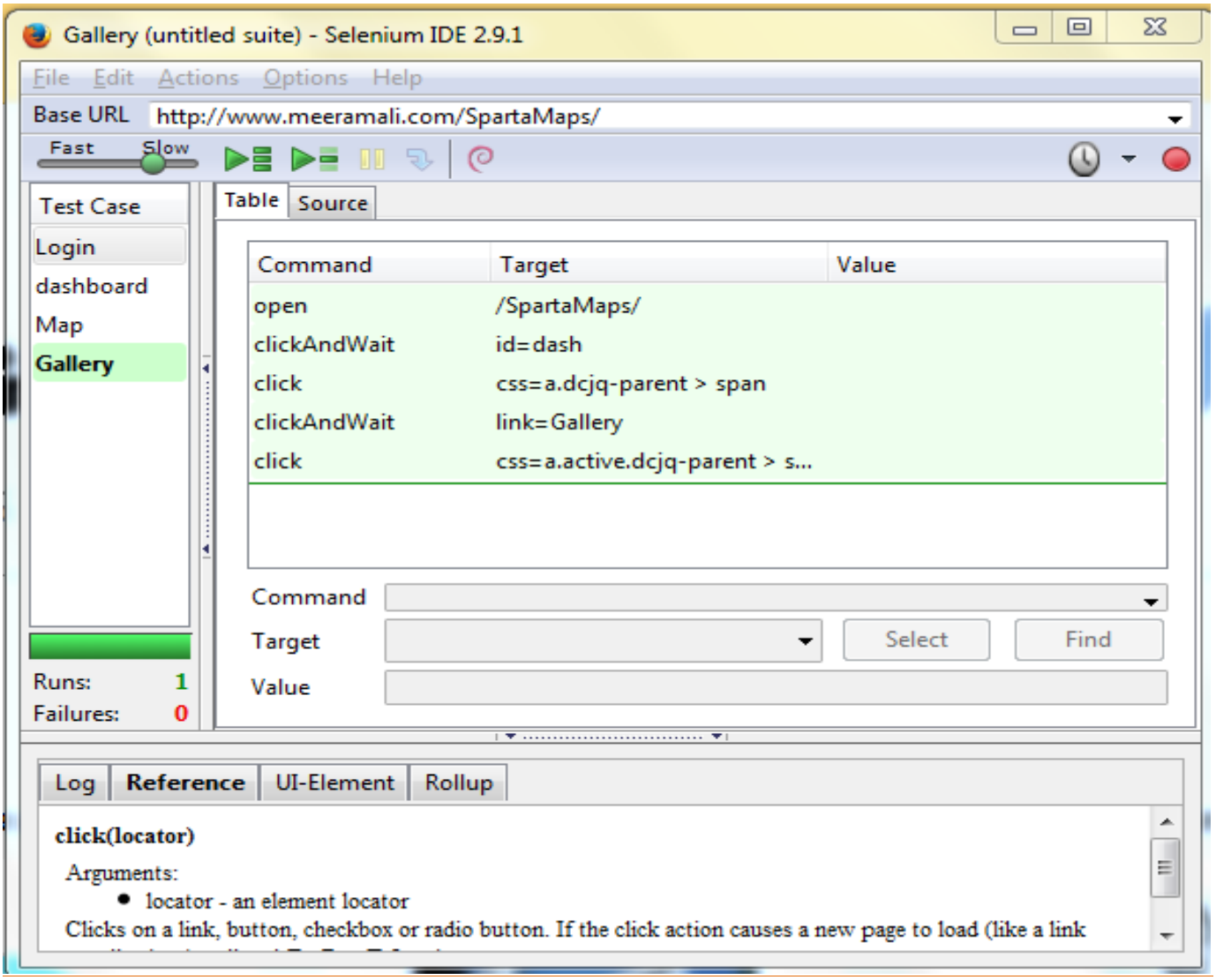

Fig 86: Test-Case 4

5.3 Cross Browser Compatibility

The term Cross browser compatibility means that the design and appearance of a website should be consistent regardless of any platform used to view it. This has to be taken care of by the designers and developers of the website as there are a wide variety of browsers like Mozilla Firefox, Google Chrome, Opera, Safari and Internet Explorer available today and a consistent performance with each of these across various existing versions is important. It is the responsibility of a website designer to make sure that their design is free of errors and completely functional for any platform or browser used.

Fig 87: Cross Broswer Compatibility

The advent of CSS3 brought with it much more easy to use features for website designers on one hand, but on the other hand, many new capabilities introduced in the CSS3 version are not supported by all major browsers. Some are not even recognized, hence ignored, which brings up serious design issues. Also, the support for design techniques vary with browser versions as well which adds to the complexity of the problem.

There are some tips and ways to achieve compatibility across browsers which are as follows:

1. All browsers have their layout defaults. One browser may have a tag defined with 8px margin, while other may have a 10px margin, say on body tag. Developers can't control the use of browsers, so has to make sure the site appears identical in every browser. Applying a browser reset stylesheet will make you take control of the default style of elements.

Eg:

```
* {
  margin: 0;
  padding: 0;
  border: 0;
}
```

2. CSS3 allows you to use many new features, but not every browser and its versions support them. For instance, CSS3 gradients. Older browsers ignore and skip the content and code they don't understand properties. We need to provide keywords before new properties so that preference is given to new properties like for instance, use of vendor prefixes is needed. So, when defining a CSS, write vendor prefixes, and after that, write W3C standard defined new property. This way, older browsers would use those properties because of prefixes and newer would simply use it according to the standard.

   The following vendor prefixes are used for each of the popular browsers:
   Safari, Chrome: -webkit-, Firefox: -moz-, Internet Explorer: -ms-, Opera: -o-

   Vendor prefixes are defined as:
   background: -webkit-gradient*(linear, 50% 100%, 50% 0%, color-stop(0%, #2c99ce), color-stop(76%, #459dcf), color-stop(95%, #74b9e0), color-stop(100%, #abe1fa));*

   background: -webkit-linear-gradient*(bottom, #2c99ce, #459dcf 76%, #74b9e0 95%, #abe1fa);*

   background: -moz-linear-gradient*(bottom, #2c99ce, #459dcf 76%, #74b9e0 95%, #abe1fa);*

   background: -o-linear-gradient*(bottom, #2c99ce, #459dcf 76%, #74b9e0 95%, #abe1fa);*

3. Avoid using nested elements and use right elements for right job. Ul and li elements should be used to create a navigation menu instead of using a series of paragraph <p> elements. Keep your CSS separate from your HTML page. Keep it for your content and do the styling separately.

4. A browser switches to quirks mode if your page doesn't include a proper valid DOCTYPE. The document type declaration (DOCTYPE), if not specified in the page, leaves it on the browser to specify the rules you will be using in your code which will result in inconsistency since every browser would have its own rules. You can define a DOCTYPE like:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

5. A lot of internet users do not have an updated version of their browsers which creates a need to test older version of browsers too to ensure compatibility. This task can be made easier by tracking the users of your websites and determining their browsers and versions to make it easy for you to prioritize browsers and versions for compatibility testing. Commonly used browser versions are: Internet Explorer 6, 7 and 8, Firefox 3, Safari 3, Opera 9 etc.
   Testing in all 3 versions of IE is a problem as one copy of windows will support only one version of IE. Using multiple machines, hacking, or virtualization, are some ways you can solve this problem with, but none of these are a feasible solution.

   Instead, you can make use of online testing services like:

   Adobe Browserlab:
   It is a service to test designs on multiple browsers and operating systems. Functions like zoom, measurement tools, rulers and ease of moving around areas of screenshots are some main features of this tool. The designers can save a combination of browsers for customizing and testing.

   Netrenderer IE:
   It is a free compatibility tester for different versions of Internet Explorer.

   Browsera
   This is a testing tool, which not only enables the designer or developer to use this as a screenshot tool, but also works at a site level instead of just a page level tool. This tool gets the pages on your website through crawls and tests them for layout inconsistencies and scripting errors.

   Litmusapp
   This tool provides a compatibility report in addition to website screenshot to be reviewed by clients.

   Browsrcamp
   This tool can be used to test compatibility with Mac OS browsers like Safari 3.1.2. It just needs the URL for your website as an input.

6. Make use of fall backs and avoid the breaking of your code across browsers. For example, do not forget to specify an alt attribute to provide an alternative display in

case an image fails to load for any reason. Avoid use of flash for navigation, or else provide an alternative that can be used if flash is not supported. Make use of <noscript> tags to define alternatives in case of disabled JavaScript as not every browser allows it by default, sometimes the user has to enable it.

7. Make use of **Javascript libraries** like **jQuery, YahooUI, MooTools, Dojo** abstract away the differences in the DOM, AJAX and JavaScript.

8. To fix IE design issues, there is a need for CSS and JS hacks which is not a good approach. As an alternative to giving hacks for code to work, including IE-only mark-up with conditional comments solves the problem in a better and cleaner way. Eg:

   <link type="text/css" href="style.css" />
   <!--[If IE]> <link type="text/css" href="IEHacks.css" />
   <![endif]--> <!--[if !IE]> <link type="text/css" href="NonIEHacks.css" />
   <![endif]-->

9. W3C Validation Service validates multiple versions of XHTML and HTML, outputting many useful errors and warnings to help users create a perfect website. **W3C Validator: http://validator.w3.org/**
   W3C Css Validator: http://jigsaw.w3.org/css-validator/

10. There are some JavaScript solutions that can help us serve responsive websites to older browsers lacking CSS3 support like **Respond.js** and **Modernizr.**

    Respond.js eases the task of understanding and executing media queries and media types for older browsers. The script, when included, adds support for max-width, min-width and other CSS3 instructions in order to work properly even in old browsers. Script you need to include:
    <script type="text/javascript" src="js/respond.min.js"></script>
    Modernizr checks if the browser has HTML and CSS features natively implemented or not using javascript driven feature detection, and then creates a JavaScript object using the results and adds classes to the HTML element which you can write CSS for. The following script has to be included to use it:
    <script type="text/javascript" src="modernizr-latest.js""></script>

Our project works with all versions of all browsers. The above guidelines like defining valid **DOCTYPE** and using Javascript libraries like **jQuery** and **Bootstrap** helped us to achieve cross

browser compatibility. We also made the use of **Vendor prefixes** to achieve compatibility across various browsers and platforms. The screenshots of our website on various devices are as follows:

1. **Internet Explorer:**

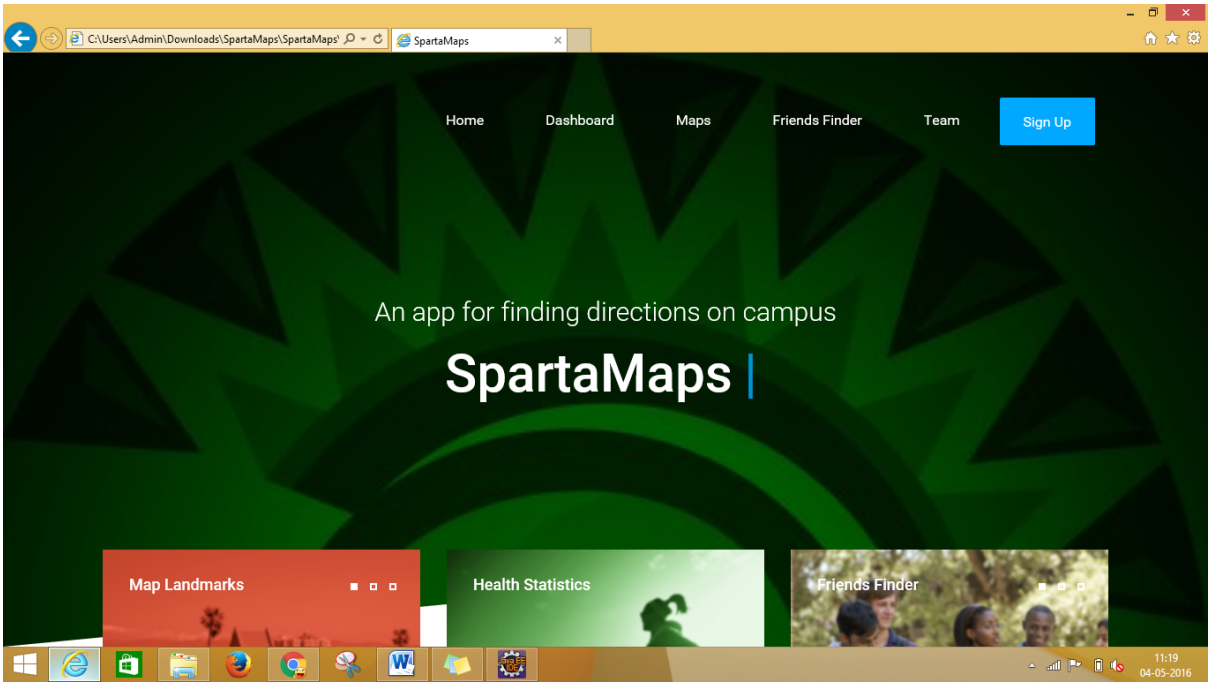

Fig 88: Cross Browser IE Compatibility

2. **Chrome**

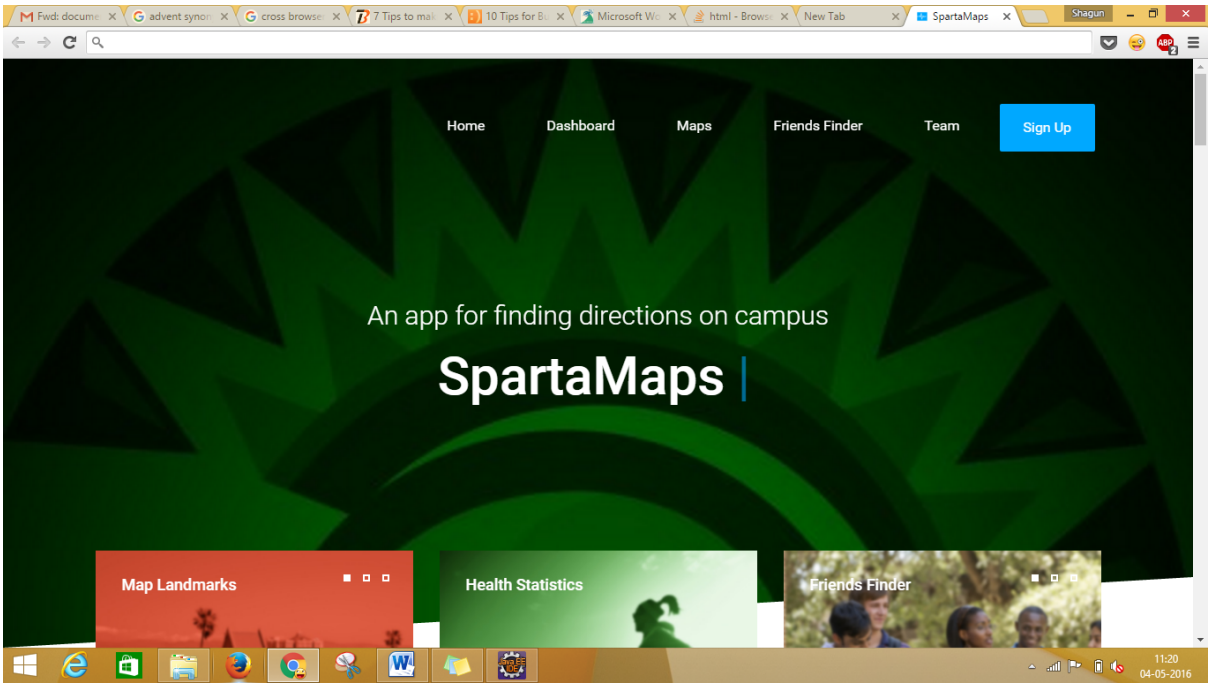

Fig 89: Cross Browser Chrome Compatibility

3. Firefox

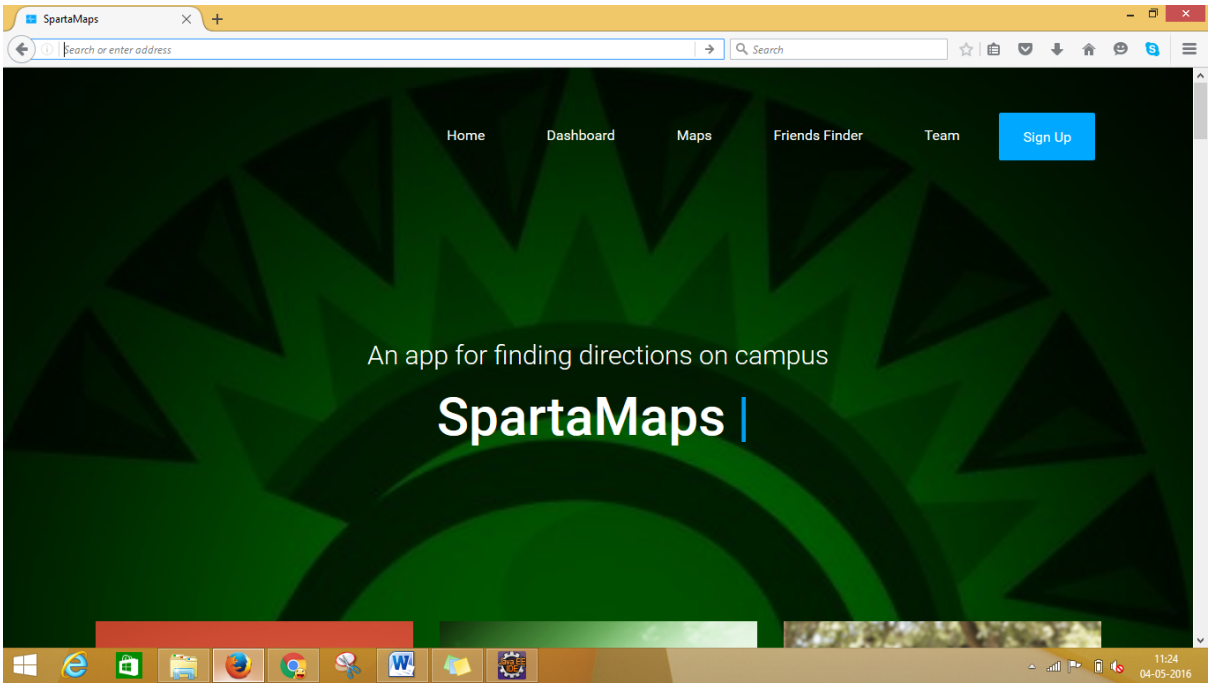

Fig 90: Cross Browser Firefox Compatibility

4. Safari

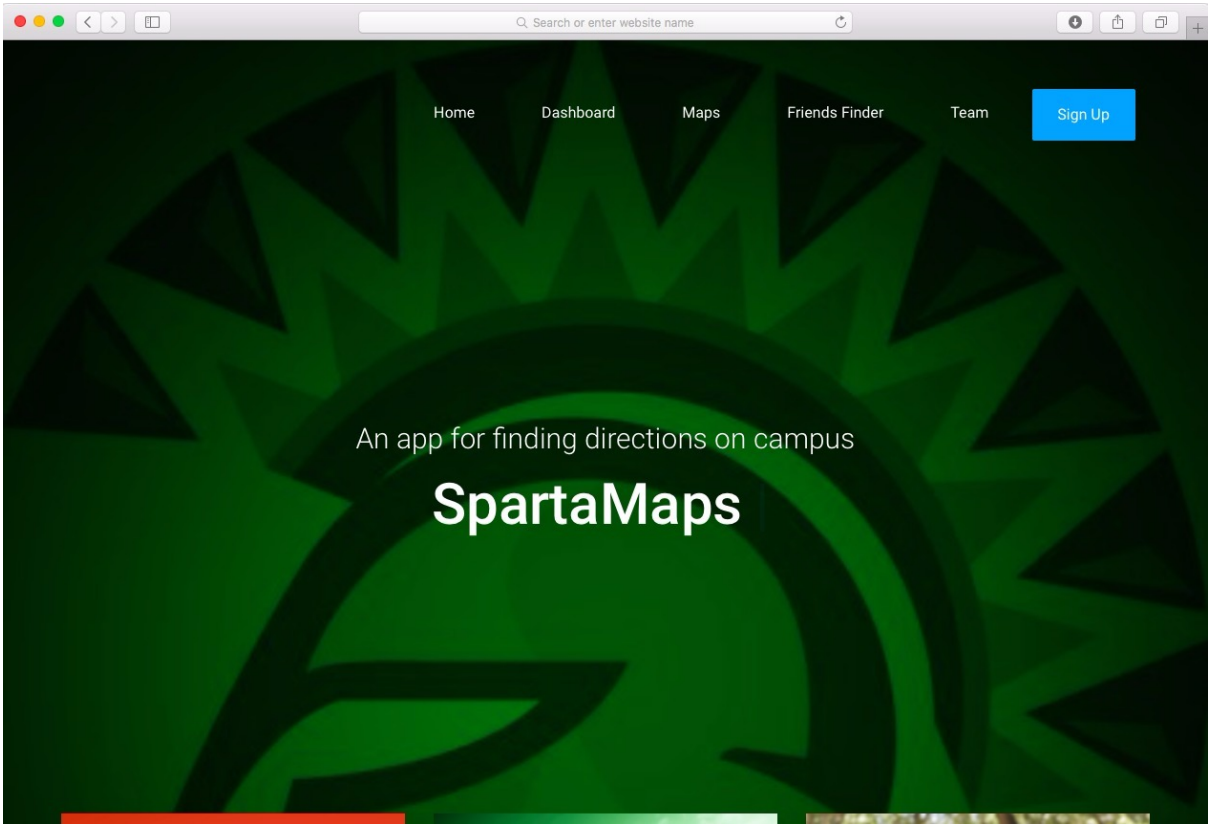

Fig 91: Cross Browser Safari Compatibility

5. Iphone 6

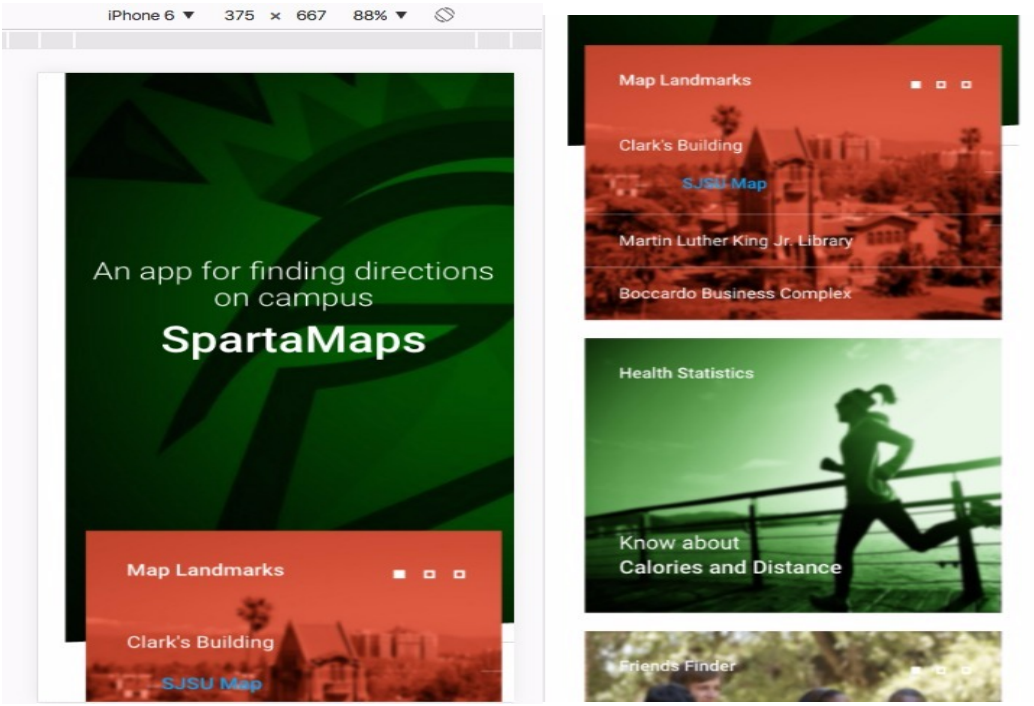

Fig 92: Cross Browser Safari Compatibility

6. Ipad :

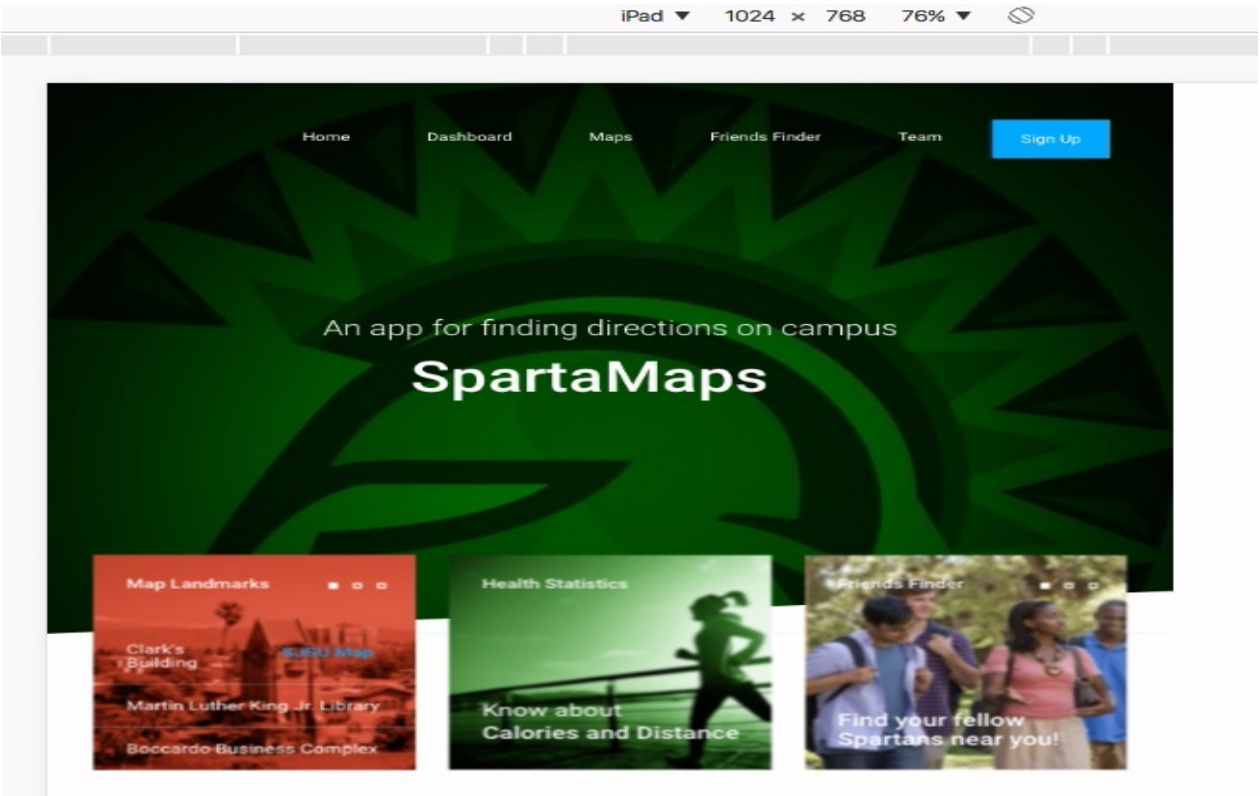

Fig 93: Cross Browser IPad Compatibility

5.4 Profiling

Profiling in general is defined as an activity of describing someone by collecting information about them. In computer science, Profiling the website is used to measure its performance in terms of various aspects.

There are many tools for the browsers to achieve profiling. We have used in-built Profiling tools provided by Google chrome browser (Chrome Developer Tools).

TimeLine Panel

This is a basic level measurement of our website. The *Timeline* panel is used record and analyze all the activity in your application as and when it runs. It is used mainly to investigate performance issues, which are perceived in our application.

A recording is initially created from which the timeline is retrieved.

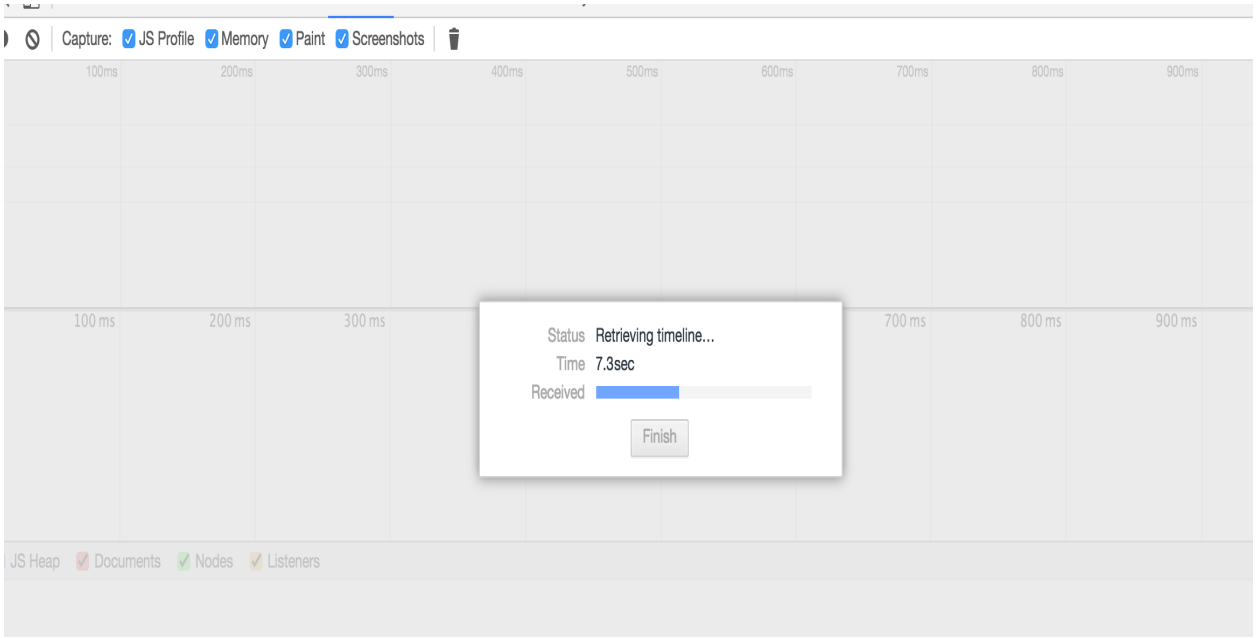

Fig 94: Timeline Panel

After the completion, the timeline Panel shows the recording details and the profiling details like JS Profile etc.

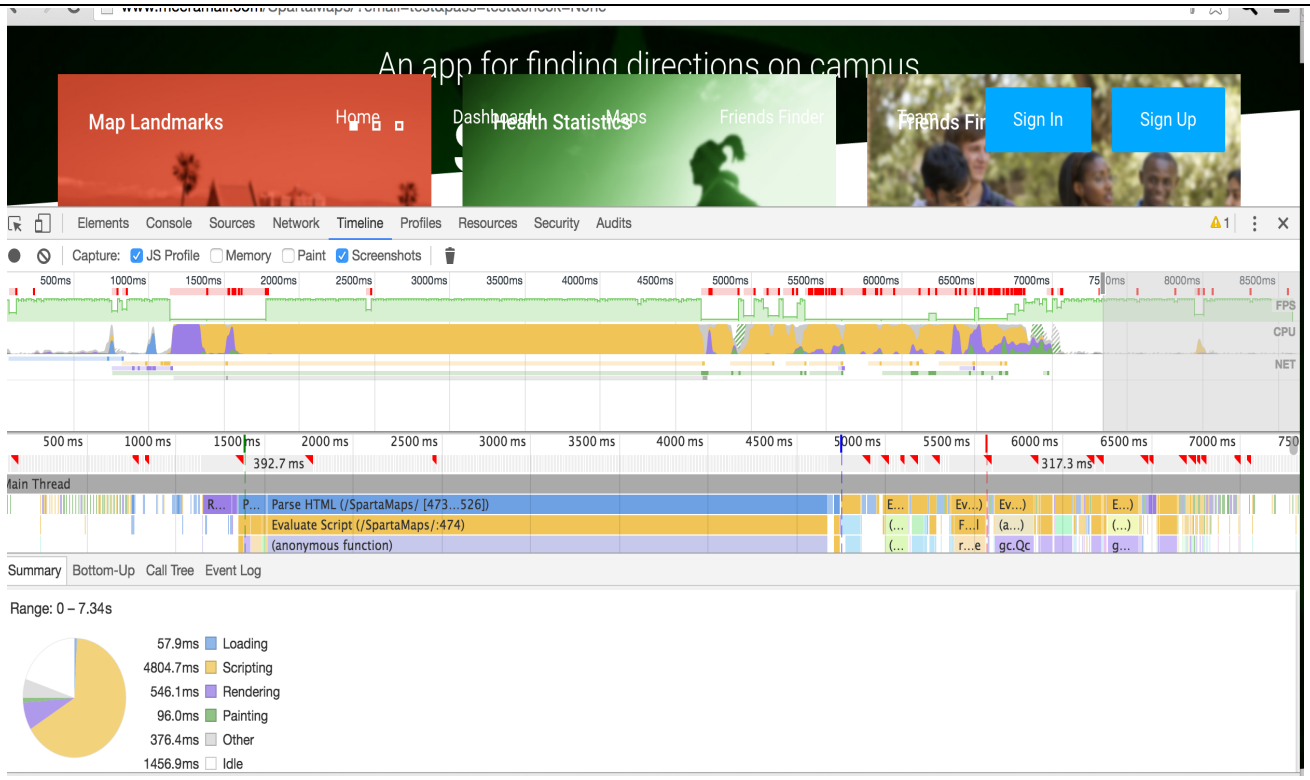

Fig 95: Timeline Panel

Profiles Panel

This panel is used to profile the memory usage and execution time of our web pages.
We can get a better understanding about where the resources are spent which will in turn help to optimize the code.
It has two profilers:

- CPU Profiler
- Heap Profiler
- CSS Selector Profiler - It measure the aggregate time spent on matching selectors to elements in the DOM tree.

CPU Profiler

It shows where exactly execution time is spent in our web page's Javascript functions. Here again we can start , run the website and stop the recording. The Javascript Profile will be collected and can be loaded to analyze. Sample Javascript Profiling for our website:

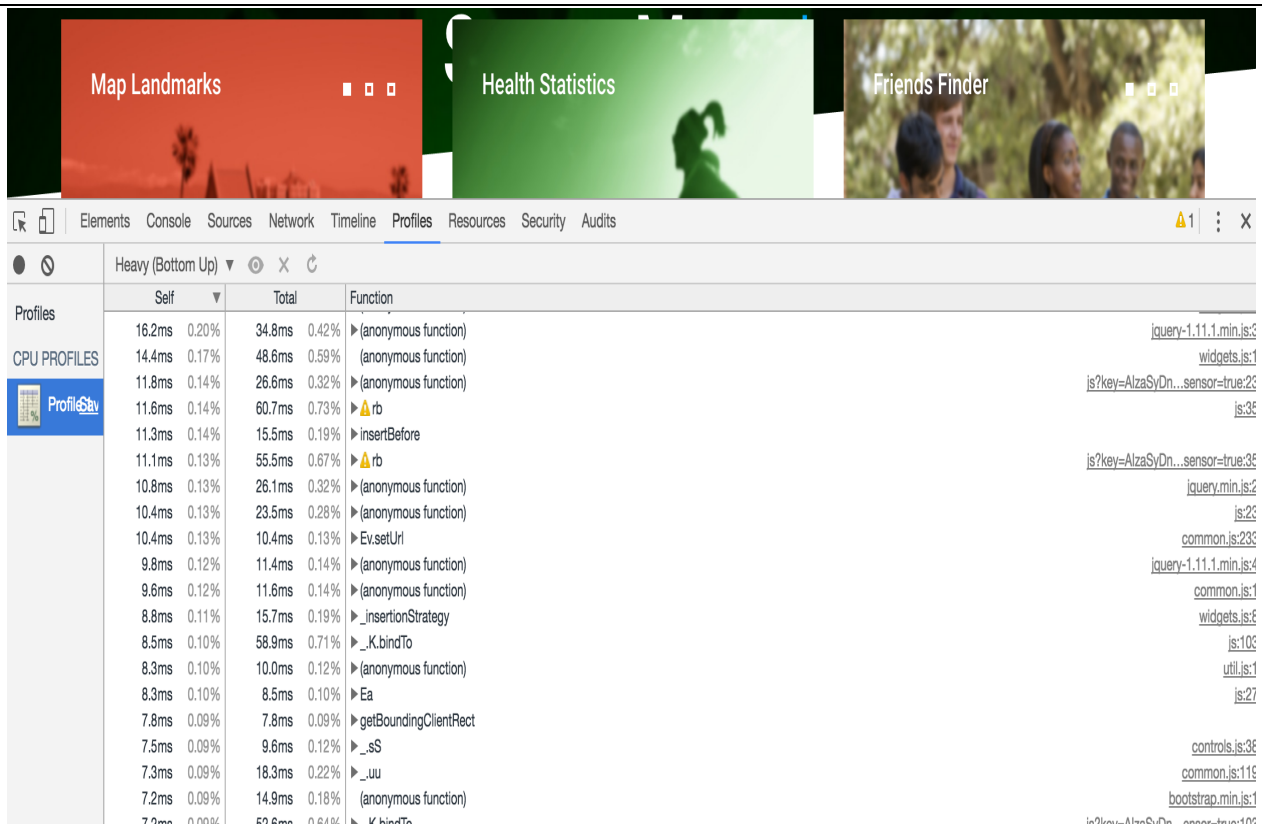

Fig 96: CPU Profiler

Heap Profiler

It shows the memory distribution of our web page's JavaScript objects and the related DOM nodes. For our website:

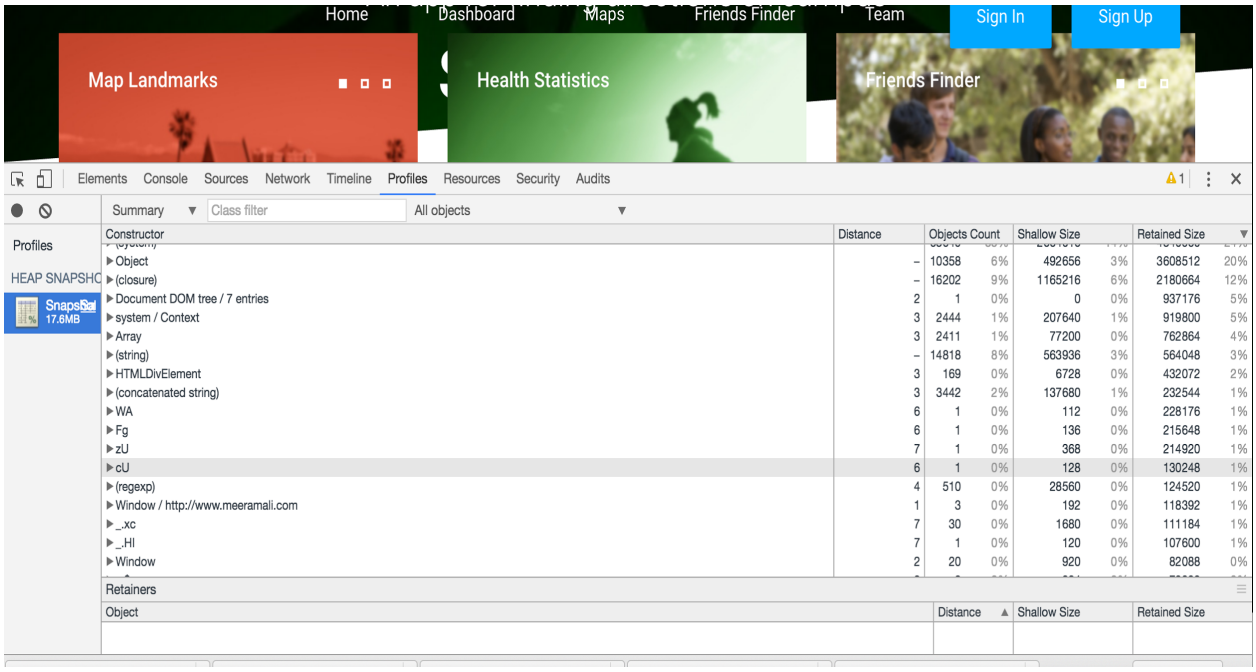

Fig 97: Heap Profiler

Chapter 6

Admin Monitoring – Google Analytics

Google analytics is a webpage monitoring service provided by google that tracks and reports the traffic on the webpage. It is the most widely used analytics service on the net. It provides a SDK for embedding the analytics on the webpage we wish to monitor.

The approach is to show a dashboard for the administrator, which can identify poorly performing, pages, the visitors, the browsers, the duration of the visit, the geographical position. It also helps to show the site's transaction, revenue and many more.

In SpartaMaps, we have used google analytics to track the different analytics and also check Cross browser compatibility and localization.

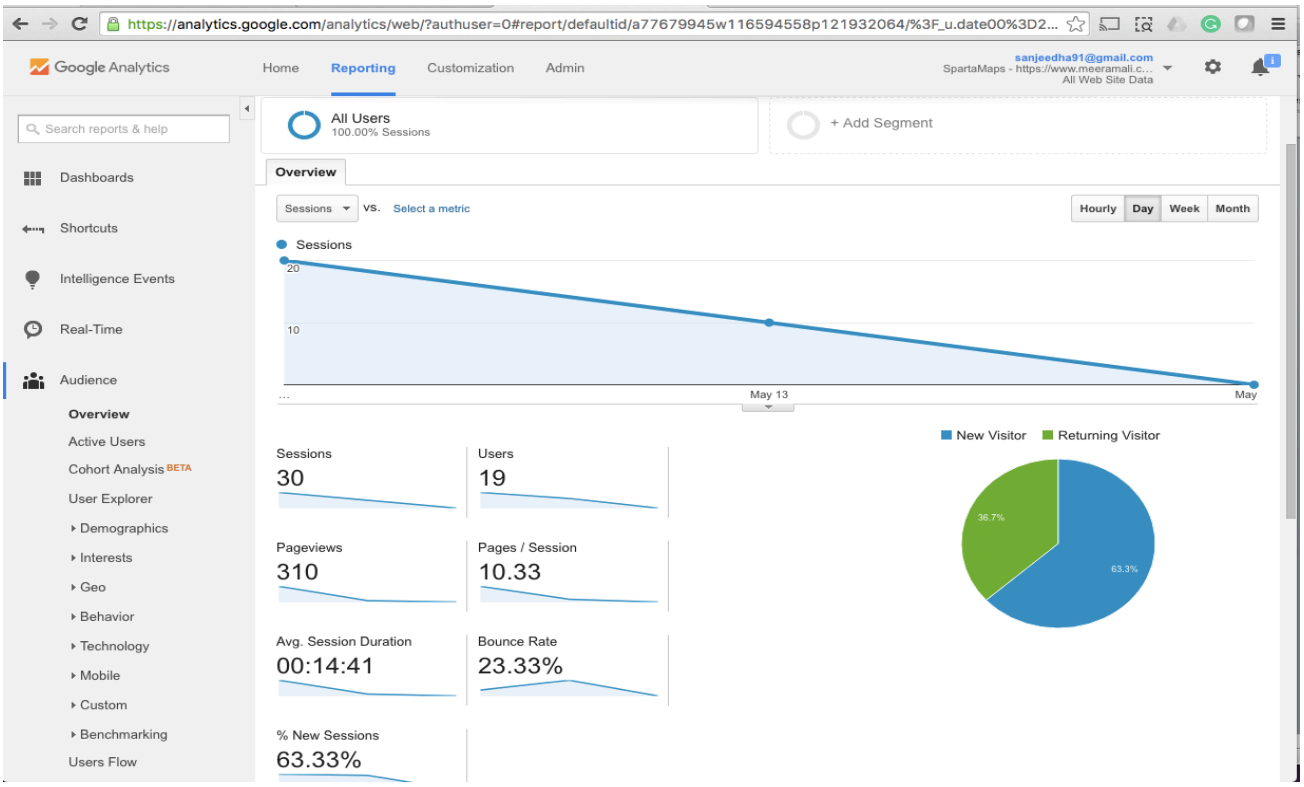

Fig 98: Google Analytics for SpartaMaps

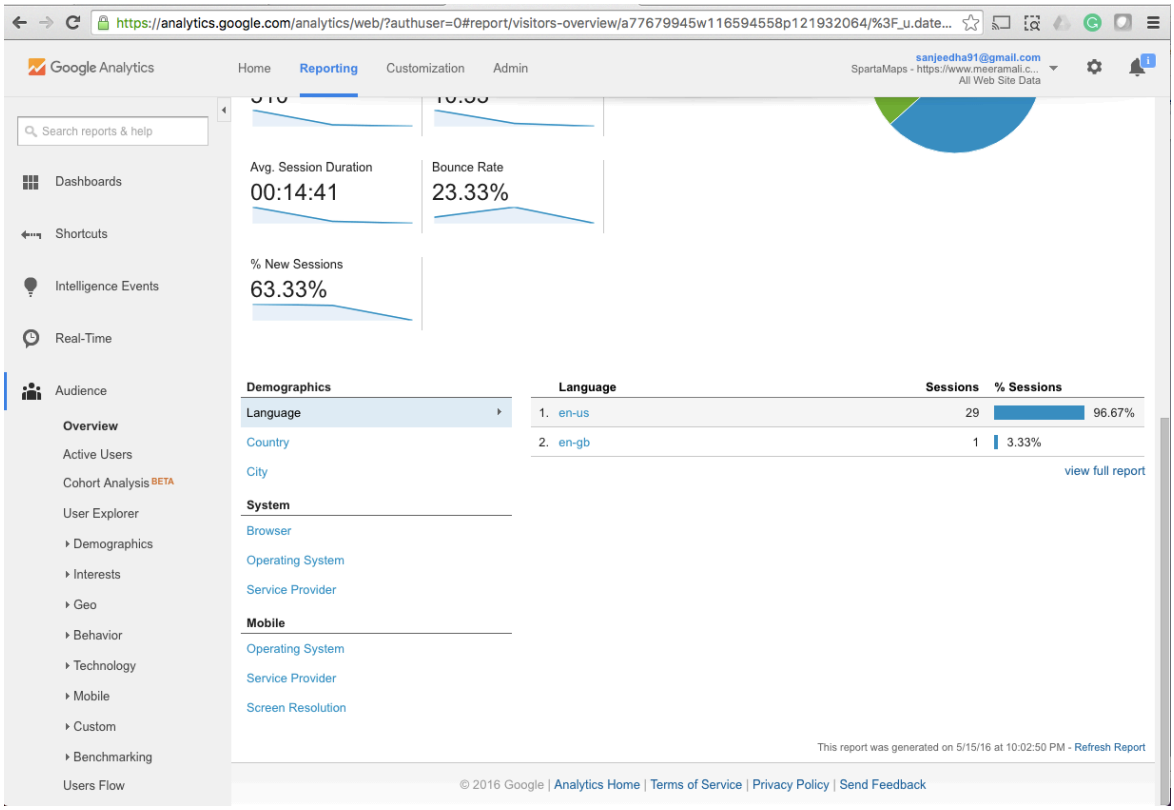

Fig 99: Localization

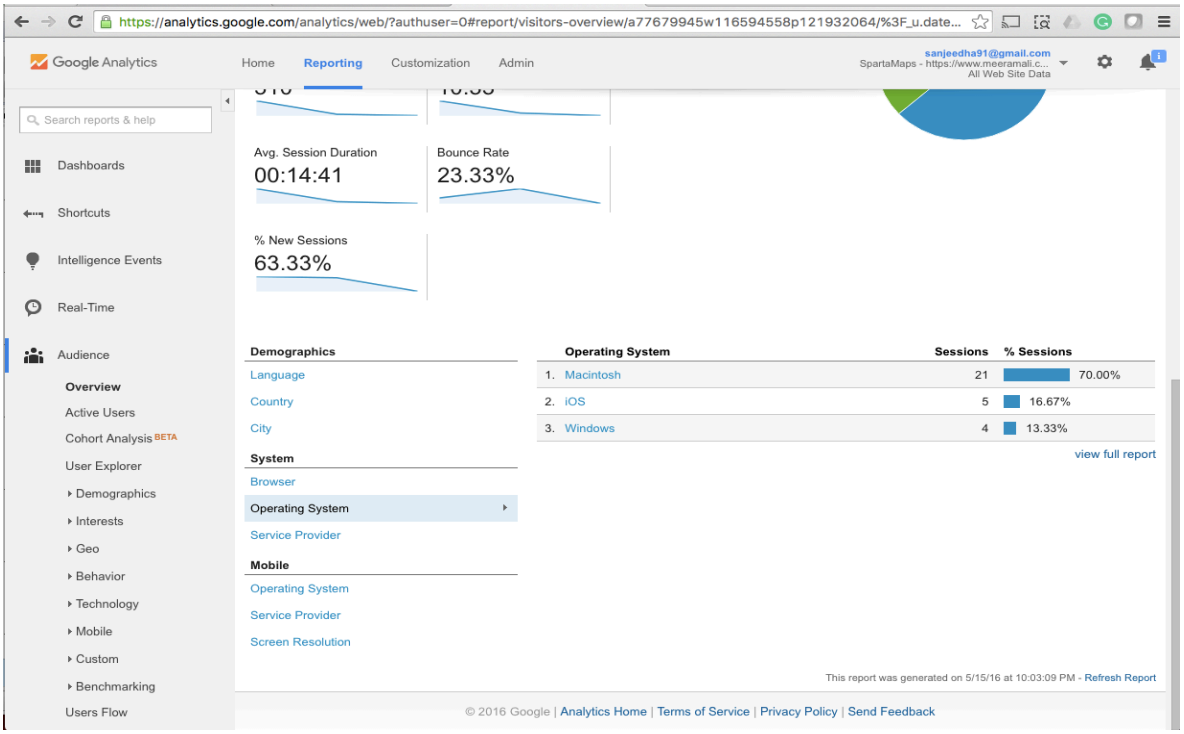

Fig 100: Cross- Browser compatibility

Chapter 7

References

[1] http://www.usabilitybok.org/storyboard

[2] Lee, Newton; Krystina Madej (2012). *Disney Stories: Getting to Digital*. London: Springer Science+Business Media. pp. 55–56. ISBN 9781461421016.

[3] Brown, Dan M. (2011). *Communicating Design: Developing Web Site Documentation for Design and Planning* (2nd ed.). New Riders Press. ISBN 978-0321712462.

[4] Garrett, Jesse James (2010). *The Elements of User Experience: User-Centered Design for the Web and Beyond*. New Riders Press. ISBN 978-0321683687.

[5] Knight, Pamela; Corder, Aaron; Liedel, Ron; Giddens, Jessica; Drake, Ray; Jenkins, Carol; Agarwal, Paul (October 2002). "Evaluation of Run Time Infrastructure (RTI) Implementations" (PDF). *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* (Society for Modeling and Simulation International). Retrieved3 March 2015.

[6] Fujimoto, Richard. "The High Level Architecture: Introduction" (PDF). Retrieved March 3, 2015.

[7] https://en.wikipedia.org/wiki/Pagination

[8] Mikheev, Oleg (28 August 2007). "Ajax programming with Struts 2". Network World, Inc. Retrieved 6 February 2009.

[9] Baptiste, Lyndon (30 November 2007). "Perfect PHP Pagination". SitePoint. Retrieved 6 February 2009.

[10] Gervasio, Alejandro (10 May 2005). "Previous or Next? Paginating Records with PHP – Part 3". Developer Shed. Retrieved 6 February 2009.

[11] Innovative, Php (3 February 2011). "PHP Pagination from Scratch". InnovativePhp. Retrieved 3 February 2011.

[12] http://www.sitepoint.com/understanding-responsive-web-design-cross-browser-compatibility/

[13] http://www.momentech.ca/website/

[14] https://www.tinfoilsecurity.com/blog/cross-browser-development-tips-css

[15] http://www.airccse.org/journal/ijsea/papers/0711ijsea05.pdf

[16] http://www.elated.com/articles/cross-browser-website-tips/

[17] http://techbrij.com/7-tips-to-make-your-website-cross-browser-compatible

[18] http://www.w3schools.com/

[19] http://istqbexamcertification.com/what-are-software-test-types/

[20] https://en.wikipedia.org/wiki/HP_LoadRunner

[21] https://en.wikipedia.org/wiki/Google_Analytics