Principles of Software Engineering 2023

*[FAU: CEN 4010]*

# Project Name

Group: Team 2

??? (Team Lead)

**Team members: Eli Antoine, Garret Bell, Jayde Shaw, John Kantlehner, Manny David**

Milestone 1

Date: 02/20/2023

Revision History

TABLE OF CONTENTS

## (this will be auto-generated in the formatting of the Word doc)

Executive Summary……………………….. #

Competitive Analysis ………………………………… #

Data Definition ………………………………… #

Overview, Scenarios, and Use Cases ………………………………… #

Initial List of High-Level Functional Specifications ………………………………… #

List of Non-Functional Requirements ………………………………… #

High-Level System Architecture ………………………………… #

Team Roles ………………………………… #

Executive Summary

Often, the best vacation experiences often come from insider information - a “local” gave some great advice on where to go and what to avoid. This “local” often also knows you personally, and their advice is tailored for you, and the things you like. Most of us, however, don’t know someone who lives locally in every town we visit, and so we are inevitably left to guess at what would be worthwhile to spend our time on and what wouldn’t. This is where a travel advisor website usually comes in.

There are many travel advisor sites, and they each share a problem - being data rich, but information poor. This means that although the amount of data may be large, if one can not access the data in a meaningful way, it’s not very useful. The question becomes not how to find something to do - but how to find something that you, as an individual, would want to do. People are all different; a great time for one person may be not very enjoyable for another. There is a need for a travel advisor that will take into account the user’s preferences, while at the same time providing the best, most diverse selection of activities and locations to search from. Voluntas Traveler has been developed for just that purpose.

The Voluntas Traveler travel advisor website takes its name from the latin word for “wish” and “good will”, and was developed with the goal to become a more *accurate* advisor for travellers everywhere .Voluntas Traveller provides what others do not by returning relevant results that are tailored to the member’s need, having weighted ratings and a living database that can grow dynamically; as well as a wishlist to store places you find for later reference.

Voluntas Traveler provides the best attractions, events and activities of every kind, where ratings given by locals are differentiated from those who are visiting the area. In this way, users to the site can see what spots locals rate highest, and lowest… and then still get the big picture and also see how visitors from out of town rated it. Users can also add locations and activities, making our database more complete, because it is a live and ever-changing list. Vacations and trips can be planned for in advance, using the wishlist feature, which will save or “bookmark” a list of places you want to visit

In addition to the features tailored to the individual's needs, Voluntas Traveler reduces the time spent on searching, by logically storing data in a way that returns your results faster; while still returning the most relevant results, based on the individual member’s preferences. This results in a travel advisor service that is more accurate, faster, easier to find the results you are looking for.

Voluntas Traveler - a travel advisor website that’s a traveler's wish come true.

Use Cases

## Use Case - Search

The user comes to the page and wants to use the search capabilities to plan a trip.

User arrives at the Home Page and use the search bar field to enter search criteria. The system displays the search results and the user chooses among the selections.

1. **Description:**

Use case describe the process of how the User will utilize the search feature of the system.

1. **Actors**:
   1. User
   2. System

1. **Preconditions**:
   1. User has an active internet connection
   2. System is available

1. **Primary Flow of Events**:
   * 1. User arrives on web page.
     2. User enters search criteria of activity or location into the search bar
     3. Web page displays relevant search results.
     4. User selects Location/Activity among the search results.
     5. Terminate Use Case: Search

1. **Alternate Flows**
   1. **User Enters Prohibited Characters/Format into Search Bar**

If in step 2, user enters prohibited characters or a prohibited format in the search bar. (User enters all numbers or punctuation)

* + - 1. Web site notifies user that the characters they used are prohibited
      2. Return to step 1
  1. **User Enters Unknown Location/Activity**

If in step 2, user enters a location/activity that doesn’t exist in the database.

* + - 1. System performs Use Case: Add a Location
      2. Terminate Use Case: Search

## Use Case - Add a Location/Activity

The User wants to add a missing Location/Activity to the searchable database. The User does a search (See: Use Case Search) and the location/activity does not exist in the database. User will select Add Locations/Activities option, and the system will check to ensure the user is logged in. If not, the user is asked to log in, or create an account if one does not exist. Once the system confirms a successful login, the user will fill out the form provided by the system. User will press submit, and the new Location/Activity is added to the database.

1. **Description:**

This use case describes the process of how the User will add a Location/Activity to the system.

1. **Actors:** 
   1. User
   2. System

1. **Preconditions:** 
   1. User has an active Internet Connection
   2. System is available
   3. User has an active account
   4. User is logged into system

1. **Primary Flow of Events:** 
   1. User has searched (See: Use Case Search) for location/activity not in database
   2. System checks if user is logged in
      1. If user is not logged in, go to Alternative Flow 5.1
   3. User fills out form provided by system
   4. User presses Submit button to save added location/activity to the database
   5. System saves added location/activity to the database
   6. Terminate Use Case - Add Location/Activity

1. **Alternative Flows:** 
   1. User is not logged in (from Primary Flow 4.2.1)
      1. System requests user enters credentials
         1. If user does not have an account, go to Alternative Flow 5.2
      2. User enters credentials
      3. System validates credentials
         1. if credentials are not valid, return to step 5.1.1
      4. End Alternative Flow 5.1 - Return to Primary Flow ,step 4.3

* 1. User does not have an account (from Alternative Flow 5.1.1.1)
     1. System requests the user create an account
        1. If the user does not want to create an account, End

Alternate Flow 5.2, return to Primary Flow 4.6

* + 1. User clicks CREATE ACCOUNT button
    2. System displays Account Creation Form
    3. User completes form, and presses submit
    4. System creates new member account in the member database
    5. End Alternative Flow - return to Primary Flow, step 4.3

## Use Case - Comment

The user comes to the page and wants to add a comment about a recent activity or location they visited. User arrives at the Home Page and logs in. User then performs Use Case: Search to find a specific Location/Activity. User selects an Location/Activity. User rates Location/Activity. User clicks on the Comment button. User enters a comment and hits submit button.

1. **Description**:

Use Case describes the process of how the user will post a comment into the system for other users to view.

1. **Actors**:
   1. User
   2. System

1. **Preconditions**:
   1. User has an active internet connection
   2. System is available
   3. User has an active account
   4. User is logged into system
   5. User has rated the Location/Activity

1. **Primary Flow of Events:** 
   * 1. User arrives at Home Page.
     2. User logs into their account.
     3. User performs Use Case: Search or Use Case: Browse
     4. User rates Location/Activity
     5. User clicks on Comment button.
     6. Web page displays field to enter comment.
     7. User enters comment and clicks submit button.
     8. System thanks the user for the comment.
     9. Terminate Use Case: Comment

1. **Alternate Flows**:
   1. **User Isn’t Logged In**

If user does not perform step 2 and they reach step 4

* + - 1. Web page requests user to either log in or create an account.
      2. User logs in or user creates an account
      3. Return to step 4
  1. **User Tries to Comment Without Rating First** If user skips to step 5 without doing step 4 first 1. Web page requests that user rate first.
     + 1. User rates.
       2. Return to step 5
  2. **User Exceeds Comment Maximum Length**

If in step 7, user types over the maximum length of the comment.

* + - 1. Webpage informs user that their comment has exceeded the maximum length.
      2. User shortens their comment.
      3. Return to Step 8

## Use Case - Rate a Location/Activity from Home Page

The user comes to the page and wants to quickly rate an activity or location they visited. User will arrive on system and will attempt to rate location and/or activity located on the home page’s activity listing. Web page will ask user to log in or create an account to rate then will allow user to add a rating of 1 to 5 stars to a Location/Activity.

1. **Description:**

Use Case describes the process of how the User will rate a Location/Activity from the home page. See Use Case: Rate a Location/Activity from Information Page for alternate method of rating activities.

1. **Actors:** 
   1. User
   2. System

1. **Preconditions:** 
   1. User has an active internet connection
   2. System is available
   3. User has an active account
   4. User is logged into system

1. **Primary flow of events:** 
   * 1. User arrives on web page.
     2. User attempts to rate Location/Activity

2a. If user isn’t logged in, go to Alternative Flow 5.1 3. User rates Location/Activity with 1-5 stars.

4. Terminate Use Case - Rate a Location/Activity from Home Page

1. **Alternative Flows:** 
   1. **User isn’t logged in**

If in step 2, user attempts to rate and isn’t logged in.

* + - 1. Web page displays a prompt asking user to log in or create an account.
      2. User logs in or creates a new account.
      3. Return to step 3.

## Use Case - Rate a Location/Activity from Information Page

The user comes to the page and wants to quickly rate an activity or location they visited. User will arrive on system and will do a search for a location and/or activity. Once they arrive to the Information Page, they click on the rating graphics to rate it from 1-5 stars.

1. **Description:**

Use Case describes the process of how the User will rate a Location/Activity from the Information Page. For information on rating from the home page, see Use Case:

Rate a Location/Activity from Home Page

1. **Actors:** 
   1. User
   2. System

1. **Preconditions:** 
   1. User has an active internet connection
   2. System is available
   3. User has an active account
   4. User is logged into system

1. **Primary flow of events:** 
   * 1. User arrives on web page.
     2. User does a search for a location and/or activity.
     3. User attempts to rate Location/Activity

4a. If user isn’t logged in, go to Alternative Flow 5.1 4. User rates Location/Activity with 1-5 stars.

1. Terminate Use Case - Rate a Location/Activity from Information Page

1. **Alternative Flows:**

5.1 **User isn’t logged in**

If in step 4, user attempts to rate and isn’t logged in.

* 1. Web page displays a prompt asking user to log in or create an account.
  2. User logs in or creates a new account.
  3. Return to step 5.

Data Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Meaning** | **Usage** | **Comment** |
| Visitor | actor | Use Case scenarios, | this person is not familiar with the area of reference |
| Local | actor | Use Case scenarios | This person lives in or near the area of reference |
| User | actor | Use Case scenarios | General definiton for both visitor and local |
| Member | actor | Use Case Scenarios | A user who is registered with the system. |
| Non-Member | actor | Use Case Scenarios | A user who is not registered with the system. |
| Location/Activity | data | Searchable places or activities | Represents a hotel, restaurant, attraction or place that is searchable and allows the user to rate it. |
| Restaurant | data | Activity Type | One of the services we let users add and rate |
| Hotel | data | Activity Type | One of the services we let users add and rate |
| Attractions | data | Activity Type | One of the services we let users add and rate |
| Account | data | Use Case scenarios | Store users information |
| Recommend | data | Use Case scenarios | Store the highest rating activities |

Data Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Meaning** | **Usage** | **Comment** |
| Newset | data | Use Case scenarios | Store the newest adding activities |
| Rating | service | Site user service | Allows visitor or local to rate different services/attractions |
| Comments | service | Site user service | Allows users to leave their opinions |
| Navigator | service | Site user service | Helps visitor or local to navigate web page easier |
| Photos | service | Site user service | Allows visitor or local to post pictures |
| Search | service | Site user service | Allow user find location/activities |
| Voluntas List | service | Site user service | Allows users to add  their favorite activities |
| Filter | service | Site user service | help user search more clearly by giving specific options |
| log in | service | Site user service | Allow user have ability to use comment, rating, photos, and Voluntas  List activities |
| Web Site | User Interface | User interface | Front end display for user interaction |
| Home page | User Interface | User interface | The first page that a user go |

Data Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Meaning** | **Usage** | **Comment** |
| Information Page | User interface | User interface | Page that is displayed when an activity is clicked. This page displays photos, comments, etc. |
| System | platform hardware and services | Use-case scenarios | The mySQL database, all code, front end design and back end supporting services. |
| Voluntas Traveller | Domain Name | Use Case scenarios | A real usable name that represent all web  pages and the web  site |
| sfsuswe.com | production server | Use Case Scenarios | It’s a server that store all of our data |

Initial list of functional specs

## Non-Member expectation

### 1. Creating Account

○ **1.1** The system shall allow the user to create an account by storing

UserID, Password, Date of Birth, First Name, Last name, Location, and answer to security question/phone number.

The system shall not allow the User to Create an account if the UserID choose by the User already exist in the System’s Database Also the system shall prevent the user from creating an account if the User’s chosen password does not match the re-enter password field. System shall prevent the creation of the user’s account following fields is not filled. Fields that have to be filled are First Name, Last Name, Location, UserID, Password, Re-enter Password, Security Answer Security Question or Phone number, and Date of Birth.

### ○ 1.2 Stimulus/Response Sequence

i. User enters a UserID (same as email) ii. User enters a Password iii. User re-enters Password for confirmation iv. User shall enter their First and Last Name v. User shall enter their Location vi. User shall enter their date of birth

1. User shall provide to an answer to given Security Question Or provide their phone number
2. System shall check if UserID is available ix. System shall validate Password
3. System shall store user Name, date of birth, and answer to selected security question/phone number
4. System shall confirm that the account was created to the User xii. System will have a button to redirect the user back to the home page at will

○ **1.3 Function requirement label**

i. REQ 1.1 Creating Account

### 2. Browse the Location/Activities

○ **2.1** User browse by selecting Location/Activity from a list in Popular Picks The system shall not let the user to filter out the Popular picks list that has been provided to the user because it will be filtered already. The user will

not be able to add to the popular list picks because it will be listing the top 20 rated Location/Activity and will be changing with ratings.

### ○ 2.2 Stimulus/Response Sequence - Popular Picks

1. User scrolls to the Popular Picks
2. System shall have a filtered list by popularity for the User ○ **2.3** User will browse the system by entering in a search by Geographic Location or Name.

The System shall prevent the user from deleting Locations/Activities from the list that the user filters out. the system shall prevent the user from searching Location/Activity in a different region once user has filtered out the list. The user has to hit new search.

### ○ 2.4 Stimulus/Response Sequence - Search

1. User enter search criteria( geographic Location or Name of cities) into the search
2. System shall supply the user with a list of activities and tourist attraction within the region
3. System shall have a button that will allow the user to return to the home page

### ○ 2.5 Function requirement label

i. REQ 2.1 Browse by Popular Picks ii. REQ 2.3 Browse by Search

### 3. View the Specific Location/Activity

○ **3.1** The system has provided a list of tourist attraction and activities within a region. The User has the option to click on a specific attraction/activity view specific facts about particular attraction/activity.The system shall not allow the user to edit the information for a specific Location/Activity on the information page.

### ○ 3.2 Stimulus/Response Sequence

i. User clicks on an attraction/activity among the list ii. System shall provide specific facts about the item

iii. System shall have a button to allow the user to return to the region list

○ **3.3 Function requirement label**

i. REQ 3.1 view selected item on list

### 4. About Us

○ **4.1** System provides information to the user about the functionality that

can benefit prospective user with links that will direct the user to specific topics and button on the bottom to bring the user back to the topics. The system shall not allow the user to erase any information about the system and how the system functions for the user.

### ○ 4.2 Stimulus/Responsive Sequence

i. User uses the navigation bar and clicks “About us” ii. User is redirected to About us page iii. User sees links about topics at the top of the page iv. User clicks a topic and get redirected to that topic v. User after reading topic clicks “back to top”

1. User redirected to top of the page with links and repeated until satisfy
2. System has a button that allows the User to return back to Home page

○ **4.3 Function requirement label**

i. REQ 4.1 AboutUs

## Members expectations

### 5. Edit Profile

○ **5.1** User shall be able to edit their profile by providing a name, date of birth, and a brief summary about themselves and the System shall store name, date of birth, and summary when the user clicks the save button.The System shall prevent any changes to the User’s profile if any of the fields are left blank. The user has type the information in a valid format in order for the system to store the information.

### ○ 5.2 Stimulus/Responsive Sequence

1. User will navigate to “my account” and a drop down menu will be activated
2. User will click on edit profile
3. User will input their name, date of birth, and write a summary about themselves
4. User shall click save
5. System shall store their name, date of birth, and summary vi. System shall refresh to user profile with updated information as a confirmation

vii. System will have a button to redirect the user back to the home page at will

○ **5.3 Function requirement label**

i. REQ 5.1 Edit Profile store

### 6. Rating Locations/Activities

○ **6.1** User will navigate to specific Location/Activity through the browse functions and have the ability to rate certain Location/Activity. System shall provide a 5 star(or other thing) scale that User will use, 5 being the best

The system shall prevent the user from rating more than 5 or less than 0

### ○ 6.2 Stimulus/Responsive Sequence

i. User will use the browse function to navigate to specific

Location/Activity ii. User will click on Location/Activity of choice

1. User will scroll to star rating feature and highlight the amount of stars the User wish to rate.
2. System shall store the user rating score
3. System will prevent user from rating again
4. System shall display a gratitude pop up stating that the rating has been stored
5. System will have a button to redirect the user back to the home page at will

○ **6.3 Function requirement label**

i. REQ 6.1 Rating Scale

### 7. Comment about Locations/Activities

○ **7.1** The system shall have a comment box where user will have the liberty to state their experience of the location/activity.The systems will prevent the user from commenting if the user have not created an account and have not given the Location/Activity a rate. Users are suppose to be members and have to rate the Location/Activity they wish to comment.

### ○ 7.2 Stimulus/Responsive Sequence

i. User shall use the browsing function ii. User will navigate to desired Location/Activity

iii. User shall scroll to the comment box at the bottom of the page iv. User shall type their experience via keyboard v. User shall click “Post Comment” vi. System shall store User’s comment

vii. System shall display User’s Comment at bottom of the screen with a sorting of newest to oldest order as a confirmation viii. System will have a button to redirect the user back to the home page at will

○ **7.3 Function requirement label**

i. REQ 7.1 Post Comment

### 8. Rate comments posted by users

○ **8.1** User will navigate to a specific Location/Activity they will read the facts and read comments provided by users. Some the user will find useful and up-vote the comment others the users will find unuseful and downvote the comment. System shall keep track of these upvotes and downvotes.

The system prevent the user from deleting upvotes or downvotes. Once an Upvote or Downvote has been past it stays. The system shall prevent the User from giving multiple Upvotes and Downvotes on a specific location.

### ○ 8.2 Stimulus/Responsive Sequence

i. User will use the browsing features to view list of

Locations/Activities ii. User will click on desired Location/Activity among the list of suggestions

iii. User will read facts and then scroll to Users comments iv. User will upvote comments that they find useful and downvote comments they find unuseful

1. System shall record upvote or downvote on specific comment
2. System shall prevent the user from upvoting and downvoting again on same comment
3. System shall show a comfirmation that the user have voted on the comment.
4. System will have a button to redirect the user back to the home page at will

○ **8.3 Function requirement label**

i. REQ 8.1 Upvote DownVote

### 9. User Add location/Activity

○ **9.1** User will be allowed to add Location/Activity to the database if, and only, if Location/Activity is not found in the database. The system shall store the given information at the moment the User clicks on “add” button. System shall prevent Users from Deleting Location/Activity from the Database.

### ○ 9.2 Stimulus/Responsive Sequence

1. User shall navigate the browse function
2. User shall navigate to desired location/Activity and realise such location/Activity is not found. iii. System shall prompt user,”Do you wish to add Location/Activity” iv. User shall hit yes

v. System shall redirect User to add Location/Activity vi. User shall enter required information and upload a picture vii. System shall store Location/Activity once user clicks Add viii. System shall thank the user and redirect User to home page

○ **9.3 Function requirement label**

i. REQ 9.1 User add

### 10. Contact Customer Service

○ **10.1** User will be allowed to contact the developers via email. user shall type in their name, email, and their query. System shall store these fields and submit it once the user click the send button. System shall prevent the User from contacting developers directly. User have to submit a ticket and the ticket shall be redirected to the correct personnel.

### ○ 10.2 Stimulus/Responsive Sequence

i. User shall navigate to Contact Us among the navigation bar ii. User shall fill in Name, Email, and type up their Query iii. User shall then click “Send” button underneither the Query box iv. System shall store the information and submit

v. System shall state that the information was sent and thank the user vi. System shall a button to redirect the user to Home page

### ○ 10.3 Function requirement label

i. REQ 10.1 Contact Us

List of non-functional specs

**Performance Requirements**:

1. *Responsiveness*: The system will also be responsive, operating on various monitor sizes, ranging from 10” netbooks to 24” desktop monitors. It will also be responsive with a wide variety of resolutions, from 1024 x 600 through 1900 x 1200.
2. *Cycle Time*: The cycle time at expected performance will be 1.0. With this in mind, the system will operate between 1.0 - 1.2 with a load of 5-10 concurrent users or a slight lag. The system will operate with a 1.21 - 1.30 with a load of 11-25 concurrent users or a moderate lag. The system will operate with a 1.31-1.50 with a load of 26-45 concurrent users or a heavy lag. Finally the system will operate with a 1.51-1.70 with a load of 46-50 concurrent users or a very heavy lag. Any number of concurrent users over 50 will cause the systems performance to halt briefly until a user finishes.
3. *Speed Per Transaction*: The speed per transaction will be between 20-100 milliseconds, depending on the cycle time. The system will process 10-50 transactions per second.
4. *Test Requirements*: the test requirements for performance will include an expected load test as well as testing on all of the functional specs listed and their speed per transaction.
5. *Reliability*: The mean time between failures is that it must have 1 hour or less of downtime in a total of 3 months. This downtime can be used to perform maintenance and update information. The system should be operational 100% of the time for the first 99.8% of the calendar for the first year of its operation.
6. *Minimum Bug Counts*:

● No more than 5 bugs in the system during integration and testing. ● No more than 3 bugs can remain in the system after delivery.

1. *Execution Speed*: Execution speed of the initial home page on a high-speed internet connection should load within 100-200 milliseconds, depending on the current cycle time.
2. *Storage Utilization*: Storage utilization should be within 75 - 90% of the available storage provided at the time as to not get too close to using all storage and having a technical issue if more storage is needed for an emergency situation.
3. *Robustness*: The time needed to restart after a failure should be under an hour. The percentage of events that cause a failure will be under 0.1%. The probability of the data being corrupted on failure must be below 0.8%.

**Ease of Use:**

1. *Training Time*: Training time should be minimal if any at all as our site will be very user-friendly and should be able to be used by anyone who visits it.

**Interoperability Requirements:**

1. *Browser Compatibility*: The system will be a web-based web app that operates on at least two of the all of the major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. It will have functionality in it that will provide alternatives if the browser does not have JavaScript installed on it.

3. *Computer and OS Compatibility*:The system will operate on various types of operating systems, including Windows, OS X and Linux. It will also operate on any type of computer which can run a browser which is supported.

**Expected Load**:

1. We will make allowances for up to 50 users at the same time. Load testing will be done to record performance times during periods of high traffic, both in a continuous and in a spiked pattern.

**Security Requirements**:

1. *Login/Password System*: Our system will have a login/password system to maintain the preferences, ratings and reviews of our sites visitors and locals. This implementation will also require a password confirmation upon creation. We will also ask the user for a security question that we will store along with the answer and if the user forgets their password, they will be able to retrieve it by providing the answer.
2. *Encryption*: The website will not be encrypted as no purchases or exchange of valuable information, such as credit card numbers, will occur.
3. *Access Control*: The ability to edit the front end code and back end code and databases will be provided to everyone on the development team. The users and visitors will have a limited access to using the system based on the user interface.
4. *Spam Protection*: The site will ask the user to enter a string of characters shown on a picture to create an account, thereby preventing bots from spamming the site and creating bogus accounts.
5. *Resource Utilization*: Resources such as the mySQL database on the sfsuswe.com server will be access through the PHP code using the usernames and passwords therein. All access to the sfsuswe.com servers and their resources will be obtained with the usernames and passwords given. The system will utilize bootstrap and jQuery as frameworks and will document proper licenses and/or qualifications of each.

**Portability Requirements**:

1. *Platform Compatibility:* The system has plans to be developed for mobile and tablet, but its based on whether it will be able to be developed in the given time frame.

Its possible that in a further version, mobile and tablet compatibility will be implemented.

1. *Percentage of Target-Dependent Statements*: In this version of the system, 0% of statements will be target dependent. In a future version for use on mobile and tablet, 30% of statements will be target dependent.

**Supportability Requirements**:

1. *Coding Standards*: Our system will be coded in a range of 75-80% of coding standards for HTML 5 and CSS3. The code will be produced, reviewed, tested, reviewed again and finalized by yet another developer for efficiency.
2. *Naming Conventions*: HTML classes and id tags will be coded in lowercase except in the situation where there is more than one word in a name, and then it shall use camelcase convention. SQL tables and data names will be agreed on by the team and always use first letter capitalization and lowercase remaining letters.

**Storage Requirements**:

1. The storage for our system will consist of the sfsuswe.com server holding our mySQL databases within an unknown capacity as well holding our files for the actual site.

**Survivability:**

1. Our system will be stored on a sfsuswe.com servers and the major files and documentation will be backed up on Google Drive to prevent loss in case of a fire or other physical destruction of the servers.

**Availability Requirements:**

1. *Accessible Times*: Our system should be available for use 24 hours a day, 7 days a week. It will be up and running as long as the sfsuswe.com server is available.
2. *Downtime Impact*: The down time will be minimal but when it is necessary, a splash page will be used to identify that the system is in maintainence. Downtime impact is expected to be minimal and the scheduled downtime will be announced ahead of time.
3. *Support*: There will be support availability by email that will filter to several assigned developers. They will be responsive within 24 hours.

**Fault Tolerance:**

1. *Exception Handling*: There will be exception handling provided on all situations where an exception could occur. This will provide the user with an explanation as to why an exception occured and give them a chance to either input the correct answer or they will be taken back to the home page.
2. Self-checking software will not be implemented as this system isn’t critical enough to warrant it.

Competitive analysis

The analysis of competitors’ web sites will focus on six main features (homepage, design, navigation, search, content, usability) and three additional features

(ask a local, wish list, add item). The competitive analysis will utilize a numerical scale (1=bad, 2=poor, 3=fair, 4=good, 5=outstanding) and consists of five web sites chosen for their focus on travel, social networking, and use of rating/review systems.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Voluntas Traveler** | **Trip Advisor** | **Virtual Tourist** | **Zoover** | **Not**  **Seen**  **World** | **Tripbod** |
| **Homepage:** | 5 | 4 | 5 | 2 | 2 | 3 |
| **Design:** | 5 | 5 | 4 | 3 | 3 | 3 |
| **Navigation:** | 5 | 5 | 5 | 4 | 3 | 3 |
| **Search:** | 4 | 5 | 4 | 3 | 3 | 4 |
| **Content:** | 4 | 5 | 5 | 3 | 3 | 3 |
| **Usability:** | 5 | 5 | 5 | 3 | 1 | 3 |
| **Ask a Local:** | 5 | 0 | 0 | 0 | 0 | 4 |
| **Wish list:** | 5 | 4 | 0 | 0 | 0 | 0 |
| **Add Item:** | 5 | 4 | 3 | 3 | 0 | 4 |
| **Mean:** | **4.70** | **4.10** | **3.54** | **2.30** | **1.89** | **3.00** |

**Voluntas Traveler (4.7)** http://www.sfsuswe.com/~f13g31/P1/index.php

The Voluntas Traveler homepage is simple with eye catching photos and a prominent search bar. The navigation is consistent across the site and uses a breadcrumbs feature. The search is quick and utilizes autocomplete. The content is focused on travel ratings and reviews. The site is very user-friendly. All of the additional features are implemented on this site including ask a local, wishlist, and add item.

**Trip Advisor (4.10)** http://www.tripadvisor.com/

Trip Advisor has a very informative homepage that loaded quickly. The design of the page is clean, simple, and mostly text with limited photos/multimedia. The search is centered prominently on the homepage. The main navigation is consistent across the site and easy to use. The search responds quickly to queries and utilizes autocomplete. The site content is focuses on travel reviews and popular places. It is user friendly. It does not have an “ask a local” feature, but it does have a feature similar to a wish list and the ability to add a location.

**Virtual Tourist (3.54)** http://www.virtualtourist.com/

The virtual tourist homepage features large images and a large search bar. The navigation is consistent and easy to use. The search lags a little when entering queries and utilizes autocomplete. The content is focused on serving as a travel guide and offering travel reviews. The site is easy to use. It does not have an “ask a local” feature or wish list, but it does have the ability to add a new item for review.

**Zoover (2.30)** http://www.zoover.com/

The Zoover homepage is not eye-catching. The design of the results page is not impressive. The navigation is consistent. The search responds quickly but does not utilize autocomplete. The content is focused on providing vacation reviews. The usability is average based on the search and design. It does not have a “ask a local” feature or wish list feature, but it does have the ability to add an item for review.

**Not Seen World (1.89)** https://www.notseenworld.com/

The homepage for Not Seen World asks you to sign up or login before you can use their site. When trying to sign up for an account, I am unable to change my birth year from 2013. This causes an error message to be displayed that I am too young to register. The search bar is displayed on the registration page so I could access content anyway. The design and usability of the site are poor. It does not utilize any of the additional features.

**Tripbod(3.00)** http://www.tripbod.com/

The homepage for Tripbod is simple, but it is not clear where to go to sign up for an account. The site navigation is consistent. The search is quick and utilizes autocomplete. The content is focused on tourism and travel services. It has the “ask a local” feature and ability to add a new reviewable item. It does not have a wish list.

**Planned advantages:**

Voluntas Traveler is all about quality. Many travel sites overwhelm the user with a large amount of data, but Voluntas Traveler is focused on delivering quality over quantity. Unlike some travel sites, both registered and unregistered users can easily search the Voluntas site. Joining Voluntas Traveler is as simple as providing an email and password. Registered Voluntas users will receive the best and most relevant data based on their preferences. Registered users have the ability up or down vote other registered users. This serves as an excellent moderation tool that many other travel sites lack. Registered users will also enjoy planning future trips and activities utilizing our VoluntasList—Voluntas’ version of a travel wishlist. Very few travel sites offer this feature. Ask a local is another great feature that sets Voluntas Traveler apart from the competition. Our users will be privy to all of the hidden gems only locals would know about. For users seeking quality content, simple interface, and a place to share travel experiences Voluntas Traveler is the answer.

High-level system architecture

1. **sfsuswe.com Lamp Server**: sfsuswe Lamp Server with Amazon Web Services, shall be hosting our Fall 2013 web development project for a duration after the semester has been completed.

1. **Squirrelmail Web Mail Application**: Squrrel mail web mail application is the means that the group shall be communicating with each other for the development of the project during the semester. All communication has to be through squirrel mail

1. **Bugzilla:** Bugzilla is a bug tracking system used to track project features, issues and bugs, the group will use this tool during the semester to track the development process for the final project

1. **sfsuswe Discussion Forums**: sfsuswe discussion forums will be used to check if instructors have posted any information that was not sent through squirrel mail web application

1. **mySQL Database**: MySQL database is the database that is being used for the data that will be handled for the project. Users will be adding to the database via the website input function and developers will be managing the data (by either deleting) or adding items from and to the database.

1. **Netbeans Integrated Development Environment (IDE)**: Netbeans is the IDE that the developers will be using to create the code for the website Languages to be used for the development of the website will be the following
   1. Hyper Text Mark-up Language(HTML) - will be the language that will allow the browser display the website
   2. Cascading Style Sheets(CSS) - will be the language used to decor the web pages
   3. Personal Home Page (PHP) - will be the language used for server side functionality for the database and real time edits in the tables
   4. Javascript - will be the language used for client side functionality that will be handled for User Interface(UI) needs to make the user experience enjoyable
   5. JQuery - will be the language used for client side functionality (Link to

License: https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt) ---

( Jquery APIs)

* 1. JQuery UI - will be the language used for client side functionality that will be handled for User Interface(UI) needs to make the user experience enjoyable (Link to Lisence: https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt) --- Note

(Jquery APIs)

* 1. Bootstrap - Bootstrap will be the framework that will be used for code construction for web pages within the groups project. (Link to License:

#### https://github.com/twbs/bootstrap/blob/master/LICENSE)

1. **Subversion**: is a tool that the developers will be using to store, share, and allow access for team members to view source code that they have written. It allows for easier version control of the project.

1. **Browser Compatibility**: The system will be a web-based web app that operates on at least two of the all of the major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. It will have functionality in it that will provide alternatives if the browser does not have JavaScript installed on it.

**Team Roles:**

**Team Leader, Front & Back End Developer:**

**Front End Developers:**

* **SVN Repository Admin -**
* **UNIX Admin -**

**Back End Developers**

* **MySQL Admin -**