Assignment 4 - Software Prototype

pARK

Authors: Jacob Leggatt, Gordon McMahon

HOW TO RUN:

Scenario 1: Using Mozilla Firefox open Login.html
Scenario 2: Using Mozilla Firefox open Logn.html
Scenario 3: Using Mozilla Firefox open Login.html

Notes: While checking for the ability to complete the three scenarios please notice that all users share the name kittenloving username intentionally.

Addressed Issues from Assignments 2, and 3:

Problem Statement: It may not be clear to the user that the Quickfind option exists, and if it does where they can find it.if the user doesn't see it it might as well not exist for them.

Solution: Incorporate the Quickfind option into the overhead tab system to allow for visual recognition and ease of access.

Problem Summary: On the Search Tab the Advanced Search option is not clear. Also it is not clear that clicking on the activity makes a popup appear with more options.

Solution: The items of advanced search are now automatically shown so the user can see them and use them.

Problem Summary: The Add to favourites option does not properly provide feedback to the user telling them that the action has been completed successfully.

Solution: A small popup should appear to announce to the user the success of the addition to their favourites.

Problem: It may be difficult to tell that entries in activity lists can be clicked to get details.

Solution: This violates Norman's principle of affordance. Entries in a list do not afford clicking, they afford reading. Buttons and hyperlinks afford clicking. It is recommended that the name area of the lists be formatted to look like a hyperlink.

Problem: Itinerary and schedule are used interchangeably throughout the system.

Solution: This violates the principle of consistency. Users may be confused and believe that two different things are being referred to, and become annoyed when they cannot find the second. It is recommended that one word be chosen and used consistently.

Scenarios

Scenario 1: Steve is planning to take his longtime girlfriend on a day of adventures for their anniversary where money is no object, but he has no ideas. He wants to avoid travelling too far because that isn't romantic. The application must remind him o this anniversary so that he can find things and plan a day. After finding activitie he can add items to his itinerary to create a day plan. **Explanation:** This will be the most common, and most important use of our application. If this doesn't work nothing else matters. The user must be able to plug in information about their requirements "Price, Range, Activity level, mode of transportation, etc" and find both a suggested day and the ability to mix and match suggestion to create their own day. Then they must be able to add the to the itinerary.
Scenario 2: Jamie has fallen in love with a restaurant and wants to save a preference for a particular restaurant. **Explanation:** This scenario encapsulates one of the more difficult ideas of our system. The user must be able to navigate to the desired restaurant, and save a preference for it. That necessitates a unique user account, and a unique listing for the restaurant.
Scenario 3: Jamie wants to quickly find a single activity nearby to pass an hour. She hits th quickfind button, and the application finds his current location and checks her, profile if she has one, to find preferences for activity types and suggests something close to him, and adds the activity to her itinerary. **Explanation:** Finding an activity, nearby and quickly is of great importance for one of our two main types of personas: the tourist. While the college student has time to be picky the tourist is very limited in the time they will spend in the city and must efficiently plan their day.