

TRAVELBUDDY: A CLOSED-COMMUNITY CARPOOLING SYSTEM  
AS A CASE STUDY FOR WEB DEVELOPMENT WITH FOCUS ON USABILITY,  
DESIGN AND INFRASTRUCTURE

by  
SELİM KURAR

Submitted to the Graduate School of Arts and Social Sciences  
in partial fulfillment of  
the requirements for the degree of  
Master of Arts

Sabancı University  
Summer 2011

TRAVELBUDDY: A CLOSED-COMMUNITY CARPOOLING SYSTEM  
AS A CASE STUDY FOR WEB DEVELOPMENT WITH FOCUS ON USABILITY, DESIGN  
AND INFRASTRUCTURE  
AS A CASE STUDY

APPROVED BY:

Lanfranco Aceti: (Thesis Supervisor) .....

Onur Yazıcıgil: .....

Selçuk Artut: .....

DATE OF APPROVAL: .....

© Selim Kurar 2011

All Rights Reserved

## ABSTRACT

### TRAVELBUDDY: A CLOSED-COMMUNITY CARPOOLING SYSTEM AS A CASE STUDY FOR WEB DEVELOPMENT WITH FOCUS ON USABILITY, DESIGN AND INFRASTRUCTURE

Selim Kurar

MA, Visual Arts and Communication Design, 2011  
Thesis Supervisor: Lanfranco Aceti

**Keywords:** web design, usability, infrastructure, responsiveness, carpooling

The Web and the technologies that revolve around it are on a rapid rise. As the use and utilization of Web grows broader, so does the dependency on Web technologies. If a web site aims to survive and mature, in a world where 10.000 web sites per day are published, the web designer/developer must assess the factors associated with Web site success during the development process. While a web site's long-term success is dependant on many factors, this thesis's focus is on its initial development which consists of three interconnected factors; usability, design and responsiveness provided with infrastructure.

Usability, design and responsiveness provided with infrastructure are the three primary factors needed to establish a quality and complex web site. While each factor has its own priorities and requirements, they are all dependant on each other. Establishing a balance between these factors are key to success. This thesis will discuss how to utilize the three factors to create a successfull web site.

The concluded information will be used with the aid of research on carpooling, to create "TravelBuddy", a closed-community carpooling system prototype. The thesis will also discuss the development process in association with Web site success factors and planned future work.

## ÖZET

### WEB İÇİN GELİŞTİRMEK: KULLANILABİLİRLİK, TASARIM VE ALTYAPI ÖRNEK PROJE OLARAK YOLARKADAŞIM: KAPALI SİSTEM ARABA PAYLAŞIMI PROTOTİPİ

Selim Kurar

Görsel Sanatlar ve İletişim Tasarımı Yüksek Lisans Programı, 2011  
Tez Yöneticisi: Lanfranco Aceti

**Anahtar Kelimeler:** web tasarımı, kullanılabilirlik, altyapı, çabuk yanıt verebilme, araba paylaşımı

Web ve onun etrafında dönen teknolojiler çok hızlı bir yükselişler. Web'in kullanımı ve faydalanma yöntemleri arttıkça, Web teknolojilerine olan bağımlılıkta artmaktadır. Eğer yeni bir web sitesi, günde 10.000'in üzerinde web sitesi yapılan bir ortamda hayatta kalabilmek ve büyümek istiyorsa, web sitesi başarı faktörleri tasarımcı/geliştirici tarafından geliştirme sürecinde çok iyi değerlendirilmelidir. Bir web sitesinin uzun süreçteki başarısı birçok faktöre bağlı olsada, bu tez onun ilk gelişim sürecinde ihtiyacı olan 3 ana faktörü ele almaktadır. Bunlar sırasıyla, kullanılabilirlik, tasarım, ve altyapı ile sağlanan çabuk yanıt verebilme yeteneğidir.

Kullanılabilirlik, tasarım, ve altyapı ile sağlanan çabuk yanıt verebilme yeteneği; kompleks ve kaliteli bir web sitesi oluşturmak için gerekli olan 3 ana faktördür. Her faktörün kendine has öncelikleri ve ihtiyaçları olmakla birlikte, her biri birbirine bir şekilde bağımlıdır. Başarılı bir web sitesine giden yol, bu faktörler arasındaki doğru dengeyi bulabilmekten geçer. Bu 3 faktörün ne şekillerde kullanılması gerektiği, bu tezin içerisinde tartışılacaktır.

Elde edilen sonuçlar, araba paylaşımı üzerine yapılan araştırmadan elde edilen bilgiler ile birlikte, "YolArkadaşım" adındaki kapalı sistem araba paylaşımı sisteminin prototipini oluşturmak için kullanılacaktır. Tez aynı zamanda, "YolArkadaşım" projesinin geliştirme ve tasarım sürecini, web sitesi başarı faktörleri ve ilerisi için planlanan gelişmeleri de göz önünde bulundurarak tartışacaktır.

## TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZET .....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES .....	ix
CHAPTER 1: Introduction.....	1
1.1 Introduction.....	1
1.2 Terminology.....	2
1.2.1 Web Linguistics.....	2
1.2.2 Specific to Carpooling.....	3
1.3 Background.....	4
1.3.1 Internet & Web Success.....	4
1.3.2 Carpooling.....	6
1.4 Aim and Purposes.....	7
CHAPTER 2: How To Design & Develop For Web.....	8
2.1 Introduction to Web Site Success Factors.....	8
2.2 Usability.....	11
2.3 Design & Interface.....	15
2.3.1 Navigation & Linking Structure.....	15
2.3.2 Page Layout.....	17
2.3.3 Content & Content Placement.....	18
2.3.4 Other Visual Elements.....	19
2.3.5 Tables & Forms.....	20
2.3.6 Achieving Better Design.....	21
2.4 Infrastructure & Responsiveness.....	22
2.4.1 Software vs. Hardware.....	22
2.4.2 Responsiveness.....	23

2.4.3 Better Responsiveness & Proposed Solutions.....	25
2.5 Success on Web.....	26
CHAPTER 3: Carpooling.....	27
3.1 Introduction to Carpooling Systems.....	27
3.2 Research & Literature Review.....	28
3.3 Existing Carpooling Systems.....	29
3.4 Problems & Risks.....	32
3.5 Success for Carpooling System.....	33
CHAPTER 4: TravelBuddy.....	34
4.1 Background & Introduction.....	34
4.1.1 Motivation.....	34
4.1.2 Prototyping & TravelBuddy Prototype History.....	35
4.1.2.1 Prototyping.....	35
4.1.2.2 TravelBuddy Prototype History.....	35
4.1.3 Limitations.....	37
4.1.4 Technologies, Languages & Software Used.....	38
4.1.5 Required Basic Use Cases Model.....	42
4.2 TravelBuddy - The Prototype.....	50
4.2.1 Technical Decisions.....	50
4.2.1.1 Site Hierarchy.....	50
4.2.1.2 Database Design.....	51
4.2.2 Design & Interface.....	51
4.2.2.1 Page Layout.....	51
4.2.2.2 Color Scheme & Fonts.....	53
4.2.2.3 Navigation & Linking Structure.....	55
4.2.2.4 Content & Content Placement.....	58
4.2.2.5 Tables & Forms.....	59
4.2.2.6 Other Visual Elements.....	61
4.2.3 How It Works Considering Usability & Responsiveness: Use Cases, Structure & Infrastructure.....	62
4.2.3.1 General Details on Back-End Infrastructure.....	62

4.2.3.2 Home Page, Login & Registration.....	62
4.2.3.3 Creating Events: Sharing a Ride & Requesting a Ride...	65
4.2.3.4 Browsing Events.....	68
4.2.3.4.1 Textual Browsing.....	68
4.2.3.4.2 Visual Browsing.....	71
4.2.3.5 Visuals & Graphics.....	78
4.2.3.6 Personal Pages.....	81
4.2.3.7 Forums.....	88
4.2.3.8 Background Processes.....	88
4.2.4 Accordance to Other Web Success Factors.....	91
4.2.4.1 Legality & Credibility.....	91
4.2.4.2 Semantic Approaches & SEO.....	91
4.2.4.3 Testing & Debugging.....	92
4.2.4.4 Error Handling.....	92
4.2.4.5 Maintainability.....	92
4.2.4.6 Marketing Strategy.....	92
CHAPTER 5: Conclusions & Future Work.....	93
5.1 Thoughts & Conclusions.....	93
5.2 Future Work.....	94
REFERENCES.....	96
SECONDARY REFERENCES.....	100



## LIST OF FIGURES

Figure 1 - The working login page of v0.2b.

Figure 2 - Browsing interface of v0.2b.

Figure 3 - The first web interface for Travelbuddy v0.3b.

Figure 4 - Basic use diagram derived from needed use cases.

Figure 5 - Site Hierarchy created with WriteMaps (<http://writemaps.com/>).

Figure 6 - Main navigation idle (not hovered) displaying what page the user is on.

Figure 7 - Both stages of control panel. Second one displays personal menu items that appear after successfully logging in.

Figure 8 - Dynamic table showing active rideshares.

Figure 9 - Form for sharing a ride.

Figure 10 - Homepage for TravelBuddy.

Figure 11 - How Login and Registration procedures work.

Figure 12 - Creating an event.

Figure 13 - Page for sharing a ride.

Figure 14 - Page for requesting a ride.

Figure 15 - Textual Browsing for Requests viewed in iframe.

Figure 16 - Diagram for browsing events.

Figure 17 - TravelBuddy Event Browser page

Figure 18 - The school map.

Figure 19 - Portion of school map displaying requests, contact card opened.

Figure 20 - Relativity View.

Figure 21 - Visuals & Graphics page.

Figure 22 - Mock up density map as the system doesn't have enough data to create one.

Figure 23 - Mock up destination chart as the system doesn't have enough data to create one.

Figure 24 - My Trips page where users can view their active events and unregister.

Figure 25 - phpBB3 forum.

Figure 26 - Background Processes.

## **CHAPTER 1:**

### **Introduction**

#### **1.1 Introduction**

This thesis will discuss the importance of and how to utilize usability, design and infrastructure to create a complex and quality web site/application, for this case a functioning web site prototype for carpooling in Sabanci University will be created as a project and case study.

Chapter 1 is aimed to be a brief introduction to readers by providing background information on designing & developing for web and carpooling. It will also cover the aims and purposes of this thesis along with a brief explanation of terms that will frequently appear in the thesis.

The development process for a complex web site/application and how to utilize usability, design, infrastructure trio will be discussed in Chapter 2. This chapter will be a combination of literature reviews on the subject and conclusions that are derived from those. It will cover the main requirements to create a quality and functional web site.

In Chapter 3, the requirements to create a carpooling web site and application will be discussed. A research covering literature review on published papers and proposed solutions will be discussed. Also in this chapter, currently active carpooling web sites and applications will be briefly reviewed.

In Chapter 4, the project "TravelBuddy" will be discussed. The motivation behind the project, limitations, technologies used will be covered. The project will be revealed in-depth and its positives and negatives will be presented. Its development process will be covered in accordance to usability, design and infrastructure. Other aspects of development process such as legality, marketing strategy, semantic approaches, error handling and maintainability will be discussed as well.

Chapter 5 will be the conclusion. It will also discuss possible future implementations and improvements.

## **1.2 Terminology**

This section is aimed to provide brief explanations to terms that will be frequently used throughout the thesis.

### **1.2.1 Web Linguistics**

Web Site:

Website, also referred to as Web site, or only site, is a collection of related web pages containing text, images, videos or various other digital elements. A website is hosted by one or more web servers, and is accessed via the Internet by a certain address called Uniform Resource Locator.<sup>1</sup>

---

<sup>1</sup> Conventional knowledge gathered from <http://en.wikipedia.org/wiki/Website>

Web Application:

A web application is an application that is specifically designed and coded to work in browsers and is accessible via network such as the Internet or an intranet. The application must be coded in a browser-supported language to enable web browsers to render the application executable.

Web Technologies:

Collection of browser-supported languages or back-end software that can be used to create web applications, such as HTML, PHP, Java, Javascript, MySQL.

SEO:

Short for Search Engine Optimization. A set of rules and required actions to achieve better visibility of a website or a web page in search engines.

Debugging:

A systematic process of finding, correcting and reducing the number of bugs, or unexpected errors, in a software; in an attempt to make it work more smoothly.

Responsiveness:

General term used for the speed, download time and response times of a web site or application.

### **1.2.2 Specific to Carpooling**

Carpooling:

System or idea which allows people to share rides depending on priorities such as location, destination, time.

Rideshare:

The term used for sharing a car ride between two locations by 2 or more people.

TravelBuddy:

A systematic solution aimed to allow people looking for a rideshare and people that propose rideshares to connect.

## **1.3 Background**

### **1.3.1 Internet & Web Success**

There has been a fast-paced and remarkable growth in the use and utilization of Internet over the past few years. World Wide Web is advancing on a daily basis, web sites and applications that are utilizing social networking, online shopping, webmail etc. are rapidly increasing over the Internet (Liu, Abdulla, Johnson, & Fox, 2008). Internet and the technologies that revolve around it act as an invaluable opportunity for people to satisfy their basic needs such as exploring, learning, organizing. Thus in daily life, people's dependency on the internet is growing rapidly as well. In 1994, it was stated that the internet allows us to communicate, share, interact, and even socialize (Berners-Lee, Cailliau, Luotonen, Nielsen, & Secret, 1994). Today, it has reached such a point that people need the internet almost in every field of daily life. People tend to use the internet for social interactions, communication, online transactions, or even paying their bills.

With the emergence of smart-phones, hand-held devices, e-book readers, and larger screens, the spectrum of hardware that could connect to the internet has increased massively. Also the infrastructure underlying the internet technology improved excessively which allows greatly increased transfer rates. These two aspects together provided the Internet users and developers alike with new and limitless opportunities and availabilities which lead to the exponential growth of web sites/applications. However, this growth also

brought a countless number of defected, useless, and badly designed web sites/applications along with it due to an unorganized development process that mainly disregarded usability, design and/or infrastructure.

Even though there has been some improvement on design and development of web sites/applications over the last few years as a result of more-educated and self-aware designers and developers, it is still impossible to say that web sites/applications have matured enough. A structured development process and knowledge along with skilled designers and developers with different backgrounds are required to create complex and quality web sites/applications (Barnard & Wesson, 2003). This development process should include; analysis of requirements, design, coding, testing and implementation (Lang & Barry, 2001).

Currently, there are a vast range of users with different backgrounds, cultures, and languages. As a result, it is impossible to meet the requirements of all users, thus a systematical approach should be taken to accommodate for most (Fitzpatrick, 2000). However as this systematic approach varies according to different people, a great confusion arises on how to most effectively implement these (Tredinnick, 2004). Evidently, there is a need for a proper development process which would organize and combine primary and secondary requirements for a functioning web site.

According to Liu and Arnett (1999), a web site's success depends on four major factors. These are quality of information and service, system use, playfulness, and system design quality. "WebQual" is a web site quality research conducted by Loiacono, Watson, and Goodhue (2000) which claims that web site quality begins with ease of use, usefulness in gathering information, usefulness in carrying out transactions, and entertainment value. Constantinides (2004), claims that usability and interactivity are the two major factors in the success or failure of a web site and that they are closely related to each other. He also adds that convenience, navigation, accessibility and speed are very important. User experience is another deciding factor on the success and failure of a web site. According to Preece, Rogers, and Sharp (2002), a good user experience depends on being satisfying,

entertaining, enjoyable, funny, helpful, and informative. It is also stated that look and feel contribute to the user experience. McNamara (2003) summarizes these factors into three; usability, functionality and customer satisfaction.

The preliminary research provides that usability, design and infrastructure are the primary factors for web site success. Usability is needed for good functionality and user experience. Design is vital because the final product must be enjoyable, entertaining, visually and emotionally fulfilling. Consistent and good design contributes to usability. Navigation is another important factor that contributes to usability but as it is a part of design, it will be discussed along with design and interface. Infrastructure is chosen as primary as it is the deciding factor on responsiveness, speed and accessibility of a web site/application.

### **1.3.2 Carpooling**

Carpooling is the process of sharing a car ride by 2 or more people while traveling to a pre-determined and mutually agreed destination. In practice, it is organized hitchhiking which involves some sort of communication structure to connect people offering and requesting rides.

In theory, carpooling seems beneficial according to a number of factors such as decreasing congestion on roads, reducing environmental pollution, allowing both drivers and passengers to cut costs. For example, Murphy (2007) states in "The Smart Jitney: Rapid, Realistic Transport" that carpooling would be even more efficient in lowering greenhouse gas emissions in than electric or hybrid cars. However, it is evident that carpooling has never become the first choice of transportation options due to the risks and problems it possesses such as security or credibility. There is a need for a system that can overcome these liability issues as urban transportation infrastructure is becoming more and more incapable in fulfilling the needs of continuously increasing number of drivers.



According to a survey conducted by U.S. Census Bureau (2009) in the States, studies have shown that 38% of people get to work by carpooling. This isn't a low number however considering the States has the highest participation rate in carpooling worldwide and that there isn't a single report or survey as such in Turkey, we can assume that the rate is much lower. Another evidence that points to low rate of ridesharing in Turkey is the lack of High Occupancy Lanes which are specifically reserved for vehicles with a driver and one or more passengers. Due to these reasons, creating an open-community ridesharing system such as Pickup-Pal isn't a logical choice to kick-start carpooling in Turkey, and would probably fail to succeed. However creating a closed-community ridesharing system, one that specifically targets certain small communities such as universities, or small residential areas; have a chance at creating its own community.

#### **1.4 Aim and Purposes**

This thesis aims to research a methodological process on how to create a quality web site and application focusing on utilizing usability, design and infrastructure. It will discuss how to use them in harmony so that each will benefit the other rather than obstructing each other.

Purpose of this research is to create a functioning closed-community carpooling web site/application prototype for Sabanci University called "TravelBuddy" which later on can be modified to suit any closed-community. For this purpose, this thesis will also discuss existing and proposed carpooling solutions and their problems to be able to overcome possible future problems.

## **CHAPTER 2:**

### **How To Design & Develop For Web**

#### **2.1 Introduction to Web Site Success Factors**

In order to create high quality Web sites and applications, developers and designers alike, must follow a well structured development process. Choosing the correct approach strategy for web development is critical to the success of the website and web-application. A well implemented web development process should result in a product which is targeted at the needs of a specific audience, while still properly conveying the view of the developer\designer. Developing and designing web sites that are responsive to user needs, is critical to all web designers and developers (Price, 1997).

A proper and well implemented design and development strategy must take several aspects into consideration. These aspects can be split into two pools as primary and secondary as stated in the web success factors in Introduction chapter.

Primary Aspects:

- Usability
- Design & Interface
- Infrastructure & Responsiveness

#### Secondary Aspects:

- Legality & Credibility
- Semantic Approaches & SEO
- Testing & Debugging
- Error Handling
- Maintainability
- Marketing Strategy

Primary aspects create the backbone of the project, while secondary aspects can be viewed as the muscle behind the progress. Primary aspects are crucial to the survival of a website or application, however that does not mean secondary aspects are less important. While they might seem optional, all of the secondary aspects are needed to make a difference in world wide web. Also we can't make a clear distinction between the importance of each item, as no one set of design or development principles will be satisfactory for all conditions (Smith, Newman, & Parks, 1997). A developer should always make a clear distinction on the vitality of each aspect in the early stages of the development process, according to the needs of the specific audience, he/she is targeting.

Usability is the most crucial aspect for both websites and web applications. If the website or application is difficult to understand, use or if people get lost trying to figure it out; they will just leave. There are many other options out there. A good design, interface and a proper navigation is just as vital, as they do contribute to usability in a great way. If the website doesn't have a clear navigation, where people can reach the desired information quickly, the users will again opt to leave. Infrastructure, and a good responsiveness level are also very important as they are the main elements behind the consistency of a website and application.

Legality and credibility are needed on two different platforms that interact with each other. While credibility will tell your visitors that you are trustable, legality will steer you clear of any law-related problems, thus protecting your credibility.

According to Netcraft (2011), there are over 110 million active web sites. It is claimed that there are two mainly accepted modes of user interaction, these are focused searching and exploratory browsing (Marchionini, 1995; Smith, Newman, & Parks, 1997). Nielsen (1997a) states that about half of the web users are search-dominant, which means, rather than following a trail of hyperlinks or peer suggestions, people started preferring reliable search engines (Google, Yahoo, Bing etc.) to reach the desired information. This fact makes the semantic approach and search engine optimization, a very important aspect if you want to reach out to people, as it is known that people will not look further than the second page of the search results.

Testing, debugging, and error handling may seem time consuming and troublesome. However all will contribute to the credibility of a web application or site. Testing will let the developer figure out problems beforehand, while debugging will let users to have a better experience.

Maintainability and marketing strategy goes hand in hand as well. As both are vital factors for the future and growth of a web site and application. If the developer wants a stable growth and turn the web site into an investment, both of these aspects should be well planned.

For the purpose of creating a new web site and introducing a fairly new system, this thesis will focus on utilizing the primary factors. The following sections contain research and literature review on usability, design and responsiveness with infrastructure.

## 2.2 Usability

Usability is a term, more associated with engineering approaches as an attempt to identify a set of systematic principles to ensure a usable and intuitive end-product (Pearrow, 2000). The same principles have also become valid for designing web sites and applications as the web became an increasingly essential interface (Nielsen, 2000a). Usability for web, refers to the ease of use and understandability of your final product, depending on the targeted specific audience. It can also be summarized as the interaction between the final product and the user.

Nielsen (2000a) states that the web site or application should be constructed in such a way that the user could immediately be able to understand the flow and the structure of it, and be able to see what it can provide.

According to Dr. Jakob Nielsen (2000a), author of *Eyetracking Web Usability*, there are five key factors contributing to usability:

1. Ease of learning the site and how to use it.
2. Efficiency of use once they have learned the site.
3. Memorization or how easy it is to remember tasks on the site.
4. Error avoidance and handling as they happen.
5. Overall satisfaction with the use of the site.

Bedi and Banati (2006), relates to these factors but also includes:

1. Appearance - Design and content placement
2. Feeling - Satisfaction of visuality
3. Trust - Overall credibility

According to Preece et al. (2002) there are four components of usability: learnability, flexibility, throughput and attitude. Shneiderman (1998) states that if a web site is able to fulfill the user's perceptions and the purpose of the user, only then it has a high level of usability. Constantinides (2004) adds that load times, navigation, information structure, search capability, and accessibility are also required to enhance the usability of a web site or application. These are also very important aspects that contribute to usability, however as they have their own separate strategy point, we will discuss them in the following sections.

To achieve a good usability, there should be certain standards and rules that create a usability guideline for developing web sites and applications. As discussed in the previous section, authors mostly have similar opinions while differing at minor extra aspects. Here we will take into account Nielsen's approach as it is a very well structured and widely accepted one. It is called Nielsen's Ten Usability Heuristics.

Nielsen has ten heuristics to serve as a usability guideline. Even though these are mainly for applications, it is known that web site design can benefit greatly from usability principles (Schneidermann, 1998). Nielsen (2005) prefers to call them heuristics as he says "...they are more in the nature of rules of thumb than specific usability guidelines." They were originally developed by Nielsen, in collaboration with Rolf Molich in 1990 (Molich & Nielsen, 1990). Later on, he refined the heuristics based on a factor analysis of 249 usability problems (Nielsen, 1994a) to derive a set of heuristics with maximum explanatory power, resulting in the current revised set of heuristics (Nielsen, 1994b).

"Visibility of system status:

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world:

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

#### User control and freedom:

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

#### Consistency and standards:

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

#### Error prevention:

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

#### Recognition rather than recall:

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

#### Flexibility and efficiency of use:

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

#### Aesthetic and minimalist design:

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

#### Help users recognize, diagnose, and recover from errors:

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

#### Help and documentation:

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large." (Nielsen, 2005)

Depending on the facts, we can conclude that usability of a web site or application depends on focusing on a specific targeted audience and desired outcome. All the items that are given as a required aspect for usability in above sections, contribute to usability in different ways, deciding on which ones to emphasize is a decision that must be made throughout the development process. For example, an e-commerce site should emphasize on consistency and standards, while a site aiming to create a social community should focus on user control, freedom, and constituting a match between system and the real world.

To improve the usability of both web sites and applications, the developer should keep consistent and clear content (Young & Watkins, 1997). The content must be informative and relative to its own right. The information should be hyperlinked, titled and indexed accurately and clearly for even the simplest user to easily access. Here, a special care should be given to titles, as they are mostly used by search engines while indexing and also appear as a page is bookmarked. Navigation and browsing features should be very clear to avoid stranding the user which means there should be no dead ends (Shotsberger, 1996). User should always be provided with a way to get back to whatever point he/she desires to return (Goldberg, 1997). Browsing should not be hindered under any situation, be it a different device or platform. Web site and application should be designed to work under all, if not most, current browsers.

Providing search options and site maps also enhance usability, especially if the web site consists of 50+ pages. Both features will make it easier for the visitor to access the desired information and prevent them from going back to search engines which will probably take them to other sources. A site map's main benefit is to provide the visitor with an overview of the site. It creates a visualization of the information architecture of your site for them to quickly browse through. Search options on the other hand, can go from a simple word search to advanced categorizable search filter depending on the amount of content. Even though the methods differ, they have a common point that has to be achieved for better usability. Search results should be accurate, to the point and clear. They should not include irrelevant links, they must be short and well described.



Finally, it can be concluded from the research that "learnability" is the most common and important factor contributing to usability. This means that first and foremost a web site/application must be easily understandable.

## **2.3 Design & Interface**

Poorly designed interfaces has lead to the failure of several high-profile websites (Chain Store Age, 1997). A proper interface design should include presentation, organization and interactivity (Schneidermann, 1998). While designing and interfacing for web, following items should be carefully considered;

- Navigation & Linking Structure
- Page Layout
- Content & Content Placement
- Visual Elements & Media
- Tables & Forms as means of conveying and retrieving information

### **2.3.1 Navigation & Linking Structure**

Navigation is the core of any web site's linking infrastructure. As users move through one page to another through hyperlinks placed on the page, navigation is vital to the accessibility of the web site. Machlis (1998) states that navigation should make the information easier to find. Navigation should have a clear and understandable mechanism and a consistent style of presentation, to allow users easy access to information. Nielsen (2000a) states that the users will not be able to understand the web site's flow and structure, if they can't figure out where on the site they currently are. He suggests that each page should clearly indicate which site the user is on as well as where in the specific structure of

the site. He also suggests that a proper navigation interface should provide the user with what they have previously visited, and where can they go from here.

Navigation can be implemented in a number of varying and inter-connected ways. Some of these are using plain text, graphic images as buttons, maps and keyboard shortcuts. All of these and some other less conventional methods can be used in any number of combinations to create an efficient navigation.

First and foremost, navigation must be visible to the user. Nielsen (2000a) suggests that the navigation area should be emphasized in a certain way, for example by using a different color background, larger or bolder fonts. Also it is suggested that the primary navigation should be placed on the left of the page or at the top of the content as those are the most distinguishable places on a web site. Both have its advantages and disadvantages. While top-tier navigation is more appealing and distinctive visually compared to left-side navigation, it lacks the space and scalability. To compensate for space and scalability, top-tier navigation can resort to drop-down or roll-overs, however this is undesirable by some users as Nielsen (2000b) says that users don't like to hover over every option to see what else is available.

To avoid stranding the user, navigation must be persistent, which means it must always appear in the same locations and in the same style on all of the pages with one possible exception. Home-page may have a different styled or placed menu, as long as it is not drastically different. Logo is also commonly accepted as a part of the navigation. Suggestion is that it should be linked to the homepage on all pages.

To improve navigation, website structure should be forthright and simple, avoid any unnecessary congestion or levels. This will provide minimal amount of "clicks" while the user navigates through the site's structure. If possible and visually consistent, providing a page track would help the user to navigate more clearly as well.

Linking structure is very vital to navigation. Links must be self-explanatory. If they are vague in any way, they should be supported by either sub-headers or visual elements (Berners-Lee, 1995). If linking structure is not clear, users will have problems in comprehending the structure of information. Links should not open in new windows, and only open in new tabs if they are pointing to an external http. According to Nielsen (1996), making links that open a new browser window and links that use html redirect are sins that should be avoided. Links should be made in such a way that browser's "back" button wouldn't be needed and yet shouldn't become unfunctional. This relates directly with what Nielsen (1996) states, for example, if a link opens a new browser window or even a tab, browser's "back" button will become useless; or if an auto-redirect html request is used, "back" button will become unfunctional and the user will be stranded on the page.

As there are millions of users with different backgrounds and experience, it is nearly impossible to create a single perfect navigation mechanism that will suit all. Considering this, it is viable to state that the aim of good navigation design is creating clear interface where any user can move intuitively.

### **2.3.2 Page Layout**

Along with the content, page layout is one of the most important factors for a user deciding if the site is appealing or not. A layout should contain, but is not limited to, a header, footer, and content area. Sidebars and additional toolbars may be included depending on the requirements.

Logo should be placed inside the header. Main navigational menu can either reside in header, sidebar or an additional toolbar. Any additional information such as copyright text or contact information that the developer desires to have in every page should be placed inside the footer.

While creating a layout, a grid based system should be applied. Brockmann (1996) says "The grid system is an aid, not a guarantee. It permits a number of possible uses and each designer can look for a solution appropriate to his personal style. But one must learn how to use the grid; it is an art that requires practice." Grid system for web design is pretty much similar to print design with the exception of negative space. Unlike designing for print, the designer should not expect the user to see the final product as a whole. The medium in this case is some kind of a screen and its resolution dictates what the user will view.

### **2.3.3 Content & Content Placement**

Content is the heart of a website. If it fails to deliver the required information to the user, rest of the website will fail as well. However it is not only providing the information that is important, it is how you provide it and how you place it.

In most cases, text is the main piece of content. Reading via monitor is harder and more tiring compared to reading from paper (Bostock, 1997). DeBra (1996) says that reading from a monitor takes around %30 longer than reading from paper. According to a research only 16-20% of the users read most of the text, the rest prefers scanning rather than reading every word. Nielsen (1997b) states that text should be kept short and scannable. Its width should be less than the content area, having short lines should be preferred to having fewer but longer lines (Hansen & Haas, 1988). To achieve scannability, text should be concise meaning that it should be written specifically for web, nearly half the length compared to writing for print (Nielsen, 1997b). DeBra (1996) similarly suggests that text given on a web page should be limited to keep the user focused. In the case that a long text has to be presented, using a number of columns is suggested (DeBra, 1996). Nielsen (1997b) suggests having one idea per paragraph and having the conclusion at the beginning rather than the end, because that is the best way to catch the attention of the user. Cotrell and Eisenberg (1997) also support this idea by stating that an "inverse pyramid" approach

should be taken which promotes important information to be placed on the beginning rather than the end. Spool and others add that white space should not be over-used as it decreases scannability (Spool, Scanlon, Schroeder, Snyder, & DeAngelo, 1999).

A professional or web specific font face should be used and using fancy or clumsy fonts should be avoided. Using a standard font size of 11 or 12 pixels is suggested, so that font-faces are clear and readable. Nielsen (2000) states that very colorful items should not be placed through the text as it will distract the reader due to the banner blindness that the user has developed. He also points out that blue should not be selected as the primary text color as it will confuse the user by making the text feel as if it is a link.

#### **2.3.4 Other Visual Elements**

Graphical and visual elements such as images and videos play an integral part in aiding users while also making the web site or application more attractive. However these visual and graphical elements should only be used when they are directly related to the contextual element of the site or application, otherwise they will become distractions (Cotrell & Eisenberg, 1997). Duval and Main (1995) suggest that more than three large visuals should not be placed in a single web-page, especially in the homepage as it will cause clutter.

As previously stated in Navigation section, links allow the user to navigate through the site's architecture. Clickable objects contain links (both hypertext and images such as buttons), visual enhancers (such as light-box), form and table controls (submit, sort etc.). The most vital property for links and clickable objects is they should look and feel clickable. If it is a text link, it should have a different color than the rest of the text. If it is a clickable object, it should be large enough to catch attention and should either have a hover effect or transition to make it feel clickable. Also for both text links and clickable objects, a slight downward shift while clicked can be used to give the feeling of pressing a button.

### 2.3.5 Tables & Forms

Standard textual or visual content aside, a proper web site will also include one or both of the following items; tables and forms. Depending on different situations, handling their design properly will contribute to the web site's usability and functionality.

Tables are an efficient way of presenting lists of information such as search results, list of registered people, schedules. Tables mostly consist of textual data that users would like to browse and analyze. Lammi (2010) suggests that tables should have a pattern because users should be able to make sense of the data provided. Lammi states the following approaches should be taken to create a proper table for web;

- "- Give the table a descriptive title, and also a brief caption if needed. Show the title at the top of the table or in the table header, and the caption below the table.
- Summarize data when possible.
- Show only the information that users really need to see, but provide a possibility to dig deeper into details.
- Remove all interface elements that are not necessary. Less visual noise means less distraction and makes it easier to concentrate on the table data. Focus should be on the information itself, not the design of the table. However, don't remove visual elements that are useful. Keep the elements that support structure, organization, and readability of the table, but use the minimum amount of visual weight needed.
- Use different styling for information; change color intensity and/or hue, use bigger font, bold text, thicker lines, different orientation to make important information stand out from other data. However, do also keep in mind that if you emphasize too much information, nothing will stand out anymore.
- Consider allowing the users to select what information to show in the table.
- Left align text." (Lammi, 2010)

She also suggests following items should be considered if the table is large;

- "- Consider making the table headers stay visible in the browser window as the user scrolls up and down a page to remind the user of what the columns in the table are.

- Use Endless Scrolling or Page Pagination pattern when all the results doesn't comfortably fit within one page.
- Allow users to sort the table by clicking table headers and sort items in a logical order by default.
- Allow users to filter the table by a search mechanism." (Lammi, 2010)

Forms are the simplest way of communicating with the user. They can be used for contact, information retrieval or certain requests. Forms consist of labels, information fields (text fields, checkboxes, radio buttons, dropdowns etc.) and control buttons.

Correct use of labels are vital to the success of the form. Labels should be concise and self-explanatory. They should be visually enhanced to catch the eye. It is suggested to have them above the information fields and left aligned.

Information fields are the main part of forms. They are the fields that the user is expected to fill. Every information field should be designed to serve its purpose without confusing the user. Text fields should only be used if the required information is an open-ended one such as e-mail or username, if the information is a containable set, then the user should be directed to make a choice rather than left guessing. For example, if the user is expected to choose a day of the week, he/she should be presented with checkboxes or radio buttons rather than filling a text field.

### **2.3.6 Achieving Better Design**

- Nielsen (1997b) and DeBra (1996) suggest that text should be broken into sequential pieces and properly connected to each other through a good hyper-linking structure.

- Fluid grids, and flexible hypermedia will make the design compatible to more devices and platforms thus improve its accessibility and possibly responsiveness. As a result the usability will increase as well.

- Correct use of positioning through padding, margin and border which are the most confused elements of styles, will make the design visually more appealing along with a better usability.

## **2.4 Infrastructure & Responsiveness**

Infrastructure for web can be split into two: software and hardware. Both are vitally important for the usability of a web site/application as they are the main factors creating responsiveness. However as we are discussing design and development, we will only briefly discuss the hardware aspect and focus more on why optimizing responsiveness is important and how to achieve it.

### **2.4.1 Software vs. Hardware**

Hardware portion of the infrastructure mainly consists of hosting, web servers and even though it is not hardware, as it is related to this content, domain name. Hosting is where all web pages, documents, styles, images, practically everything of a web site is kept. Today, most people prefer renting hosting services from hosting companies. The important factor is to choose a reliable company rather than a cheap one. If the hosting company has poor servers, the site might get a lot of downtime meaning the web site will become unavailable or respond slowly more often which will negatively affect its reliability and responsiveness. Domain name provides an identity for the site. It is unique and is the address to access the site. Choosing a relevant and easily memorable domain name will always help a newly established site. Web servers can be summarized as the server a site is running on. It is the server, the chosen hosting company provides. It is important to determine what operating system is required to run the designed web site/application depending on the coding languages used beforehand, so that a site can be functional.



Unlike hardware, choice and use of software is entirely depending on the developer. Software infrastructure is the back-end components of a web site/application. Before choosing the software following should be considered;

- Will the site be static or dynamic or a combination of both?
- Where and in what OS will it be hosted?

Only after carefully answering these questions, the developer can decide either to use a Content Management System (CMS) or build everything from scratch by making choices between PHP, Java, Perl etc. All have their advantages and disadvantages. CMS systems such as Drupal, Joomla, Wordpress are easier to manage due to their growing communities and vast resources. However they are also limited in their functionalities, and editing them to suit certain needs is harder than building those from scratch. The choice depends on the developer as there are no clear lines defining a good or a bad choice concerning these software. The important part is how to utilize them for better responsiveness and usability.

#### **2.4.2 Responsiveness**

Responsiveness refers to the response time of a web page, or simply the time for a web page to download entirely. Response time of a web page is said to be the most straightforward metric in quantifying user satisfaction (Hoxmeier & DiCesare, 2000). Nielsen (2010) states that as most people have broadband today, it might be acceptable to say that loading times are less important compared to 1990s where even image download was an issue. However he also stresses that responsiveness is still an important factor, just on a different level and it will always matter because of the following reasons;

"- Human limitations: Especially in the areas of memory and attention. We simply don't perform as well if we have to wait and suffer the inevitable decay of information stored in short-term memory.

- Human aspirations: We like to feel in control of our destiny rather than subjugated to a computer's whims. Also, when companies make us wait instead of providing responsive service, they seem either arrogant or incompetent." (Nielsen, 2010)

There are multiple researches conducted on response-time limits. Jonathan Palmer (2002) states, users demand that web pages load quickly. They generally start losing patience after 8 seconds and their chance of waiting drastically drops after 10 seconds. Liu and others (Liu, Abdulla, Johnson, & Fox, 2008) support this by stating web client timeout value should not exceed 10 seconds. Nielsen has a much more in-depth research on the subject which he claims, is based on 40-year-old research by human factors pioneers;

"- 0.1 seconds gives the feeling of instantaneous response - that is, the outcome feels like it was caused by the user, not the computer. This level of responsiveness is essential to support the feeling of direct manipulation.

- 1 second keeps the user's flow of thought seamless. Users can sense a delay, and thus know the computer is generating the outcome, but they still feel in control of the overall experience and that they're moving freely rather than waiting on the computer. This degree of responsiveness is needed for good navigation.

- 10 seconds keeps the user's attention. From 1-10 seconds, users definitely feel at the mercy of the computer and wish it was faster, but they can handle it. After 10 seconds, they start thinking about other things, making it harder to get their brains back on track once the computer finally does respond. A 10-second delay will often make users leave a site immediately." (Nielsen, 2010)

Nielsen (2010) concludes his research by saying "long delays cause unpleasant user experience. This will result in losing half of the visitors just because your site is a few seconds too slow for each page."

Consistency of response times throughout the whole web site is also very important as the users expect equal responsiveness within a site. (Palmer, 2002)

### 2.4.3 Better Responsiveness & Proposed Solutions

Designers and developers must pre-determine appropriate server requirements, bandwidth limits, programming languages, graphical elements, and applets according to their targeted user group and site content, to be able to effectively improve responsiveness. For example, if the web site is going to have over 100 pages and should be kept up to date, most processes have to be controlled by a back-end system created with PHP, or Perl.

There are some general proposed solutions and recommendations that would help improve the responsiveness of a web site.

- Avoid inline stylesheets and javascripts. It is more logical to make them external as most pages will make use of the same stylesheets and javascripts. This will make the HTML document shorter in lines of code, but more importantly stylesheets and javascripts will become cache-able by browser thus decreasing response times.
- Be careful not use the same script twice.
- After completely finishing coding, try compressing htmls, or minifying stylesheets and javascripts. Minifying is a process that can be done through programs that can be found freely on the Internet. Minifying removes unnecessary spaces, line-breaks and compresses the code into a few lines. Code becomes harder to read but also much smaller compared to its former size. On the other hand, HTTP compression is claimed to reduce large textual data around 70%.
- Optimize media (images, videos etc). For example, prefer SD quality video over HD unless it is a definite requirement according to the site's content.
- If the aim is to create a graphic heavy web site/application where no other precaution can be taken to reduce load times below acceptable amounts, try to give an indication to the user that the site/application is loading. Nielsen (1993) suggests showing the actual loading time is better than just a loading indicator but states even just placing an indicator is better than nothing.

Developers and designers should be aware that users don't really care why response times are slow. They expect the site to be responsive, low levels of responsiveness will often directly affect the level of reliability of the site and the users will move elsewhere.

Over-using graphical elements and embedding too many objects into the web page are the main reasons in poor response times. "The number of embedded objects was 2.6 times more costly than total page size for response time. Caching was found to improve response times by over 19%. The number of lines of code and the size of cookies were found to affect response times the least." (Chiew, 2009)

However it should not be forgotten that web page response time will be dependent on the characteristics of a web site and its purpose. Providing and protecting a proper balance between visuality and responsiveness will give the site/application the maximum usability.

## **2.5 Success on Web**

Success on the web depends on providing a good user experience so that the user will desire to come back to the site or application. To be able to provide a good user experience, the designer/developer must initially give priority to usability, design and infrastructure.

All the information gathered in this chapter will serve as a roadmap while creating TravelBuddy.

## **CHAPTER 3:**

### **Carpooling**

#### **3.1 Introduction to Carpooling Systems**

A carpooling system consists of mainly a web site or application or both and reside on the World Wide Web which makes them accessible through any device or platform that can connect to the Internet.

There are currently several active carpooling systems that work through their own interfaces or through social networks such as Facebook. Even though some of these systems allow users from Turkey, there isn't a significant usage currently and there are no carpooling systems that are based in or specifically created for Turkey.

Carpooling systems can be informally split into two branches. These are open-community carpooling systems which everyone can register and generally work through Google maps; and closed-community carpooling systems which are specifically designed for a certain small community (such as universities) which only people from that community can register as some certain requirements have to be met to be accepted. Closed-community systems prefer more local maps to provide pinpoint locations of pick-ups as pick-up locations are already designated.

In the following sections, a research of published material on carpooling and existing carpooling systems will be used to determine common problems and risks of

building such a web site. If available, suggested solutions will be discussed as well. Considering the facts, the question "Why a closed-community system should be preferred to an open-community one in Turkey?" will be answered.

### **3.2 Research & Literature Review**

There is still a major interest that is rapidly growing and well recognized in the direction of Rideshare systems (Sottini, Abdel-Naby, & Giorgini, 2006). In 1995, Edward Walbridge estimated that energy savings from ride sharing would be 48 million barrels of crude oil per year. "The associated reduction in congestion is estimated to save driver time worth \$6.2 billion annually." (Walbridge, 1995) These are estimations that were made 16 years ago without even considering the rapidly increasing traffic congestion in the last years. Dailey and Meyers (1999) observed that the number of users that participate and the number of carpools formed is quadratic. This quadratic formula shows that as number of users increase, the rate of carpooling will increase drastically. This observation suggests, even though carpooling might have a very little effect on traffic at the beginning, as the users increase, traffic will decrease greatly. However these systems are still not widely used partly due to people not being encouraged by local governments (Hartwig & Buchmann, 2007), and mainly due to concerns such as security and liability that people share. Resnick (2006) states in his research report "SocioTechnical Support for Ride Sharing", that the commonly known stories of murder and rape connected to carpooling don't reflect the truth; rather the biggest concern of people are bad or irresponsible drivers.

Before starting to build a carpooling system, Resnick (2006) suggests researching how the following requirements could be implemented;

- As a carpooling system depends on fixed locations and destination for pick-ups and drop-offs; the system should be very clear on directing the user. To be more clear, consider the following example; "a student **X** may need to know the most frequently used meeting point within a university campus. The idea of the Implicit Culture framework is to let the system suggests the meeting

points that are frequently used by other university students (i.e., members of that community). In this case, the system may suggest to student **X** to move to the most frequently used meeting point within the university campus, which is the Car Parking." (Sottini et al., 2006)

- A simple user interface which will allow both carpooling participants to easily submit their queries and search possible options.
- A reputation or security system which will make the overall system more reliable.

Resnick (2006) also suggest creating a ride-matching algorithm which is widely used by ridesharing systems. However, conducted research results show that explicitly entering ride and profile information and then letting the system do the matching process through a computational method is not the best strategy. It in fact hinders the system and does not necessarily return any positive results. On the other hand, allowing the user to directly interact through the system with an informal communication and negotiation of text search techniques should be preferred (Ghelawat, Radke, & Brerton, 2010).

Common reasons for failures of ridesharing systems are too complex user-interfaces, complicated rules, hard to understand instructions, and incorrect marketing strategies or targets (Kirshner, 2006). Resnick (2006) states that if correct approaches are taken, the system would not only become successful but also provide a great potential to create new social interactions.

### **3.3 Existing Carpooling Systems**

Following are some of the current web-based carpooling systems with brief descriptions of their aims and capabilities.

## Pickup Pal

A well known carpooling website, mainly based in USA, although has users all over the world. It is a social network for ride sharing. It has a simple interface using the Google Maps for displaying users and pick up locations. It is also the first carpooling system that has faced a lawsuit opened by Ontario Highway Transport Board.

## Avego

A carpooling application created for Apple iPhone. Avego uses a simple and understandable user interface powered by iPhone's GPS capabilities. The website keeps track of trips, passengers and drivers. It also keeps information on credits spent and received thus handles the funds exchange internally. It also tracks user and trip positions to ensure security.

## RideShare Online

An online service created by the cooperation of multiple carpooling services based in Washington and Idaho. It is focused on creating scheduled carpooling rather than instant ridesharing like the other services. Users can opt to drive or be a passenger and the system matches users who can travel to work together, then notifies them.

## Flinc

Declares itself as the social mobility network. It is designed for Android and Apple iPhones. It creates a connection between navigation systems and the GPS capabilities of smart-phones. It aims to be a social networking platform as well. The only drawback is, as they claim to be under development continuously, their site gets a lot of downtime.



## Share My Ride

The interface is very simple and usable. It is not a dynamic system meaning that you can't get instant rides, it is more like RideShare Online. Trips are pre-planned and the system notifies users if there are possible matches on routes.

## Piggyback

Piggyback is an application created for Android. Major difference to other systems is the fact that it relies on graphical elements rather than text which makes it intuitive. It uses its own algorithm to match users, so this one is static as well. There is a well built rating system which allows users to rate each other on subjects like driving skills and friendliness after they share a ride.

## Zimride

The most established Facebook application for carpooling. Users don't need a separate account as they use their Facebook accounts. Zimride is very successful in using the social element of Facebook while connecting people who would like to travel in the same destination. It especially focuses on universities in USA, it is claimed that Stanford has the highest carpooling rate.

## GoLoco

Another application developed for Facebook, but also has its own web site. Compared to Zimride, it is much less popular on Facebook. However its web site has a lot of users as it successfully integrates the social element of Facebook to its own. This is not a dynamic application as well. Trips are pre-planned through the system.

### 3.4 Problems & Risks

Here are some of the major issues that have to be addressed to achieve a functioning carpooling system.

#### 1) Security

Similar to the case of picking up hitchhikers, in carpooling people travel with a total stranger. This is the most important issue to be addressed because there are multiple risks from the classic psychopath to an amateur driver. One possible and yet inefficient solution suggested by Murphy (2007) is installing Auto Event Recorders on cars, something as simple as real-time feed cameras to enforce security. More logical solutions for open-community ridesharing security include using RFID, finger-prints, voice recognition, live position tracking on maps; however these are all costly and technically challenging solutions. (Gruebele, 2008)

#### 2) Trust

This issue is similar to security in many ways due to the obvious fact that there is a total stranger involved in the act. The highly suggested and most reasonable solution to avoid this problem is to implement a rating system so that people can rate each other on various subjects ranging from punctuality to friendliness.

#### 3) Money Exchange

This is a tricky issue, as some people might find it awkward and unreliable to exchange money by hand. If money is exchanged by hand, the carpooling system does not have any control over it and has to take precautions to prevent being involved in liability issues that might arise between people. The logical move would be to have a payment

system connected to the carpooling system. However, that solution is imperfect as well due to the insecurity people feel against online payment systems.

#### 4) Legality

Legal issues are another barrier that has to be overcome. For example, PickupPal have gotten in trouble by letting passengers pay drivers for rides, which bus companies thought was stealing their business, and apparently violated the Public Vehicles Act in USA. They had to put in place a set of rules to survive the lawsuit.

### **3.5 Success for Carpooling System**

The findings indicate that for a carpooling system to be successful, it has to be properly implemented with special care taken to avoid risks. A very simple and intuitive system where users can directly interact has to be created.

There are two main reasons why a closed-community system has to be preferred over an open-community one;

- 1- There were already open-community carpooling applications and web sites that were available to Turkey; yet they have never been able to reach their potential.
- 2- Security and Trust issues can be easily resolved in closed-community where most people know each other. For example, in a university people can be enforced to register with their school id and/or school e-mail thus protecting the system of impostors.

Additionally, it is stated that people prefer to spare 23% more time to pick up a friend from their social network rather than a total stranger (Hartwig & Buchmann, 2007).

## **CHAPTER 4:**

### **TravelBuddy**

#### **4.1 Background & Introduction**

##### **4.1.1 Motivation**

The overall objective of this project is to create a working prototype for an online carpooling system that is respectful to primary web success factors which have been discussed in the previous chapters.

The lack of transportation options in Sabanci University and Zekeriyakoy (which my family resides in) which are respectively 40 km and 26 km away from the midtown, were both the motivation and the driving force behind this project.

While there are some similar applications and systems that provide ridesharing online which are available in Turkey, none of them had managed to establish themselves. It is thought that a closed-community system such as this, would succeed in places where the available open-community systems have failed such as trustability. It is also thought that it can create social communities of its own where people get to meet others whom they live in close proximity and yet haven't met.

## **4.1.2 Prototyping & TravelBuddy Prototype History**

### **4.1.2.1 Prototyping**

There are 3 main types of prototyping these are concept, horizontal and vertical. All prototyping methods can be adopted alone at the early development stages of a project, as the project progresses they start to combine to create something intact both visually and functionally.

Concept prototype is created to illustrate the vision of the system with respect to design, structure and functionality. It is described as the look and feel of the system. Horizontal Prototype is described as the outer shell of a system. It is how the menus, screens, reports should look and function without actually functioning. A Vertical Prototype is the core functions of a system. They are fully functional but stripped of visuality. It is usually done to demonstrate the working system.

### **4.1.2.2 TravelBuddy Prototype History**

v0: Called version 0 because it is the origin point of the whole project. There was no web interface, only a processing program that is static. This program was using processing.js and HTML's canvas element to work in browsers.

v0.1b: Concept Prototype. Collection of a few design ideas, mock-up Photoshop interfaces, first database schema.

v0.2b: Vertical Prototype: Created database, added all basic functions, no interface but fully working prototype.

Logged in as: [selim@su.com](#) [Logout](#)

Email:

Password:

Or if you are an unregistered user, please [register](#) here.

Figure 1. The working login page of v0.2b.

Show  entries

Search:

ID	By User	Location	Destination	Capacity	Payment	Date/Time
1	selim@su.com	P6	BAKIRKOY	0	5	2011.07.13 12:43
2	nazli@su.com	P7	CADDEBOSTAN	1	0	2011.07.14 21:12
3	aysem@su.com	P3	BAKIRKOY	0	5	2011.07.14 09:11
4	doruk@su.com	P12	USKUDAR	3	10	2011.07.13 11:54
5	metin@su.com	P13	BESIKTAS	0	0	2011.07.16 12:12
6	ekin@su.com	P8-1	ERENKOY	3	4	2011.07.12 23:44
8	omur@su.com	P12	LEVENT	4	6	2011.07.14 03:05
10	izi@su.com	P6	BAKIRKOY	3	6	2011.07.13 02:05

Showing 1 to 8 of 8 entries (filtered from 10 total entries)

Figure 2. Browsing interface of v0.2b.

v0.3b: Vertical & Horizontal Prototype: First fully working web interface, new tables and functions have been added. Visual Applications added and made dynamic.

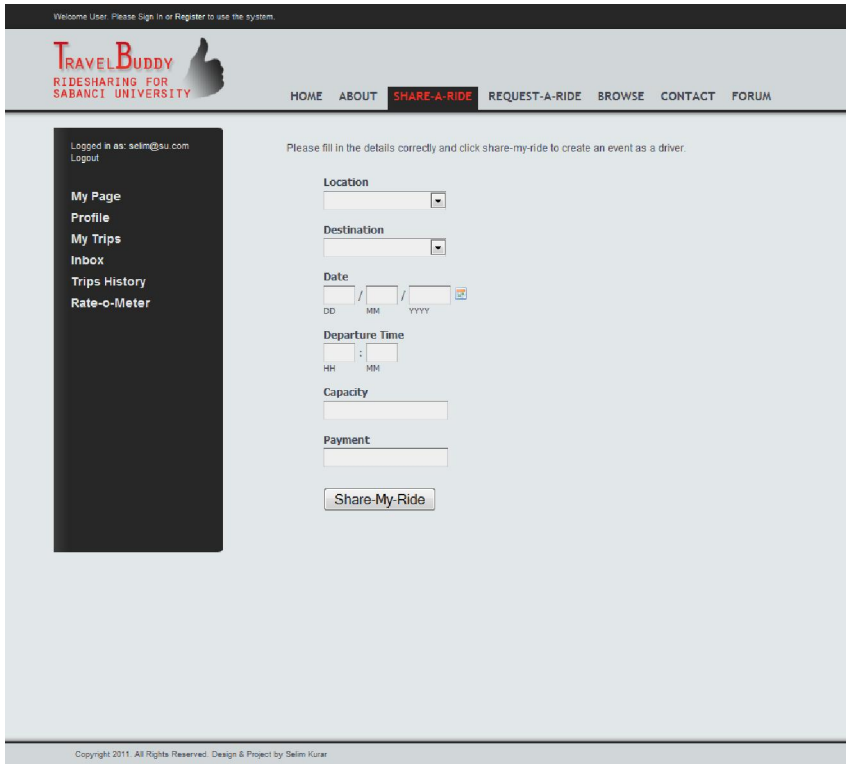


Figure 3. The first web interface for Travelbuddy v0.3b.

v0.4b: Current version which the thesis will discuss. Vertical & Horizontal Prototype. All functions are working properly, new visual applications have been added.

### 4.1.3 Limitations

Since creating a system of this magnitude and complexity by one person alone is rather irrational, the goal was to implement a base prototype that could be improved with new features as needed and later could be marketed and grow into a full scale commercial product, if seen fit.

The major difficulties encountered during the design and implementation of the project were;

- Trying to figure out and cover what kind of features, a carpooling web site must have.
- Creating a design that looks serious and yet playful in nature.
- Designing a database that can handle massive amounts of data while not hampering the system.
- Efficiently using PHP scripts so that usability and responsiveness improves while keeping data traffic low.

The current prototype version (0.4b) has been tested on Windows OS. The website functions on all popular browsers, however to be able to use the visual interfaces, user must have one of the latest versions of Firefox, Chrome, Opera or Safari. As visuals require canvas element of HTML5, currently Internet Explorer can't be used while viewing visual interfaces.

#### **4.1.4 Technologies, Languages & Software**

##### 1) MySQL & Database

MySQL is a popular choice of database management systems for use in web. It is developed, distributed and supported by Oracle Corporation. The current release that is going to be used in this project is MySQL5. The main reason behind choosing MySQL is the fact that it is supported by all popular languages and works on many different platforms. This would allow using multiple languages throughout the project without causing deficiencies. MySQL is also famous for its high performance and simple yet powerful query language.

"A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer



database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications. MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility."<sup>2</sup>

## 2) HTML & CSS

HyperText Markup Language, or more commonly known as HTML is the basic building block of a web page. Cascading Style Sheet (CSS) is a presentation semantic used to define the style and formatting of HTML documents, and thus the web page's.

## 3) Java & Processing

Java is an object oriented programming language. Java syntax is similar to C and C++ but has a simpler object model. Processing is an open-source programming language created using Java, thus it has a similar syntax. It is even possible to use Java language to some extent inside the processing language.

Processing can be used as a sketchbook to create visual context and application. This project will take advantage of Processing's on-the-go sketching element.

## 4) Javascript, JQuery & Processing.js

Javascript is an object-oriented, prototype based scripting and functional programming language. Even though it adopts names and naming conventions from java, it should not be confused with Java as they do not pose any relationship. Javascript is very useful in creating advanced interactive interfaces for web.

---

<sup>2</sup> Information directly quoted from MySQL homepage;  
<http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>

JQuery is a very popular javascript library. There are thousands of useful JQuery plugins that can add usability and flexibility to a web site.

Processing.js is the javascript library specifically designed for the web use of Processing. Processing.js allows the creation of data visualizations, digital art, and interactive animations within web standards and without requiring any plugins.

Fancybox, DataTables and LavaLamp JQuery plugins will be used in this project.

#### 4.1) DataTables

JQuery plugin used to create tables that are sortable, filterable and searchable with the data retrieved from the database.

#### 4.2) LavaLamp

JQuery plugin used for the animated top main navigation menu. Makes use of the Easing JQuery plugin to create the smooth transition effects.

#### 4.3) Easing

JQuery plugin that allows smooth transitions. It is used with a couple of personally written javascripts to create image and link transitions.

#### 4.4) Transition

Javascript that allows only to load content area and not reload other items such as header and footer as long as they don't change. Thus improves load times.

#### 4. 5) Calendar

Javascript that allows date selection from an interface similar to Window's calendar.

#### 5) PHP

PHP is a commonly used and popular scripting language that is especially suited for Web back-end development and can be embedded into HTML. PHP focuses on the server-side scripting; as a result, for the PHP parser to function, a web server and a web browser are required. This project will be using PHP5 as it allows the development of large-scale maintainable object oriented applications, and is also a very suitable environment for XML processing.

PHP will be the most common element in this project as it will support the entire back-end. It will be used for all the interactions with the database, and thus to make the site dynamic rather than static. It will also be used to process and create XML documents as needed.

#### 6) XML

XML is a simple, flexible and standardized text format which can be read, processed and written by different platforms, environments, programs and languages. This standardization, flexibility and universality makes it the perfect text format for data and information exchange between multiple non-related platforms, environments, programs and languages.

Combining the universality of XML with the extensive power of Web Services could allow the creation of complex operations which would normally be very hard to achieve. Programs and applications that are using different languages which can't normally interact through direct means, can use XML in order to interact. Thus enriching the potential of the end-product.

This project will use XML to create a direct interaction between Processing and MySQL database to create visuals from real-time data stored in the database.

#### **4.1.5 Required Basic Use Cases Model**

Following are the core requirements of the TravelBuddy web site. The web site will not be limited with these use cases; but these use cases will be used on forming the base of the web site, web site structure map and database structure.

##### 1) Registration

Precondition:

- New user.
- Fits community profiling rules.

Initiator:

- Visitor

Details:

As TravelBuddy is a closed community system, it requires registration which is limited to a certain community and protected by certain preconditions. For example, in the case of Sabanci University, user registration can be limited with SU e-mail.

Visitors can only access certain pages such as Home Page, from there they should fill the registration form properly to be registered and gain access to the rest of the web site.

The registration process should at least include e-mail, name, surname, and password. It can also include additional details depending on the community. The mandatory fields should be checked during submission and not accepted if they are not filled. Submitted data should be checked for duplicates in the database, if there are no duplicates the data will be stored in the database.

## 2) Login

Precondition:

- User must be registered

Initiator:

- User

Details:

User must login to gain access to the entire website. This is the authentication process that checks if the user has entered the correct credentials. Once a user is logged in, he/she can use the features the web site provides.

## 3) Access Protection

Precondition:

- Visitor trying to access user specific pages.

Initiator:

- System

Details:

- When any user specific page is attempted to be accessed; system throws a user authentication test. If the test fails due to user not being recognized, the user is redirected to Home Page.

#### 4) User

##### Precondition:

- Being registered

##### Details:

The user variable is the most important aspect of the interactivity of the web site. A critical user mass is required for the system to achieve success as users are the main objects that use the carpooling system. A logged in user can share, request, browse, and register for rides.

#### 5) Profile Management

##### Precondition:

- Registered User
- Logged In

##### Initiator:

- User

##### Details:

Every user should have a simple profile which includes basic information about them. Some of the information such as name, surname and e-mail are directly registered into profile during registration, and should be un-editable to protect consistency and security. Profile management can be used to allow people add extra details if they see fit, such as phone numbers, gender, age etc. Viewing profile details of people can encourage more interaction.

## 6) Creating Events

Precondition:

- Registered User
- Logged In

Initiator:

- User

Details:

Events can be split into two main actions in carpooling.

### 6.1) Offering to Share Ride

Carpooling is about sharing a ride, therefore users must offer to share their rides for the system to function. Every user should be able to create a carpooling event through filling and submitting a form. This form can have optional inputs as well as mandatory ones. The user should at least provide a location for pick-up, a destination, a departure time, capacity of the vehicle and a price if he/she desires. Optional inputs such as smoking in the car can also be used.

### 6.2) Request a Ride

Should work in the same sense as sharing a ride. Again there should be a simple form that allows users to input at least a location, a destination, and a desired latest departure time. In both cases, to prevent ambiguity locations and destinations should be limited to a list.

## 7) Browse/Search Events (Shared Rides & Ride Requests)

### Precondition:

- Registered User
- Logged In

### Initiator:

- User

### Details:

There should be a page for searching currently active events. This page should be structured in either tables or information lists format. If possible, user should be presented with the options of filtering, searching and sorting columns to suit their needs.

## 8) Delete Events

### Precondition:

- Registered User
- Logged In
- Has created a carpool or requested a ride-share that is currently active

### Initiator:

- User

### Details:

User should be presented with the option of deleting an event he/she created or unregistering from an already registered event. This can be provided through a form that has to be filled and submitted, which then checks if the user is really in that event and if it is true, then deletes that event.



## 9) Expire Event

### Precondition:

- Created Event departure time has passed

### Initiator:

- System

### Details:

- A routine maintenance action that runs on the background. It should check all events to find out which ones' departure times has passed and set them to expired in the database. Thus removing them from active status and making them unregistrable and unviewable. It should not delete the events entirely as expired events would be needed in certain action checks.

## 10) Rating

### Precondition:

- Created carpool has expired
- Rater and rated user have traveled together

### Initiator:

- User

### Details:

A rating system should be implemented to allow users that have traveled together to evaluate each other. This system should allow the rater to allocate a certain quality point to the person they are rating, and also a reason to why they have given that rating should they wish to.

## 11) Private Messaging

Precondition:

- Both users are registered

Initiator:

- User

Details:

System should allow two users to send private messages to each other to allow direct contact over the web site.

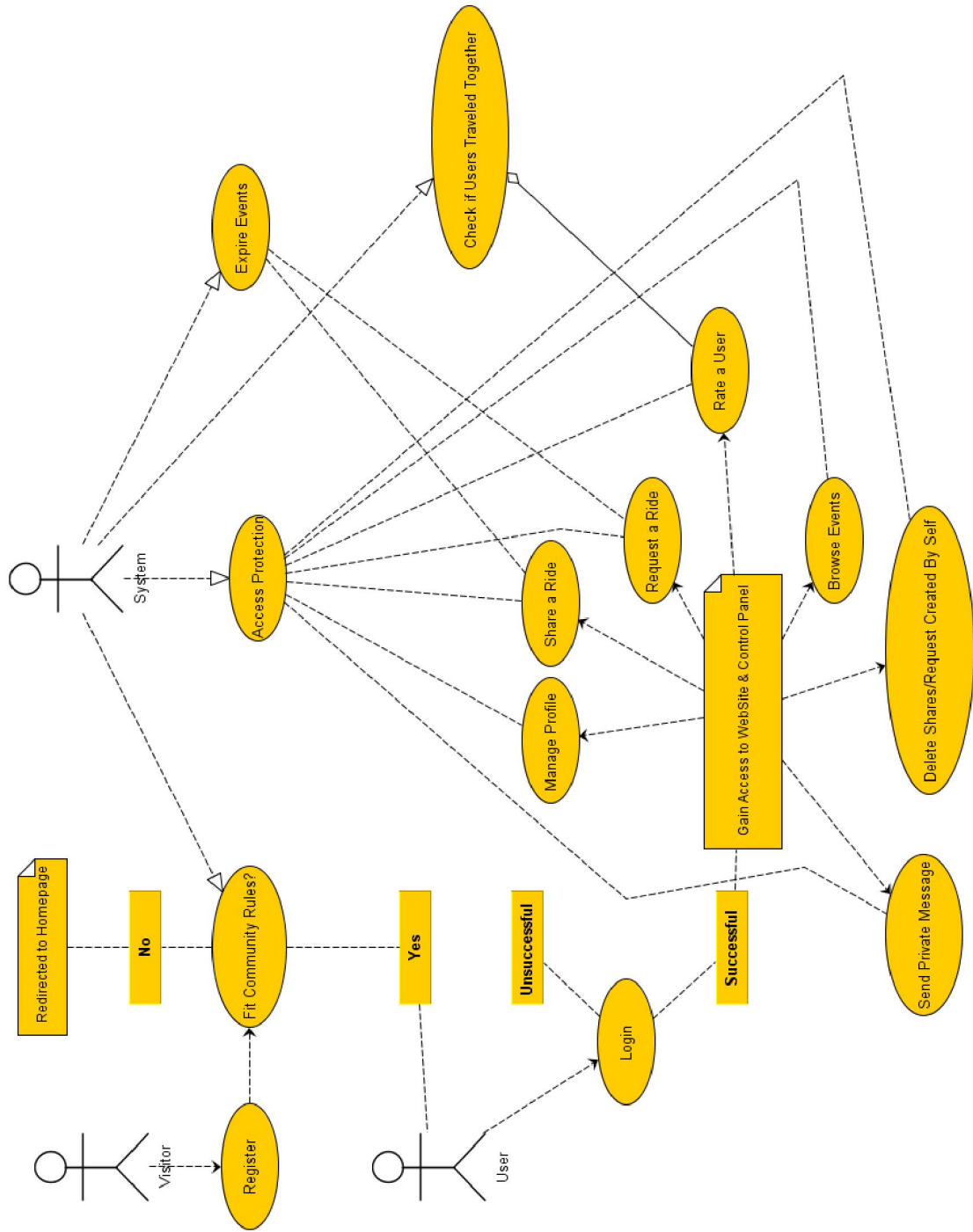


Figure 4. Basic use diagram derived from needed use cases.

## 4.2 TravelBuddy - The Prototype

This section will be discussing the project, taking in account the latest version (0.4b). It will include all details from technical issues, design, usability, responsiveness, use cases to how each choice affects the web success. There will also be ideas for future development and a section briefly discussing what have been done or can be done about the rest of the web site success factors.

### 4.2.1 Technical Decisions

#### 4.2.1.1 Site Hierarchy

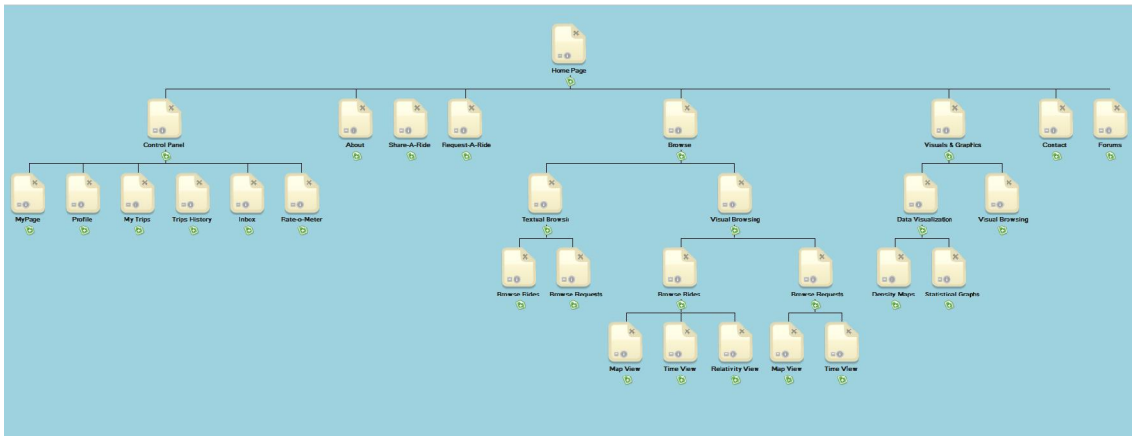


Figure 5. Site Hierarchy created with WriteMaps (<http://writemaps.com/>).

### **4.2.1.2 Database Design**

Database has been designed to accommodate core needs of the system without overloading the system. It is designed in a versatile way so that if needed new tables can be added later on.

There are currently 8 tables in the database,

Users: Holds the basic information of every registered user.

Profile: Holds the detailed information on the user if the user decides to update and add extra information to his/her profile.

Event: Holds the information on rideshare requests.

Pool: Holds the information on shared rides.

TripLog: Holds information on who traveled with whom in which carpooling event.

UserRating: Holds the rating of every user.

Ratings: Holds every rating separately with from who to whom and any additional information or comment entered.

Messages: Holds the Private Messages with from who to whom and the text.

## **4.2.2 Design & Interface**

### **4.2.2.1 Page Layout**

TravelBuddy web site layout is simple and versatile; it contains, a header, footer, content area and a sidebar. The layout is liquid in height at all times and liquid in width depending on resolution. As long as the browser width is equal to or larger than 1000 pixels, width will be static and auto-centered at all times, while height will depend on the amount of content displayed on page. It is worthwhile to mention that necessary precautions are taken to prevent distorted pages by providing a minimum height enforced.

When browser width is below 1000 pixels, which is common for users accessing through smart devices such as Apple i-Phone or Android phones, height dependencies work exactly the same way, however width is adjusted for better viewing and to prevent items going off-screen.

Header is structured in a 2 level format. The top level contains the main navigation of the web site, while the bottom level contains the site title and logo. The placement of logo under main navigation can be considered unconventional, however in this case it contributes to the usability of the web site which will be discussed promptly. It is desired to keep the main navigation in sight at all times. This is important because, if navigation does not stick to the top of the page at all times while a user scrolls down a page, it will become annoying to scroll up every time the user wants to navigate to another page. As the site title and logo should be eye-catching, it is usually designed to be larger than most other elements. This is the case in TravelBuddy web site as well, its height is roughly 2-2.5 times bigger than the main navigation. Thus placing logo/title inside the navigation area would have made the height of the header that sticks to the top, unnecessarily large. It would have caused unnecessary obstruction of space while scrolling. Also, the navigation is desired to be eye-catching and stylish, thus it is designed with large capital letters and has a distinct spacing and hover animation which makes it around 900-950 pixels; as a result it takes nearly the whole width of the layout. Thus placing the logo/title beside it, would have meant making the layout wider than 980 pixels which is the commonly accepted "correct" width for a web site. The placement of menu under logo is not preferred mainly due to 2 reasons. First, as discussed above, it would have obstructed unnecessary space. Second and more importantly, as it is desired to keep navigation fixed at top at all times but not the logo; it would have looked inconsistent while scrolling as the navigation would start 150 pixels below the top and then stick to the top in an instant.

Except a few static pages, the layout includes a left sidebar which will be referred as the "Control Panel" throughout the thesis. It serves as the main login and registration environment as the user first visits the site. After logging in it becomes the second-tier

menu which is specific to the user. It will allow the user to access and manipulate personal data such as profile, trips history, arranged trips. This second tier menu will be discussed in detail in Structure and Infrastructure chapter.

Content area is placed to the right of the left-sidebar. As this is not an informational web site that contains large blocks of text, content area is not statically split into grids and columns. Rather, the column structure depends on the amount and type of content that is specific to each page and it varies from 1 column to 3 columns.

Footer is similar in style to navigation. It is designed to stick at the bottom of the screen at all times. It currently contains the copyright information. In future it is planned to contain some footer quick access links and social icons if there is a need. It also helps keeping consistency and visual unity.

#### **4.2.2.2 Color Scheme & Fonts**

TravelBuddy web site is composed of 1 static color: a light black (#0d0d0d) and 2 varying shades of gray and lime. Prime colors are avoided as their brightness causes flickering on monitors. Black is picked as the entire background color to create a subtle setting where background does not overpower the content, header or footer. For header, text, information blocks' borders, and links that are obvious; shades of gray have been chosen. Shades of lime has been used in elements that were desired to stand out such as footer, in-text links, main headers, and hover actions.

2 different fonts have been used during the design process of TravelBuddy web site. One might discuss, there is a third font which is inside the logo/title; however logo becomes an image during the process of web design and font use in logo is not an issue in its consistency while considering the design of a web site.

As readability is vital on web design, both fonts are sans-serif and chosen from Google's web fonts which are specifically designed for web. Both fonts are backed-up with standard fonts that are considered as web fonts in case of a Google API access problem. However as this is a prototype, fonts are subject to change in future implementations, due to the known fact that Google fonts will decrease responsiveness as it increases the download time of a web page.

Molengo and Arimo have been chosen to be used in main navigation and the rest of web site respectively. Choosing a different font for main navigation is a popular choice because it makes the navigation stand out. Molengo is backed-up with Trebuchet MS and Arimo is backed-up with Arial.

Arimo, which is the dominant font in the web site and appears everywhere except main navigation, is an "innovative, refreshing sans serif font that is very similar and metrically compatible with Arial. Arimo is designed to provide improved on-screen readability characteristics."<sup>3</sup> It claims to solve the need for width-compatible fonts that achieves portability across platforms. Arimo is designed by Steve Matteson.

"Molengo is a multilingual latin typeface that is produced with Fontforge by Denis Moyogo Jacquerye. Its glyphs are designed in a CFF file but the production file is a TTF file with hinting instructions made with Xgridfit."<sup>4</sup> Molengo only appears in main navigation.

A third font will only be seen in the visual interfaces of TravelBuddy, as loading fonts into the application lowers response time of the application which already has to read a lot of data and act on it. Arial will be used in the visual interfaces because of its close relationship to Arimo.

---

<sup>3,4</sup> Information directly quoted from following Google Web Fonts pages:  
<http://www.google.com/webfonts/specimen/Arimo>  
<http://www.google.com/webfonts/specimen/Molengo>



### 4.2.2.3 Navigation & Linking Structure

This section will discuss how the navigation is styled and how it works. Navigational links will be briefly discussed in the context of where they point, but the use of the pages will not be discussed in depth as it will be discussed in "How It Works: Use Cases, Structure & Infrastructure" section.

Main navigation is a top tier vertical bar that sticks to the top of the screen at all conditions. It uses a different, larger font (Molengo, Trebuchet MS) and a different background color to become visible. Other than the hover color change which allows user to more effectively understand which link they are about to click, there is an animated background that stays behind the selected/active link. This provides the user with the feedback of what page they are on, to prevent confusion. Drop-down menu is avoided due to 2 facts, firstly as discussed in preliminary research, users would like to see every option available to them without having to go through the process of hovering over each item. Secondly, there wasn't a need for more items which would have caused unnecessary clutter of information. To avoid stranding the user and increasing usability, main navigation is persistent and appears on every page in the same location, it can not be scrolled out of the screen.

Main navigation consists of Home, About, Share-A-Ride, Request-A-Ride, Browse (which might become Browse Trips in future implementations), Visuals & Graphics, Contact, and Forum. It does not contain any links to personal data which can only be accessed from Control Panel.

Home: Links to the main page. Even though logo is also a link to the main page, as it is scrollable and won't be visible at all times, user must be provided with a link back to the beginning.

About, Contact: These are pretty straight forward. Each respectively links to about and contact pages.

Share-A-Ride: Links to the page where the user can fill out the form to open a carpooling event.

Request-A-Ride: Links to the page where the user can fill out the form to open a request to find a carpooling event.

Forums: This is the only link that opens in a new tab. It links to the phpBB3 forum interface.

Browse: Links to the TravelBuddy Event Browser where the user can choose what kind of event you would like to browse and how.

Visuals & Graphics: Links to the page where there are data visualizations, and graphical charts on active and expired trips.



Figure 6. Main navigation idle (not hovered) displaying what page the user is on.

Control Panel is a left sidebar that gives every registered user access to personal information. It serves as the Sign In area before a user is logged in. After logging in, the rest of the menu appears. Control Panel area is longer than it needs to be due to 2 main reasons; to protect the consistency of height and all new personal features that might be added in the future, will be added here. Items in the menu are;

My Page: Opens the personal page of the user which is a summary of every personal page.

Profile: Opens the user's profile for viewing and editing.

My Trips: Links to the page that displays user's registered and active trips.

Inbox: Opens the user's inbox.

Trips History: Links to the page that displays user's past events.

Rate-o-Meter: Links to the page that displays ratings about the user and everyone else.

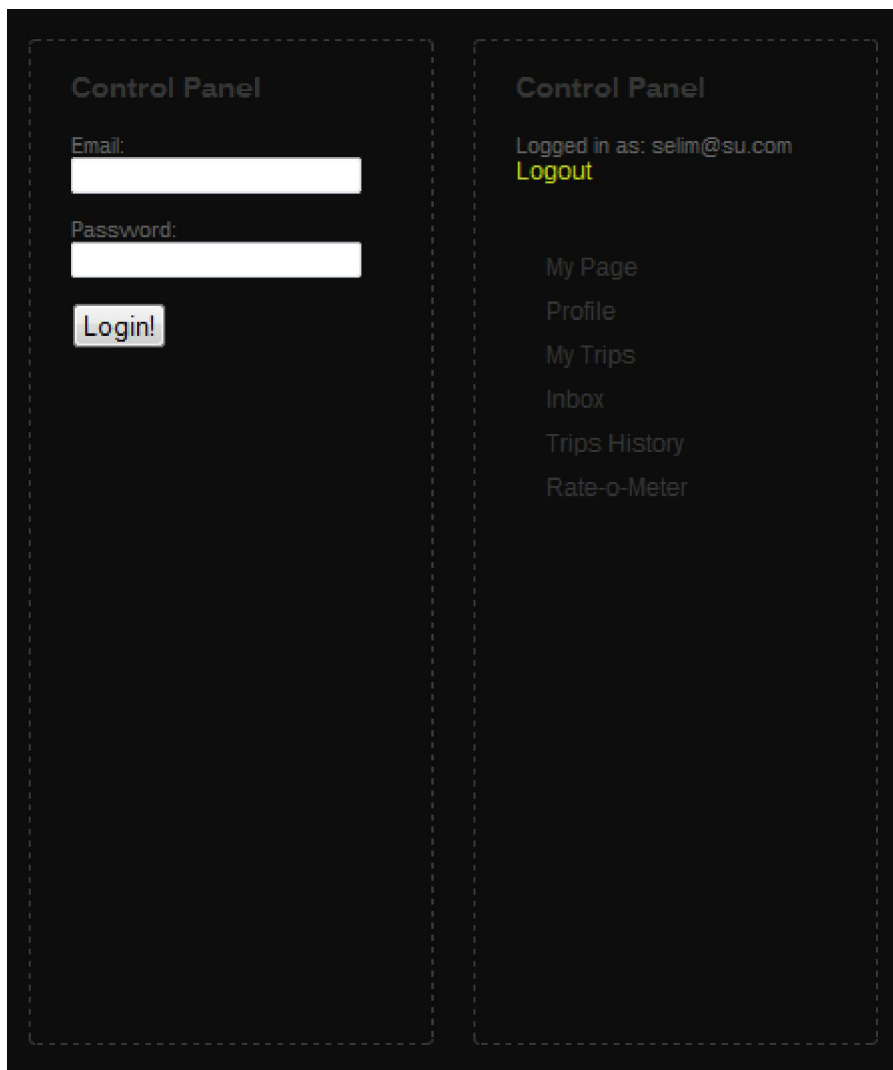


Figure 7. Both stages of control panel. Second one displays personal menu items that appear after successfully logging in.

Send-PM, and Search User are the first two planned items to be added to this list as quick access to these items is thought to be useful.

Linking structure is forthright and simple. Most links are self-explanatory, titles for links are specifically selected to be unconfusing. Links that are images or links that might have been confusing just as text are supported by each other. Links never open in new windows, and only in new tabs if the link is pointing to an external location with the exception of Forum in Main Navigation which will be discussed later on. Some links do make use of the iframe property with the stylish interface of FancyBox JQuery plugin. One or more of the following reasons were behind in the use of FancyBox;

- The directed page runs multiple scripts and applications, thus does not have a relevant web site interface.
- The page requires a much larger or much smaller content area.
- The page is only a simple form that does not require an entire web page layout.
- The direct url is not desired to be displayed.

#### **4.2.2.4 Content & Content Placement**

As discussed before, the content does not have specific style. The site is not informational thus contains minimal amount of text with the exception of About and Instructions sections which are specifically designed to have tight columns for easier scannability. Content includes pieces of short explanatory text, lists of information, information boxes, and tables. As a result of this varying content, no strict rules have been put in place for a columned structure in the content area. Entire textual content uses a gray color, and 12 pixel Arimo font. Line-heights have been properly adjusted for better readability. All image-links (unless hovered over), large tables and information lists are in duller colors compared to main colors of the web site; to prevent the banner blindness effect the brighter colors might cause.

#### 4.2.2.5 Tables & Forms

Tables play a very vital role in the flow of TravelBuddy. As textual data retrieved from the database should be presented as lists of information in many different cases such as displaying events, users, rating rankings which users would like to browse and analyze.

All tables are relative to the page they are displayed in. If there is more than one table in a page, each table has a descriptive title that is placed at the top of the table. Data provided is selected carefully from the database and summarized to provide the users with what they really need to see, rather than displaying everything and causing visual clutter. Zebra stripping is used to make table readability and scannability easier. All the text is left aligned for visual consistency.

Two kinds of tables are used depending on the situation;

1) Static Tables: Used in places where limited information is desired to be displayed such as three latest trips. These tables directly access the database, retrieve the required information, and display it in a non-sortable, non-searchable fashion. The name static should not be confused with the HTML meaning of static, in that sense these tables are still dynamic, the only reason why they are called static is that they can't be sorted, or filtered.

2) Dynamic Tables: Provided with DataTables JQuery plugin, these tables are used in places where large amounts of information with no certain limit is desired to be displayed such as browsing events. They work in similar ways to static tables, however they have both advantages and disadvantages when compared to static tables. On the positive side, these tables allow sorting and searching(filtering) which is why they are preferred on pages such as event browsing. For example, a user might want to sort events in order of closest time, or might want to search and display events for only a specific location; these are possible in dynamic tables. However on the downside, dynamic tables do a lot of

continuous database querying, which hampers the speed, thus lowers responsiveness. This is the main reason why they are not preferred in every page a table is required.

Enter ID of a ride in the box below to register

Show  entries Search:

ID	By User	Location	Destination	Capacity	Payment	Date/Time
1	selim@su.com	P6	BAKIRKOY	0	5	2011.07.13 12:43
2	nazli@su.com	P7	CADDEBOSTAN	1	0	2011.07.14 21:12
3	aysem@su.com	P3	BAKIRKOY	0	5	2011.07.14 09:11
4	doruk@su.com	P12	USKUDAR	3	10	2011.07.13 11:54
5	metin@su.com	P13	BESIKTAS	0	0	2011.07.16 12:12
6	ekin@su.com	P8-1	ERENKOY	3	4	2011.07.12 23:44
8	onur@su.com	P12	LEVENT	4	6	2011.07.14 03:05
10	izi@su.com	P6	BAKIRKOY	3	6	2011.07.13 02:05

Showing 1 to 8 of 8 entries (filtered from 10 total entries) ◀ ▶

Figure 8. Dynamic table showing active rideshares.

TravelBuddy makes use of a lot forms as the system basically depends on interacting with and retrieving information from users. All event registrations, site registration, profile and user interactions are provided through forms.

Forms are designed to be clear and simple. All fields that should be filled have light gray backgrounds so that they stand out from the black background. Labels are concise and self-explanatory. They are placed above the information fields and left aligned. When a field is chosen, the outer background around the chosen field is highlighted and if needed a short explanatory text is displayed to the right of the field. These are provided to improve usability and improve user experience, as they improve the form's understandability. Control buttons are given meaningful titles rather than keeping the standard "Submit Query", or commonly preferred "Submit", in the attempt to make users more aware of what their action will do.

**Share-a-Ride!**

Please fill in the details correctly and click share-my-ride to create an event as a driver.

Location

Destination

Where will you be heading?

- Taksim
- Kadikoy
- Besiktas
- Bakirkoy
- Uskudar
- Sariyer
- Erenkoy
- Caddebostan
- Levent
- Sisli
- Tuzla
- Ankara
- Izmir
- Bursa
- Antalya
- Other District
- Other City

Figure 9. Form for sharing a ride.

#### 4.2.2.6 Other Visual Elements

The only visual elements other than the visual interfaces, data visualizations and graphical charts that are part of the application process written with Processing which are going to be discussed in their own section; are the image-links that appear on TravelBuddy Event Browser page and Visuals & Graphics page. As they are not button-like in structure, they are made large enough to catch attention and have hover effects to implicate they are links.

### **4.2.3 How It Works Considering Usability & Responsiveness: Use Cases, Structure & Infrastructure**

This section of the thesis will discuss how the project works, what are the use cases and how they are provided. It will discuss the structure and infrastructure of the pages in respect to usability and responsiveness. There will also be ideas on planned improvements.

#### **4.2.3.1 General Details on Back-End Infrastructure**

The back-end of TravelBuddy has been coded with PHP from scratch. Content Management Systems such as Drupal, or Joomla haven't been used because they have proven to be making too much unnecessary database access for unused features. This was resulting in slower system and hampered responsiveness. A pure PHP system built for only the required functions was a key factor in improving responsiveness.

#### **4.2.3.2 Home Page, Login & Registration**

Home Page is amongst the few pages that can be visited without a registration, and is the only page that has the control panel among those pages. Home Page can be accessed through any page by using the Home option on the main navigation or by clicking the logo. It is the main page of the web site and any visitor attempting to access any other page will be redirected here, this is valid for any unlogged but registered user as well.

Home Page content area starts with a classic welcome message, and then the user is presented with links to registration and signing-in. The text includes the warning message that this is a closed-community web site and therefore only certain people who meet the requirements can sign-up to use the web site. The text continues with links to Instructions,



About and Contact which are important for a new user and are the only other pages that don't require signing-in to view.

Control Panel on this page, first checks if the user is logged in which is determined by a value named "loggedIn". This value can either be 0 or 1, which refer to not logged-in and successfully logged-in respectively. If the user isn't logged in, the panel displays a simple login form that requires a registered e-mail and a password. After user fills the form and clicks submit, form is controlled by login-handler which connects to the database and checks if the information is correct. Depending on the query, login-handler tells the control panel to either change "loggedIn" to 1 for correct credentials or to keep 0 value and continue displaying login form along with an error message.

A visitor fitting the requirements of the community can select the register option, to view the registration form through the iframe window. Registration is not an option in the main-navigation as this web site does not contain any useful information for someone outside the directed community. This is also the reason why registration is not displayed on a separate page and iframe is preferred. When user fills the form and clicks submit, registration-handler first checks the form if all the mandatory fields are filled, before querying the database so that database isn't filled with useless information. If all the mandatory fields are filled, than registration-handler will query the database to see if there is any duplicate e-mail or student number already registered, if the query returns no results, it will insert the data into Users, Profile, UserRating, and phpBB3 Forum Users tables. Profile table contains extra information fields that later on can be edited from Profile controls, data entered to the table through registration are only the basic details which are not permitted to be changed later on such as name, surname, e-mail, and student number. UserRating table is inserted with the userID (in this case e-mail, because it has to be unique), and 50 is given as the starting neutral rating as rating ranges from 0 to 100. On successful registration, user will be notified that they can close the frame and login from the control panel. Otherwise, the form will be displayed along with an error message that the registration has been unsuccessful.

If "loggedIn" becomes 1 and the user is successfully logged-in, then the control panel starts displaying the personal access menu which currently contains My Page, Profile, My Trips, Trips History, Inbox, and Rate-o-Meter. It will also contain Send PM and Search User with version 0.5b.

Home Page also contains 3 data blocks, named Latest News, Latest Users, and Top Users. Currently 2 of these blocks are dynamic and 1 is static. Latest news is the static block and will only be made dynamic if the project is marketed in the future; it will then be controlled through an admin panel. Latest Users is a dynamic block that connects to the database and retrieves the latest 5 registered users. Top Users is a dynamic block as well which works similarly to Latest Users, it retrieves the top 5 users with the highest ratings.

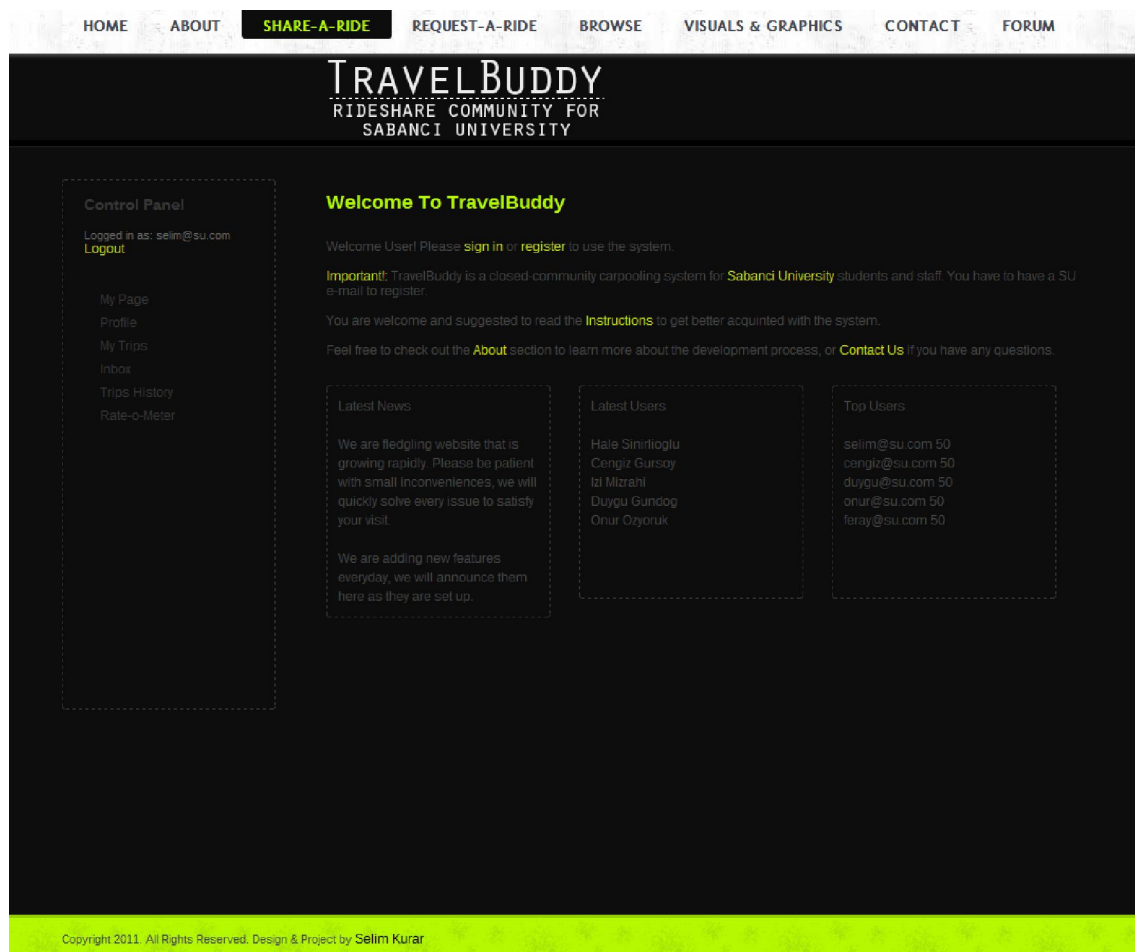


Figure 10. Homepage for TravelBuddy.

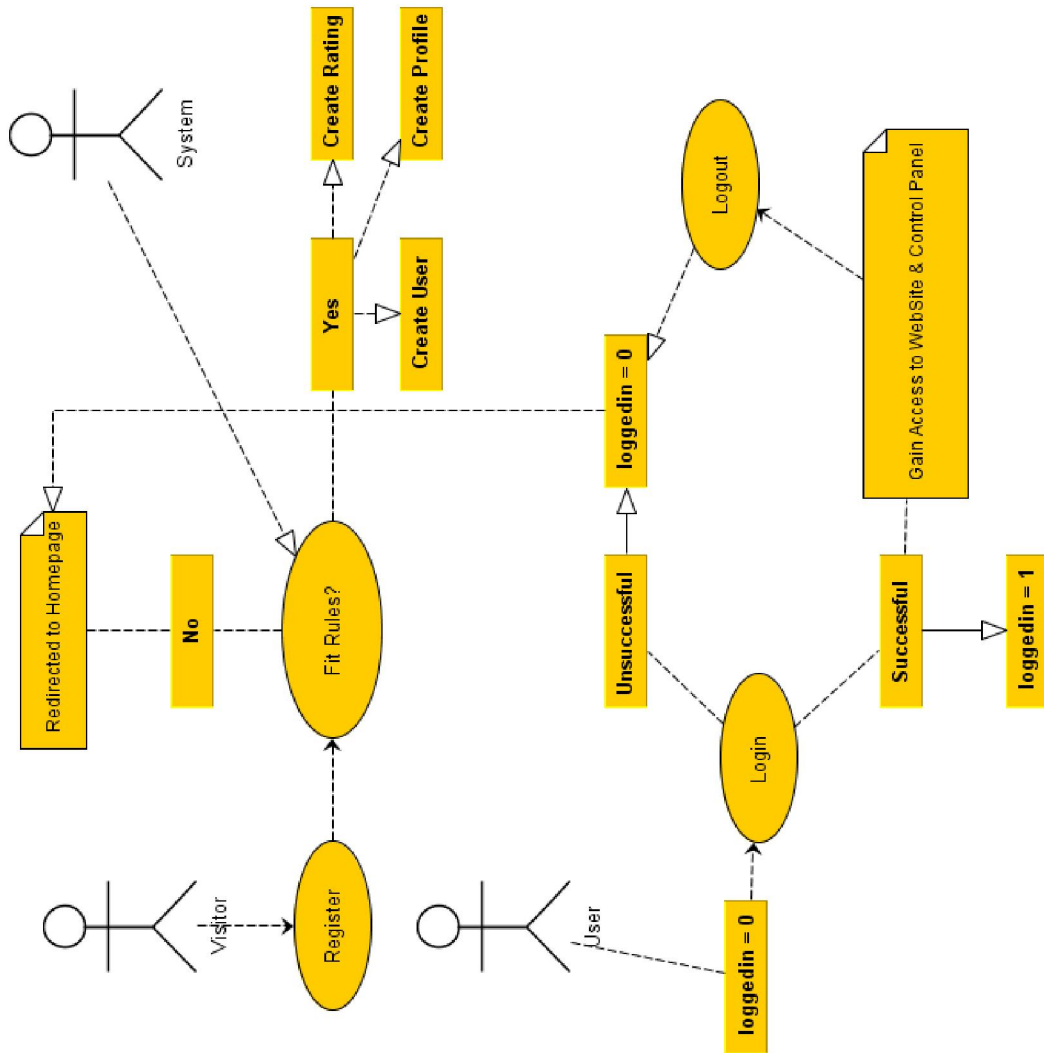


Figure 11. How Login and Registration procedures work.

#### 4.2.3.3 Creating Events: Sharing a Ride & Requesting a Ride

To create either a Share-A-Ride Event or Request-A-Ride event, user must access the relevant page through the main navigation. Both pages require the user to be registered and logged-in, otherwise they will redirect the user to the Home Page. Both pages consists of simple and easy to understand forms with proper labels, selection highlights, and if needed helpful texts.

On Share-A-Ride page, user is required to fill the following fields; Location, Destination, Date, Departure Time, Capacity and Payment. All of the fields are mandatory except for Payment which allocates 0 by default if nothing is entered. A field for smoking or non-smoking passengers is under consideration and might be added in future versions. To prevent ambiguity and ensure the proper functioning of visual interfaces; locations and destinations are limited to certain lists. Locations for sharing a ride is limited to car parking areas. After the form is filled and submitted, pool-handler checks if all the mandatory fields are filled, also performs checks on fields such as Departure Time to protect the system from false queries such as entering 25 for hour or 75 for minute. If everything is in place, then pool-handler inserts into Pools table all the details that have been retrieved, it also connects to the UserRating table and retrieves the current rating of the user, to add it into the Pools table. Another important action performed by pool-handler is to add the event into the TripLog table along with the trip's original event ID.

Request-A-Ride page is similar in many aspects to the Share-A-Ride page. It requires fewer details to be entered; Location, Destination, Date, Latest Departure Time. Request-A-Ride locations are limited to buildings including Dormitories, Faculties, Staff Houses, and some other major buildings. When submitted, this form uses event-handler to check fields, inconsistencies, and inserts the information into Event table with the current user rating if all requirements are met.

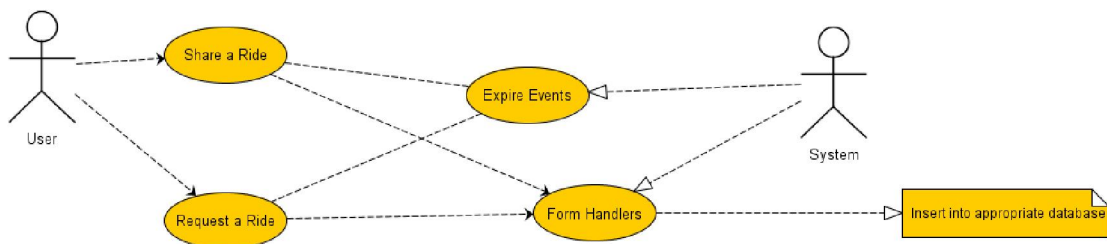


Figure 12. Creating an event.

HOME ABOUT **SHARE-A-RIDE** REQUEST-A-RIDE BROWSE VISUALS & GRAPHICS CONTACT FORUM

# TRAVELBUDDY

RIDESHARE COMMUNITY FOR SABANCI UNIVERSITY

**Control Panel**

Logged in as: selim@su.com  
[Logout](#)

- [My Page](#)
- [Profile](#)
- [My Trips](#)
- [Inbox](#)
- [Trips History](#)
- [Rate-o-Meter](#)

## Share-a-Ride!

Please fill in the details correctly and click share-my-ride to create an event as a driver.

Location

Destination

Date

Departure Time

Capacity

Payment

[Share-My-Ride](#)

Figure 13. Page for sharing a ride.

HOME ABOUT **SHARE-A-RIDE** **REQUEST-A-RIDE** BROWSE VISUALS & GRAPHICS CONTACT FORUM

# TRAVELBUDDY

RIDESHARE COMMUNITY FOR SABANCI UNIVERSITY

**Control Panel**

Logged in as: selim@su.com  
[Logout](#)

- [My Page](#)
- [Profile](#)
- [My Trips](#)
- [Inbox](#)
- [Trips History](#)
- [Rate-o-Meter](#)

## Request-A-Ride!

Please fill in the details correctly and click request-a-ride to create an event as a passenger.

Location

Destination

Date

Departure Time

[Request-A-Ride](#)

Figure 14. Page for requesting a ride.

#### **4.2.3.4 Browsing Events**

To browse both active rideshares and rideshare requests, a user should select Browse from the main navigation which will open the TravelBuddy Event Browser page. Event Browser page is only accessible after successfully logging in. This page allows users to browse through events in several ways; these options can be collected in two main types, Textual Browsing and Visual Browsing. Browsing rideshares in both methods and all types allows the user to register for any active carpooling event if they wish to. The page is structured with nice, large graphical links and concise titles for these links.

##### **4.2.3.4.1 Textual Browsing**

User can opt to choose between browsing shared rides or requests. Both links will open in iframe windows. There are several reasons why the two different types of events are displayed on two separate windows rather than one single window;

- To prevent user confusion about what they are browsing.
- To make less queries to the database at one time to improve responsiveness.
- To only show the user what they want to see.

When user chooses to browse shared rides, an iframe window with DataTables opens up. In this table all the currently active carpooling events from the Pools table are listed. Initially there are 10 events listed per page and user can use the next and previous buttons under the table to browse the rest, or can choose to display more than 10 events per page from the selector on top left of the table. Users are given the option to sort the list by any information field, both ascending and descending. The table also allows users to search and filter the table by using the search box on the top right of the table. To minimize queries and improve responsiveness, there is an added delay script to the search function. This function waits for a certain amount of milliseconds after keystrokes so that not every key pressed will be send as a query to the database and thus hamper the system greatly.

Registering for a carpooling event is very easy and can be done while on Browse Rideshares iframe. As indicated on the top of the iframe, the only thing user has to do, is to choose from the table which event they want to join and enter the ID of the event into the box. When the user clicks Register, registerthis-handler first checks the entered ID in the Pools table to see if the event is currently active and has available seats. If both variables are verified, then first registers the users into the relative row of the TripLog table and then decreases the available capacity of the vehicle by 1.

Initially making a browsing interface for requests might seem unnecessary, but a user who is about to create or has a currently active carpooling event, might want to check if there are any people who might be requesting a ride to a similar destination. This would allow them to contact each other. On the back-end, browsing ride requests works exactly the same way as browsing shared rides, it dynamically accesses the Event table and retrieves the active requests.

The screenshot shows the TRAVELBUDDY website interface. At the top, there is a navigation menu with links: HOME, ABOUT, SHARE-A-RIDE, REQUEST-A-RIDE, REQUESTS, VISUALS & GRAPHICS, CONTACT, and FORUM. The main content area features a search bar and a table of ride requests. The table is titled 'Show 10 entries' and has a search input field. The table columns are: ID, By User, Location, Destination, and Date/Time. The data rows are as follows:

ID	By User	Location	Destination	Date/Time
3	system@su.com	I4	BESIKTAS	2011.07.12 22:56
4	doruk@su.com	A5	CADDEBOSTAN	2011.07.12 23:23
5	motin@su.com	B3	NIKARA	2011.07.12 23:46
6	ekin@su.com	A6	JSKUDAR	2011.07.13 13:24
8	onur@su.com	I3	BESIKTAS	2011.07.13 08:12
9	nihat@su.com	A4	BESIKTAS	2011.07.12 23:57
10	su@su.com	I4	BAKIRKÖY	2011.07.13 01:34

Below the table, it says 'Showing 1 to 7 of 7 entries (filtered from 10 total entries)'. There are also pagination controls.

Figure 15. Textual Browsing for Requests viewed in iframe.

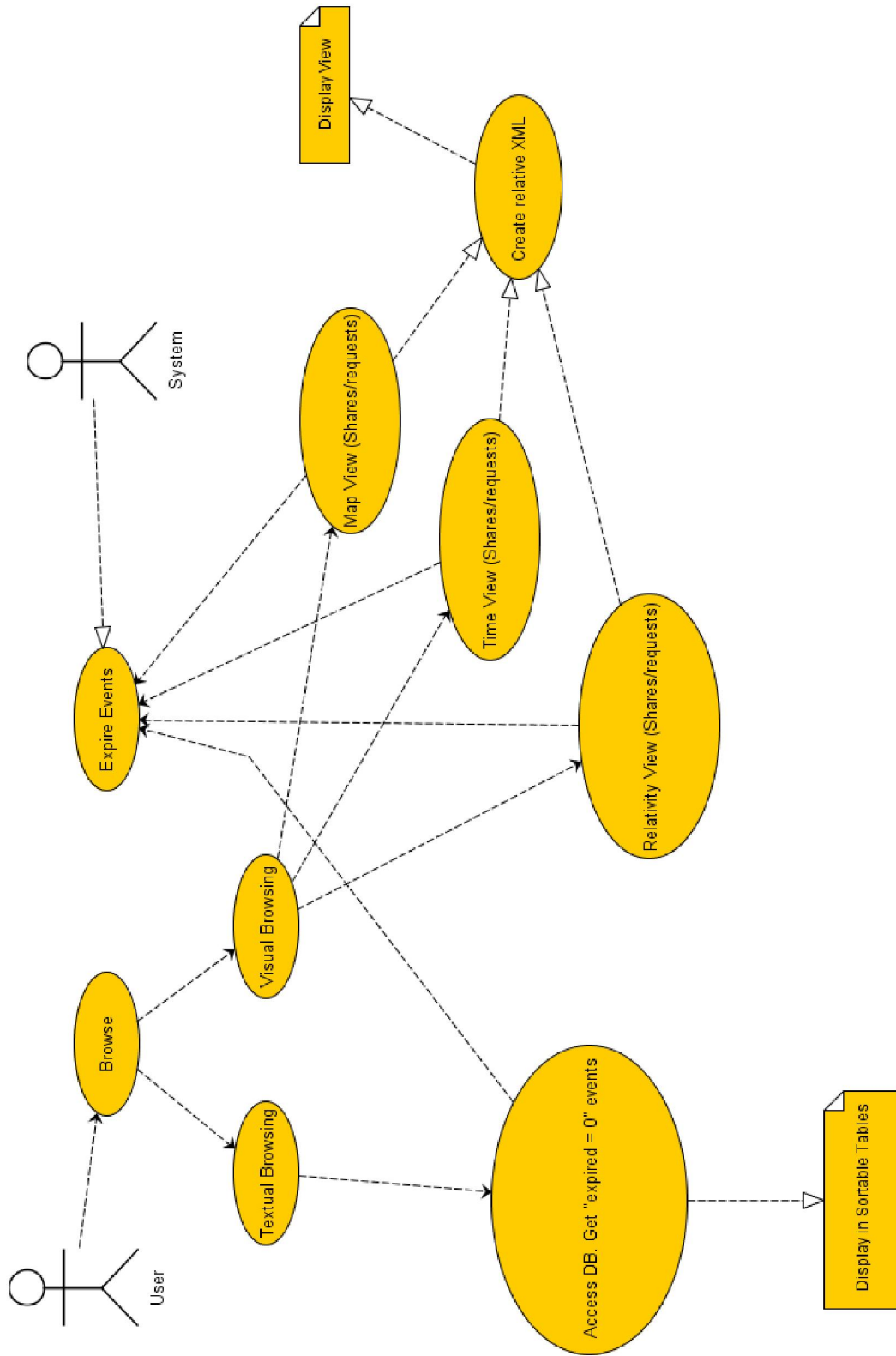


Fig16. Diagram for browsing events.



#### 4.2.3.4.2 Visual Browsing

Visual browsing is again split under two main types, rideshares and requests. This was crucial because all options under visual browsing require heavy database querying, list creations and visual sketching. All options are actually stand-alone applications written with Processing, that have some similarities and some differences. Visual browsing for rideshares and requests have 2 common options; Map View and Time View. A third option, relativity matching is only available for browsing rideshares. In all visual browsing options, requests appear as the classic thumbs-up hitchhiking gesture, and rideshares appear as a car. The color of the event icon depends on the rating of the user. Neutral ratings get black icons, negative ratings get red icons as red is perceived as a warning color, and light-blue is for positive ratings as light-blue is usually paired with innocence. In prototype v0.3b, every image used to have destination written above them, but it has been decided that it wasn't visually pleasing so it has been removed in the current version 0.4b. Displaying only the first letter of destination is considered for v0.5b but has not been decided yet. Clicking on any of these icons brings up the contact card of that event generated with the information retrieved from the database. In both rideshares and requests, contact cards display the user e-mail, event ID, time, date, user rating, location and destination. Contact cards in rideshares additionally display capacity and payment. Contact cards can be closed by clicking anywhere on the contact card as stated in the legend.

All visual browsing interfaces run through iframe windows, mainly because all require larger display areas due to number of items they have to draw. Making these display areas smaller would have enforced smaller visuals, clutter and harder to click icons which would have decreased usability.

When any visual browsing interface is launched, it first runs either the create-poolsXML which creates a XML document from the current active rideshares or create-eventsXML script which creates a XML document from the current active requests depending on the selection user makes. These XML documents are then used by Processing

to draw the images at correct locations, do calculations and display relevant information. Unlike textual browsing, it is decided to only display the current day's events. This is done by checking the event dates while the XML creating script runs and only creating the XML file with the events of today's date. This would improve the responsiveness as less information will be queried from the database, thus resulting in less draw processes and faster displaying of results. Also, unlike the demo which uses a couple of users, what would happen if 1000 people use the system at the same time has been considered as well while taking this decision. If all created events were to be displayed without filtering for date, the interface would highly likely become a visual clutter as well as either taking too long to load or crashing entirely.

Registering for a carpool event in Visual Browsing for Rideshares works exactly the same way as it is in Textual Browsing for Rideshares. Again the only thing user has to do is to enter the correct event ID into the box and click register.

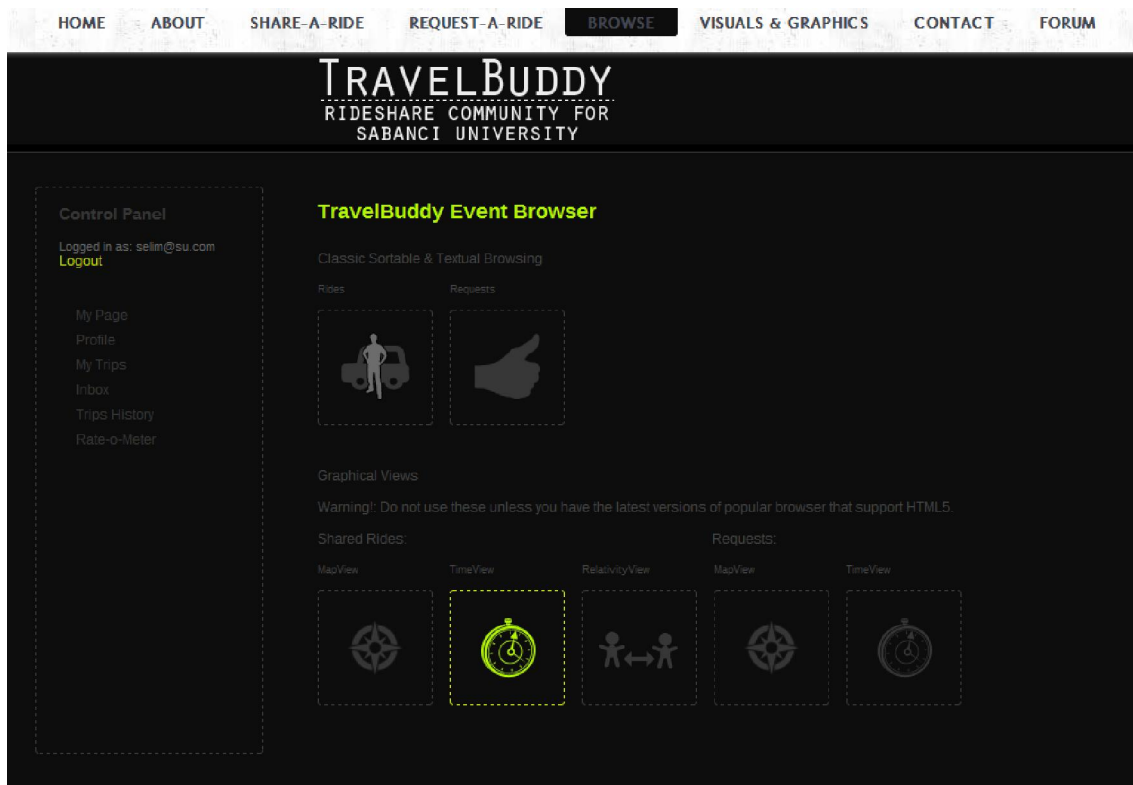


Figure 17. TravelBuddy Event Browser page.

## 1) Map Views

A simple map displaying all dormitories, faculties, staff houses, car lots and all major buildings have been created from the actual school map. There are several reasons why the real school map hasn't been used;

- The original map is too cluttered, small, and visually unappealing.
- User perception is vital in understanding interfaces, especially when a map or sense of direction is involved. A small questionnaire that I have conducted among the graduates that have studied in this university and lived in the dormitories for at least 4 years have proved my personal suspicion. Unlike the original map provided by the school's website, people perceive the president house as the top-point, the north-end of the school; and the gate as the south-end of the school. The map is derived depending on this perception.

Every type of building and all viable car parks are color coded and labelled with the names they are originally referred. These color codes and what they refer to, is presented to the user in the top left legend area.

When the application is launched, the relevant XML is generated and depending on the selection. Then the application creates a users array and stores the data of each user temporarily for the duration of that application window. This prevents errors from occurring if someone else launches the application while the user is browsing which creates a new latest XML file. The application, then uses the users array to place the events at correct locations including their relevant contact data.



Figure 18. The school map.



Figure 19. Portion of school map displaying requests, contact card opened.

## 2) Time Views

Time Views work similar to Map View in many aspects, but the main difference is that it creates time-zones depending on departure times and places the events into the relevant time zone. Time-zones are limited to a maximum of 2 hours, as it is a fact that only a user looking for an urgent ride or ride-partner would decide to browse events only depending on time. Also as this View has a smaller display area compared to Map Views, this works as a precaution toward preventing visual clutter.

Time-zones are represented by rectangular boxes of varying shades of gray that start very small from left-top of the screen and grow larger towards the right bottom. Each section's time interval is given at the left bottom corner of the corresponding box. Left-top is chosen as 0 minute point because left-top is always considered as a starting point in web design as it is always visible on initial start. Boxes grow in size toward the right bottom

because time intervals are longer for those boxes, thus there is a much higher chance of those boxes including more events.

The working of the user array and XML logic is very similar to Map Views as well. However this time, application is responsible for picking through the XML data and only selecting events that have equal to or less than 2 hours left for departure. After the application processes which event falls into which time-zone, it starts a randomization and origin point checking process to prevent to events being placed on top of each other. It is thought that this process still has potential and room for improvement, but as it has minimal importance in the generality of the project, it is not considered in the definitive improvement plans for future work.

### 3) Relativity View

Considered as the flagship of the views and the most complex one amongst them. It is only available for viewing active Rideshares as there isn't an adequate number of variables to create one for viewing Requests, and thus it is decided that it would be redundant.

Relativity View visually looks similar to Time View as it creates zones; however this time, zones refer to quality points of events. It is a way of letting the system decide which rideshare options are the best suitable ones and then allows the user to pick regarding where they want to travel.

Event quality points are calculated by the application. Quality points are given depending on the following attributes of the event;

- Time: Closer the departure time, higher the quality point.
- Driver's Rating: Higher rating means, higher quality point. Low ratings have negative quality point penalty.

- Payment: Free rides get the highest quality rating from this section. Higher the price, lower the quality point.
- Driver's Successful Previous Trips: Number of times the user has successfully shared his/her ride increases the quality point.
- Capacity: Higher the capacity means there is more space to sit, no one would prefer to be crammed in the back of a car with 2 other people. Thus higher the capacity higher the quality point.

Attributes do not carry the same weight, nor give same amount of points depending on the levels of attribute quality. Each attribute has different maximums and minimums that are carefully selected to balance out each other, also some attributes carry negative effects. The weights and values will not be given here as they can be subject to change, as there is a possibility of new attributes being added with newer version of the prototype.

The only downside of this View is; it needs many active users to use the carpooling system for a while, so that a proper database of ratings, and previous trips could be built up. It is decided that if this system is implemented in the future, Relativity View should be locked and invisible for a certain amount of time until a required mass of data is produced. Only after then, it would be truly functional. The only reason it is in the prototype right now is for test purposes.



Figure 20. Relativity View.

#### 4.2.3.5 Visuals & Graphics: Data Visualizations

Visuals & Graphics page can be accessed from the main navigation menu, and is again protected against unregistered users. This page is designed and structured very similarly to TravelBuddy Events Browser page. There is a very short informative text, and a number of large image links, labeled according to what they represent. Even though this page is created to access data visualizations, it also contains the links to visual browsing



interfaces as well due the page's title and that they fit the context. However they will not be discussed in this section as they launch the same applications that have been discussed in the previous section.

This page and all the Processing applications in it, are purely designed to be informative if the user is interested. It gives statistical visualizations depending on shared rides, requests and past events. Currently there are 5 visualization applications that can be run through this page. They can be collected under 2 main types; Density Maps and Destination Charts, which have their own separate sub-types. Data retrieval works the same way as it works in Visual Browsing, when the application is launched, it runs the relevant php script to generate a XML file from the database.

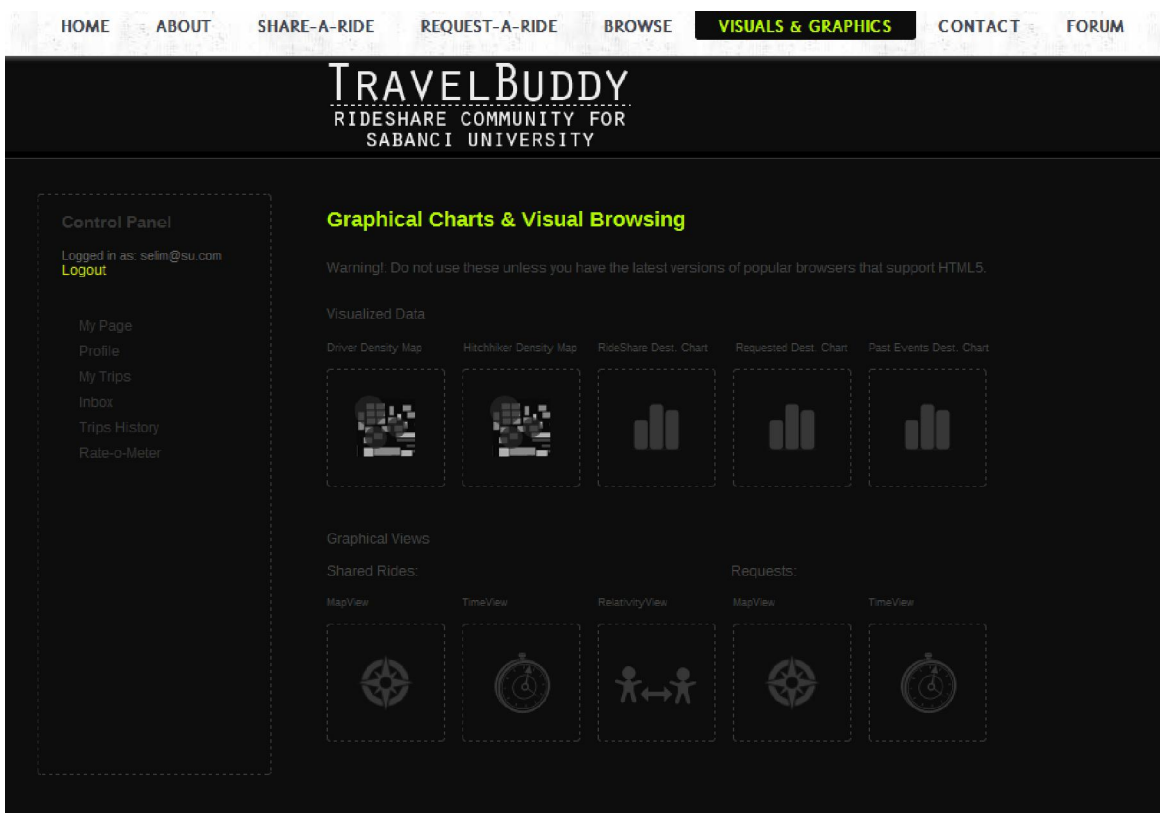


Figure 21. Visuals & Graphics page.

## 1) Density Maps

There are 2 density maps, one for rideshares and one for requests. Each works on a similar interface to Map Views. However this time the data is used to calculate which areas have more events currently active. The map is not meant for browsing events, thus does not display any events but draws red transparent circles on areas where there are active events. The radius and the brightness of the circle changes depending on the event density in that area. As the number of events increase in an area so does the radius of the circle and it gets a darker color.



Figure 22. Mock up density map as the system doesn't have enough data to create one.

## 2) Destination Charts

Destination Charts create statistical graphics that depend on the projected destinations of active or completed events. There are 3 different chart application the user can launch. These are Rideshare Destinations Chart, Requested Destinations Chart, and Past Events Destinations Chart. While Rideshare Destinations and Requested Destinations provide charts from currently active events; Past Events Destinations Chart creates a graphical chart from completed events.

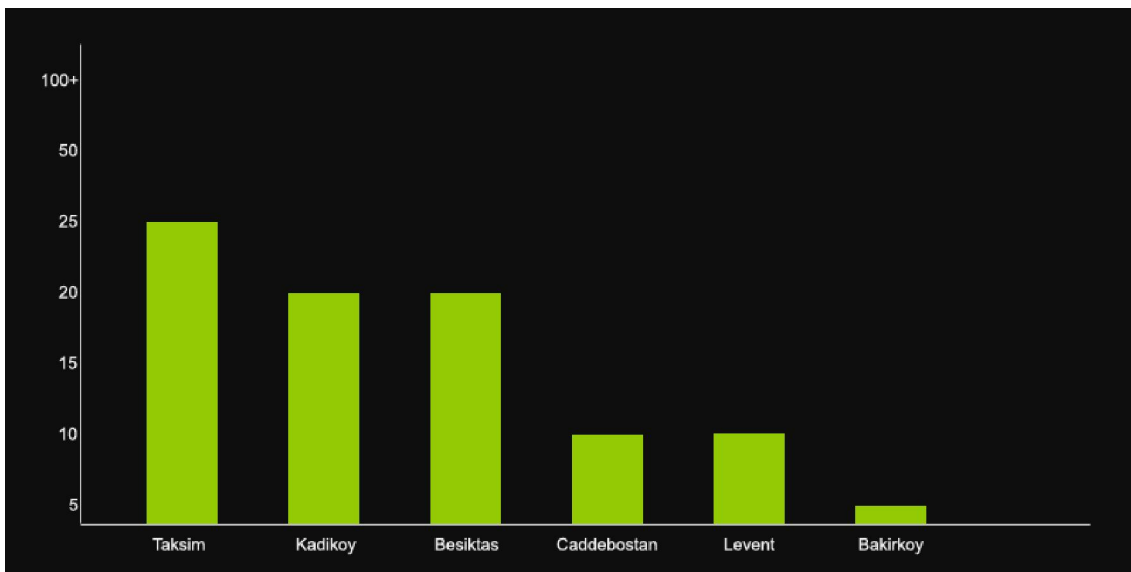


Figure 23. Mock up destination chart as the system doesn't have enough data to create one.

### 4.2.3.6 Personal Pages

These are user specific pages that are only accessible after logging in to the system. The links to these pages are placed in the Control panel and will only be visible after successful login.

## 1) MyPage

MyPage is like a summary page of all the other personal pages, and contains inner-links to most of these pages. It is structured with dynamic data-blocks and 4 static tables. Data blocks consists of Profile Summary (named Personal Details), Ratings, Latest Rating Received and Latest Rating Given.

Personal Details block displays some main information from your profile and contains a link to access the rest of the profile.

Ratings block shows your current rating and provides links to view other people's ratings and rate someone.

Latest Rating Received displays the latest rating you received, from whom and their comment.

Latest Rating Given displays the latest rating you have given, to whom and your comment.

Below the data blocks, there are 4 static tables;

i- Your upcoming ride that you have requested to join: Displays the closest active carpooling event that the user has registered as a passenger, if there is any. Below the table, there is a link to My Trips page where the user can view all his/hers registered and upcoming events and unregister from any should they want to.

ii- Your upcoming ride that you declared to share: Displays the closest active carpooling event that the user has created as a driver, if there is any. The link below the table takes the user to My Trips page for the same reasons as stated above.

iii- Your latest requests: Displays two of the most recent, completed carpooling events that the user has registered as a passenger, if there is any. Below the table, there is a link to Trips History page where the user can view all his/hers completed events and rate users whom they have traveled with.

iv- Your latest rideshares: Displays two of the most recent, completed carpooling events that the user has created as a driver, if there is any. The link below the table takes the user to Trips History page for the same reasons as stated above.

## 2) Profile & Profile Management

Profile page which can be accessed from Control Panel, displays the profile details of the user. Currently, profile page displays Name, Surname, StudentNumber, Gender, Age, Location, and Extra Information. E-mail has also been decided to be displayed, and will be in v0.5b. All the information is dynamically retrieved from the Profile table. Name, Surname, StudentNumber and E-mail are initially inserted into this table during the user's registration and will be visible when the user accesses the Profile page for the first time. This information can't be changed or edited due to liability reasons. The rest of the information can be edited at all times using the Update Profile link at the bottom right of the profile page. This link opens an iframe window similar to registration window containing a form with all details. After the user fills any details he/she desires to update and clicks Update; update-handler finds the relevant row in the Profile table and updates the row with the entered information. Upon successful update, user is presented with a success message.

### 3) MyTrips: Currently Active Events & Unregistering Events

MyTrips, also called the My Planned Events page displays all of the user's registered and active carpooling events and contains a form that allows unregistering from any event. At the top of the content area is the form for unregistering events. The form works similarly to registering events but is much more complex as it has to access multiple tables and clean all the inserted data. When the user enters the ID of the trip he/she wants to unregister and clicks Unregister, unregisterthis-handler first checks the TripLog table to see if the user is actually registered in that event and checks to see if the user is a passenger or driver. If the user is confirmed as a driver which means he is the creator of the carpooling event, then the entire TripLog row is deleted as the event is cancelled and the event is also removed from Pools table. If the user is confirmed as a passenger, then only his/her ID is removed from the relevant cell but the rest of the TripLog row stays intact as the event isn't cancelled, user's request is also removed from the Event table.

There are two static tables after the unregister event form. These tables are;

- i- You arranged to join: This table retrieves the data from TripLog for any active carpooling events that the user is currently registered as a passenger. Table displays the trip's ID, driver, other passengers (if there is any), departure date/time, and destination.
  
- ii- You arranged to share your ride: This table retrieves the data from TripLog for any active carpooling events that the user has created as a driver. Table displays the trip's ID, passengers that have registered (if there is any), departure date/time, and destination.

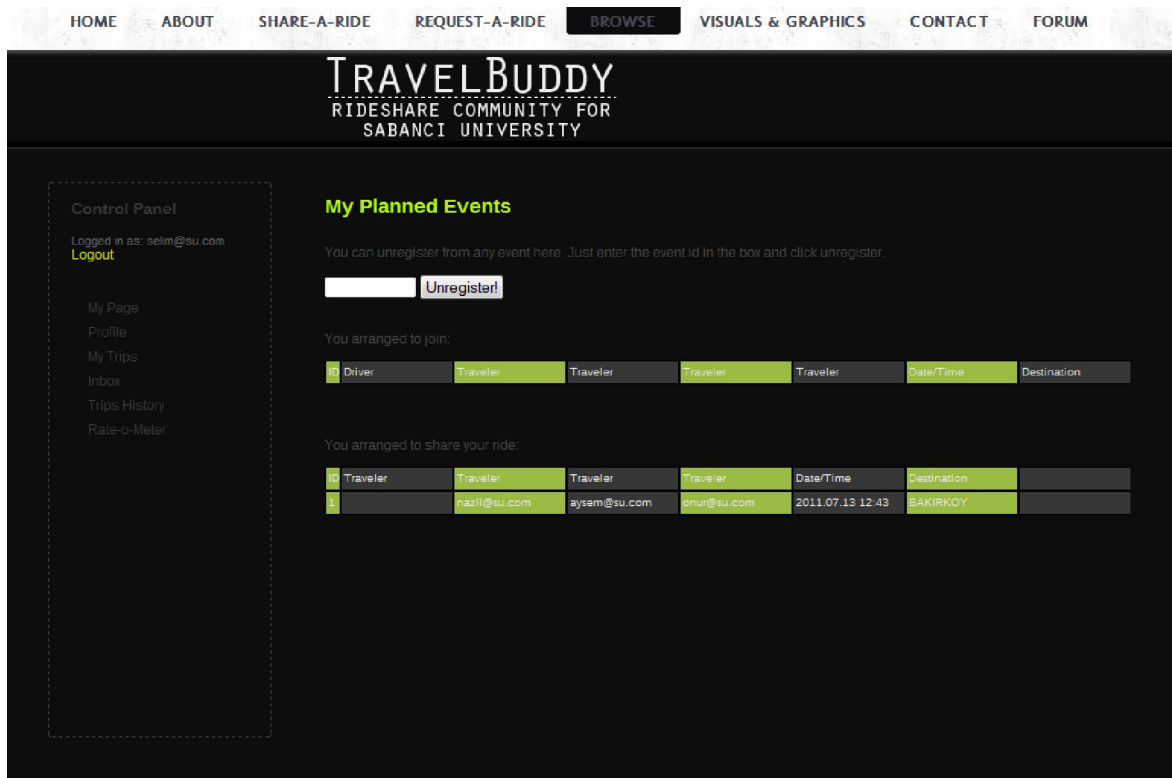


Figure 24. My Trips page where users can view their active events and unregister.

#### 4) Trips History: Completed Events & Rating Users

Trips History page displays all of the user's successfully completed events, and contains a link to a form that allows the user to rate any other user that he/she has previously traveled with. The structure and styling of the page is similar to MyTrips page. At the top of the content area is the link that opens rate-a-user iframe window. To rate another user, the user must fill the following fields; Trip ID (the ID of the carpooling event), e-mail of the user being rated, a quality value between 1-100, and optionally a comment. For v0.5b it is under consideration to remove comment entirely, and replace it with 2 mandatory fields called The Good and The Bad, which refers to what is good and what is bad about the user respectively. User will be presented with a list of possible attributes to choose from. When the user submits the form, rateuser-handler looks for the event in the TripLog with the entered Trip ID, if there is such an event, it first checks to see

if the event is completed (expired). If the event is expired, then it goes on to check if both the rater and the rated have actually been in the event. If this checks out as well, it will insert the entered information into the Ratings table.

Below the form again there are 2 static tables;

i- You have been a passenger in the following events: This table retrieves the data from TripLog for any expired carpooling events that the user is currently registered as a passenger. Table displays the trip's ID, driver, other passengers (if there is any), departure date/time, and destination.

ii- You have shared your ride on following events: This table retrieves the data from TripLog for any expired carpooling events that the user has created as a driver. Table displays the trip's ID, passengers that have registered (if there is any), departure date/time, and destination.

#### 5) Inbox: Recieve and Send PM

Inbox is designed to hold the messages the user has received. It consist of a dynamic table that accesses the Messages table and retrieves all the messages sent to the user. Table lists the messages in order of newest to oldest and displays who has sent it and when along with the message. Table is again sortable and searchable.

There is also a Send PM link on this page that allows the user to send a private message to any user, as long as he/she knows that user's ID (in this case e-mail). After the user clicks Send, message-handler first checks if the receiving user is actually registered in the database, to prevent useless database clutter. If the user is valid, then the message-handler inserts the entered information along with sender's e-mail and current date to Message table. It is under consideration for v0.5b, to move or place a copy of this Send PM link in Control Panel for quick access.



## 6) Rate-o-Meter: Global Ratings & Rating Users

Rate-o-Meter is another personal page accessed through the Control Panel. The page contains an information block displaying the user's current rating. As this is the ratings page, there is again a link that opens up the Rate-A-User iframe, which was discussed in the Trips History section. The page also contains 2 static tables displaying Top 10 users and Worst 10 users depending on their ratings.

## 7) Planned Additions to Control Panel

As discussed previously, Send PM is considered to be placed under Control Panel for quick access. A new personal quick access item that is planned to be added is Search User. This function will allow the user to search and view other user's limited profiles in either a new personal page or iframe window.

### 4.2.3.7 Forums

Forums have been decided to be integrated as they are a good primal way of quick communication. They are not a part of the actual project, and coding a forum wouldn't have been viable, thus one has been used. The choice was phpBB3 as it is free, powerful, easily integratable and has tons of free themes. A theme with similar colors to the web site has been chosen to protect integrity.

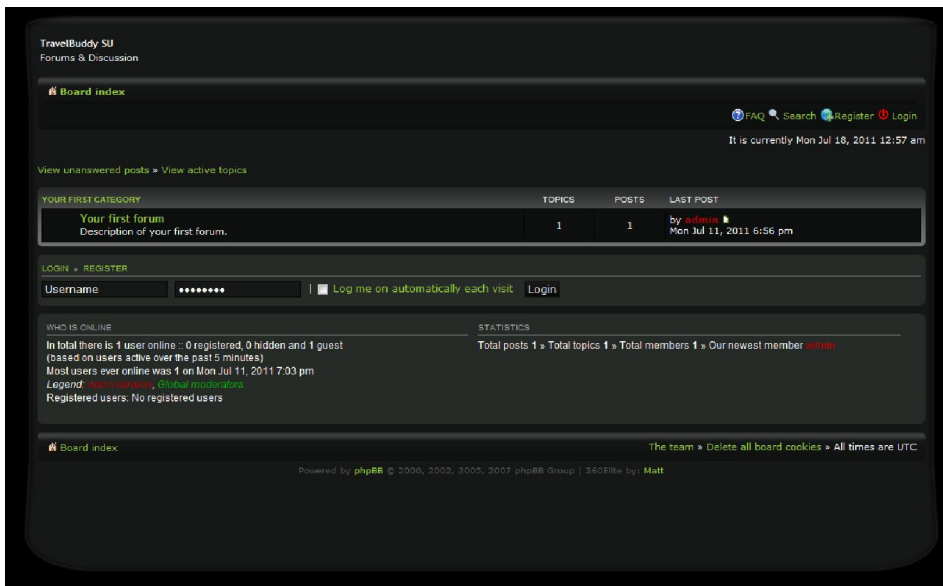


Figure 25. phpBB3 forum.

### 4.2.3.8 Background Processes

#### 1) DBSettings

DBSettings is a very simple php script that runs on most pages. It contains the required information such as database address, name, username, and password for the database access and opens a connection to MySQL if needed.

## 2) User Class & User-Handler

These 2 php scripts are needed to define what a user is to the system. All user specific pages require these scripts to run, to identify a user and check if the user is logged in or not.

## 3) Expire-Handler

Expire is a cyclic maintenance check script. It launches whenever one of the dynamic pages that access the database is opened. The script accesses the Event, Pool and TripLog tables, and looks only for active events. Then, compares the active events' departure date and times with the current date and time. If it finds any event that has passed its departure time, it sets the value of expired cell from 0 to 1 thus making the event completed.

## 4) Rating-Handler

This php script runs every time a user is rated. It accesses the UserRating table and retrieves the current rating of the user, then adds the given rating and gets the average. Finally, takes this new current rating and overwrites the one in the UserRating table, thus updating the rating of the user.

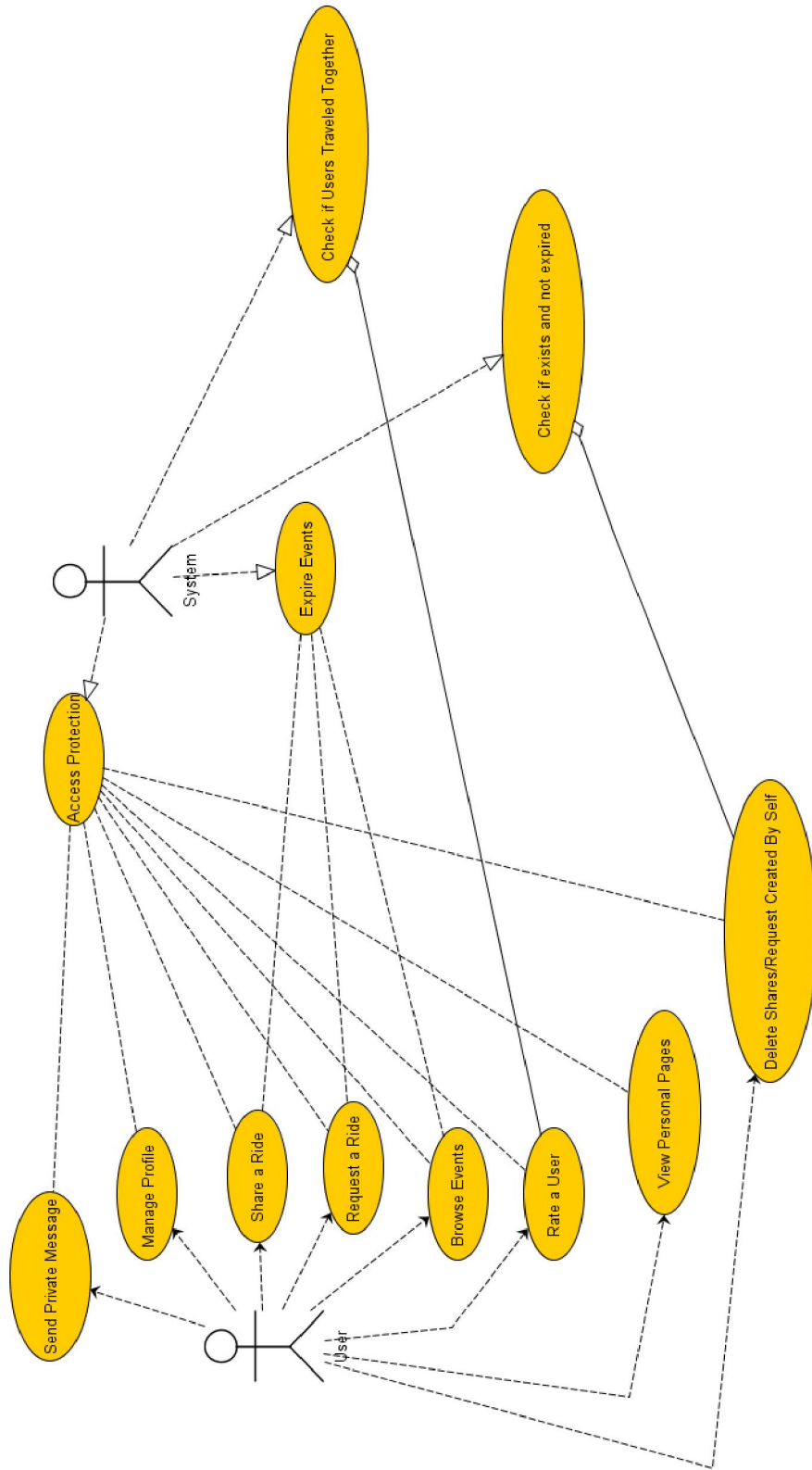


Figure 26. Background Processes

## **4.2.4 Accordance to Other Web Success Factors**

### **4.2.4.1 Legality & Credibility**

An initial short research over the web didn't turn out any laws against organized carpooling in Turkey. However, there is always the possibility of local transportation companies being irritated by such organizations and opening lawsuits similar to the one in US. It is a definite requirement to work with a legal expert before the system goes online, if it is decided to be marketed. There has to be a certain set of rules underlying what the system and the web site can and can't be hold accountable for. Only preliminary rule that is decided is the maximum number of passengers, which is limited to 4 because carrying more people can enter the area of public mass transportation.

It is thought that credibility isn't an issue as the system is a closed-community one and will ultimately belong to a certain trusted organization. For example, an implementation in a University should be done by the university's IT department, thus controlled and owned by the university itself, which would make it more trustable.

### **4.2.4.2 Semantic Approaches & SEO**

The code has been written according to W3C standards which is the globally accepted international standards for the World Wide Web. All the text is properly formatted HTML text, no image-texts has been used for crawler readability. All pages contain proper meta information needed for web indexing.

#### **4.2.4.3 Testing & Debugging**

Starting from version 0.2b, there has been rigorous testing and debugging. The web site has been used by 4 people who don't have any idea about its structure to see if they encounter problems and report any if found. There is still ongoing debugging to fix rarely occurring problems and for newly added functions.

#### **4.2.4.4 Error Handling**

Most errors are handled with proper system messages, however error handling has never been a top priority. In future, it is considered to have custom error pages for standard web errors such as 404.

#### **4.2.4.5 Maintainability**

As the web site has been cleanly coded in PHP and HTML, it is very easy to maintain and update the web site as needed. Most of the code is commented so anyone with PHP and HTML knowledge can read through the code and make any necessary changes, even add new features easily.

#### **4.2.4.6 Marketing Strategy**

Targeted audiences are closed-communities. There hasn't been a planned marketing strategy, a finance expert should be consulted if the system is decided to be marketed. The only viable idea to generate extra revenue is adopting location targeted advertisement depending on the targeted audience.

## **CHAPTER 5:**

### **Conclusions & Future Work**

#### **5.1 Thoughts & Conclusions**

Web sites/applications are increasingly becoming an integral part of our lives and the ultimate goal of web development is moving towards better user experience as the user experience is the main deciding factor between the success and the failure on web. The developer must first establish the need, then gather information, evaluate the information, and finally implement once everything is in place. This thesis was structured in that sense, as first the need for a carpooling system was established as the need. Then all the information on developing, succeeding on web and carpooling systems was gathered. Finally, after a careful evaluation of the information, the system prototype was created.

There are many different factors for web success, this thesis discussed usability, design and responsiveness through infrastructure as the primarily important factors as they are the commonly accepted major factors. Chapter 2 serves as a roadmap depending on these factors and is based on articles and generally accepted conventions. It is known that some articles and conventions are composed from personal ideas, thus all information gathered has been carefully evaluated and utmost effort has been made to validate them. The main goal for this research was to create a strategy guide for success on web that would help the systemized implementation design process of TravelBuddy by utilizing the three factors, so that they would benefit each other. The research is not intended to be a

definitive set of instructions and not necessarily applicable in all situations, however it is believed that it can be beneficial during web design and development.

Chapter 3 of this thesis provides a basis of for the development of a carpooling system by combining existing work and research in the field with identified challenges encountered during the implementation of modern carpooling systems. The research provided the functional and non-functional key issues required to create a proper carpooling system. Evaluating the issues identified that most problems could be overcome by a closed-community carpooling system that has an improvable and extendable infrastructure.

TravelBuddy has been an interesting and challenging project to work on. It was aimed to provide a better technological solution to both carpooling that is nearly non-existent in Turkey and scarcity of transportation options in distant places like Sabanci University. The detailed and scalable prototype creates a basis step for an even more complex system that could continuously evolve and improve. I am confident that through the extensive research and the prototype, this thesis was able to constitute a building block for a successful closed-community carpooling system.

## **5.2 Future Work**

The primary objective of this thesis was to create a systematic and successful development roadmap and prototype implementation of a closed-community carpooling system. Future work focuses on further improving the system and can be collected under 2 main subjects; live-testing and feature development.

In the first step, the system should be tested by real users as this would provide invaluable feedback on several different fields that would otherwise couldn't be retrieved. For example, a stress test can be performed if a user number of 500 and above can be



attained which would provide how fast the system can respond to simultaneous requests. Many problems have been discovered and dealt with during the testing phase with few people; however it is obvious that not every case can be realized right away. A live-testing with more people can help discover unforeseen inconsistencies and situations.

The second step includes further developing the system by both adding new features and improving the already available ones. Due to the nature and magnitude of the project, except a few unique features the prototype included only the must have features. Chapter 4 discussed what has been included and some of which features are planned to be added. Live-testing can also be helpful in this area as people may suggest some features they feel that are required.

After all, internet and technology are ever-evolving, thus it is envisioned that for the system to be successful, it has to be re-worked and improved continuously as well.

## RESOURCES

- Bedi, P. & Banati, H. (2006). Assessing user trust to improve Web usability. *Journal of Computer Science*, 2, 283-287.
- Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., & Secret, A. (1994). The World-Wide Web. *Communications of the ACM*, 37(8), 76-82.
- Berners-Lee, T. (1995). Style guide for online hypertext. Retrieved from <http://www.w3.org/Provider/Style> (accessed on May 1, 2011)
- Chain Store Age. (1997). Web-based retailers tell disparate tales at NRF. *Chain Store Age*, February, 45-52.
- Chiew, T. K. (2009). Web page performance analysis. Ph.D. Thesis, University of Glasgow, Scotland.
- Constantinides, E. (2004). Influencing the online consumer's behavior: The Web experience. *Internet Research*, 14, 111-126.
- Dailey, D.J. & Meyers, D. (1999). Proceedings from Intelligent Transportation Systems '99: *A statistical model for dynamic ride matching on the World Wide Web*.
- DeBra, P.M. (1996). *Hypermedia structures and systems* [HTML document]. Retrieved from Eindhoven University of Technology web site: <http://wwwis.win.tue.nl/~debra/ah98/paper.html> (accessed on April 14, 2011)
- Duval, B. K. & Main, L. (1995). Building home pages. *Library Software Review*, 14(4), 218-227.
- Fitzpatrick, R. (1999). Strategies for evaluating software usability. Online article retrieved from Dublin Institute of Technology web site: <http://arrow.dit.ie/scschcomart/1/> (accessed on April 4, 2011)
- Ghelawat, S., Radke, K., & Brerton, M. (2010). Proceedings from the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction: *Interaction, privacy and profiling considerations in local mobile social software: a prototype Agile Ride*. Brisbane, Queensland.
- Goldberg, M.W. (1997). Proceedings from the ACM's 28th SIGCSE Technical Symposium on Computer Science Education: *CALOS: First results from an experiment in computer-aided learning*. San Jose, California, USA.

- Gruebele, P.A. (2008). Interactive system for real time dynamic multi-hop carpooling. Retrieved from [www.dynamicridesharing.org/resources/Multi\\_hop\\_social\\_carpool\\_routing\\_System.pdf](http://www.dynamicridesharing.org/resources/Multi_hop_social_carpool_routing_System.pdf) (accessed on April 14, 2011)
- Hansen, W.J., & Haas, C. (1988). Reading and writing with computers: A framework for explaining differences in performance. *Communications of the ACM*, 31, 1080-1089.
- Hartwig, S., & Buchmann, M. (2007). Empty seats traveling: Next-generation ridesharing and its potential to mitigate traffic and emission problems in the 21st century. Technical report for Nokia, retrieved from <http://research.nokia.com/files/tr/NRC-TR-2007-003.pdf> (accessed on May 11, 2011)
- Hoxmeier, J. A., & DiCesare, C. (2000). Proceedings from the 4th COLLECTeR Conference on Electronic Commerce: *System response time and user satisfaction: An experimental study of browser-based applications*. Breckenridge, Colorado, USA.
- Kirshner, D. (2006). Pilot tests of dynamic ridesharing. Technical report, retrieved from [http://www.ridenow.org/ridenow\\_summary.html](http://www.ridenow.org/ridenow_summary.html) (accessed on April 4, 2011)
- Lammi, J. (2010). Data table. Retrieved from <http://patternry.com/p=data-table/> (accessed on May 1, 2011)
- Lang, M. & Barry C. (2001). A survey of multimedia and web development techniques and methodology usage. *IEEE Multimedia*, 8, 52-60.
- Liu, B., Abdulla, G., Johnson, T., & Fox, E. A. (2008). Web response time and proxy caching. Retrieved from <http://eprints.cs.vt.edu/archive/00000486/> (accessed on April 4, 2011).
- Liu, C., & Arnett, K. P. (1999). Exploring the factors associated with Web site success in the context of electronic commerce. *Information & Management*, 38(1), 23-33.
- Loiacono, E.T., Watson, R.T., & Goodhue, D.L. (2000). WebQual: A website quality instrument. Working Paper, University of Georgia. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.4410&rep=rep1&type=pdf> (accessed on May 1, 2011)
- Machlis, S. (1998). Site redesigns keep it simple. *Computerworld*, 32(43), 43-44.
- Marchionini, G. (1995). *Information seeking in electronic environments*. Cambridge: Cambridge University Press.

- McNamara, P. (2003). *Usability and eLearning, final coursework essay: certificate in online education and training* [PDF document]. Retrieved from Institute of Education, University of London: <http://nte.unifr.ch/IMG/pdf/PMcNamaraUsabilityElearning.pdf> (accessed on May 12, 2011)
- Murphy, P. (2007). The Smart Jitney: Rapid, realistic transport. *New Solutions*, 12, 1-11.
- Netcraft. (2011). June 2011 Web server survey. Updated: June 7, 2011. Retrieved from <http://news.netcraft.com/archives/2011/06/07/june-2011-web-server-survey.html> (accessed on June 14, 2011).
- Nielsen, J. (1993). Response Times: The 3 Important Limits. Retrieved from <http://www.useit.com/papers/responsetime.html> (accessed on May 12, 2011).
- Nielsen, J. (1996). Top 10 mistakes in Web design. Retrieved from <http://www.useit.com/alertbox/9605a.html> (accessed on May 12, 2011).
- Nielsen, J. (1997a). Search and you may find. Retrieved from <http://www.useit.com/alertbox/9707b.html> (accessed on June 3, 2011).
- Nielsen, J. (1997b). Be Succinct! (Writing for the Web). Retrieved from <http://www.useit.com/alertbox/9703b.html> (accessed on June 3, 2011).
- Nielsen, J. (2000a). *Designing Web usability*. Indianapolis, IN: New Riders Publishing.
- Nielsen, J. (2000b). Drop-down menus: Use sparingly. Retrieved from <http://www.useit.com/alertbox/20001112.html> (accessed on June 3, 2011).
- Nielsen, J. (2005). 10 Heuristics for user interface design. Retrieved from [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html) (accessed on May 12, 2011).
- Nielsen, J. (2010). Website response times. Retrieved from <http://www.useit.com/alertbox/response-times.html> (accessed on May 12, 2011)
- Palmer, J. W. (2002). Web site usability, design, and performance metrics. *Information Systems Research*, 13(2), 151-167.
- Pearrow, M. (2000). *Web site usability*. Rockland, MA: Charles River Media.
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: Beyond human-computer interaction*. New York, NY: John Wiley & Sons.
- Price, M. (1997). What makes users revisit a Website? *Marketing News*. March 17, 12.

Resnick, P. (2006). From the Working Notes of the Symposium on Crossing Disciplinary Boundaries in the Urban and Regional Context '06: *SocioTechnical support for ride sharing*.

Schneidermann, B. (1998). *Designing the User Interface: Strategies for Strong Human-Computer Interaction*. Reading, MA: Addison-Wesley.

Shotsberger, P. G. (1996). Instructional uses of the World Wide Web: Exemplars and precautions. *Educational Technology*, 36(2), 47-50.

Smith, P.A., Newman, I.A., & Parks, L.M. (1997). Virtual hierarchies and virtual networks: Some lessons from hypermedia usability research applied to the World Wide Web. *Journal of Human-Computer Studies*, 47, 67-95.

Spool, J., Scanlon, T., Schroeder, W., Snyder, C., & DeAngelo, T. (1999). *Web site usability: A designer's guide*. San Francisco: Morgan Kaufmann Publishers, Inc.

Sottini, F., Abdel-Naby, S., & Giorgini, P. (2006). ANDIAMO: A Multiagent system to provide a mobile-based rideshare service. Retrieved from <http://eprints.biblio.unitn.it/archive/00001139/01/097.pdf> (accessed on April 4, 2011)

Tredinnick, L. (2004). *Why intranet fail (and how to fix them). A Practical guide for information professionals*. Oxford, UK: Chandos Publishing.

U.S. Census Bureau. (2009). 2009 American community survey, 1-Year estimates. Retrieved from U.S. Census: [http://factfinder.census.gov/servlet/STSelectServlet?\\_ts=327504750252](http://factfinder.census.gov/servlet/STSelectServlet?_ts=327504750252) (accessed on June 14, 2011)

Walbridge, E.W. (1995). Proceedings from the 6th International Vehicle Navigation and Information Systems Conference: *Real time ridesharing using wireless pocket phones to access the ride matching computer*. Seattle, Washington, USA

## SECONDARY RESOURCES

Barnard, L. & Wesson J. (2003). Usability issues for E-commerce in South Africa: an empirical investigation. South African Institute for Computer Scientists and Information Technologists, Republic of South Africa.

Brockmann, J. M. (1996). *Grid systems in graphic design*. Stuttgart: Verlag Niggli AG.

Bostock, S.J. (1997). *Designing Web-based instruction for active learning*. In B. H. Khan Ed. *Web-Based Instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Cotrell, J., & Eisenberg, M.B. (1997). Web design for information problem-solving: Maximizing value for users. *Computers in Libraries*, 17(5), 52-57.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM*, 33, 338-348.

Nielsen, J. (1994a). Proceedings from the ACM CHI'94 Conference: *Enhancing the explanatory power of usability heuristics*. Boston, MA.

Nielsen, J. (1994b). *Heuristic evaluation. Usability inspection methods*. New York, NY: John Wiley & Sons.

Young, F.L., & Watkins, S.E. (1997). Paper presented at the annual Midwest Section Conference of the American Society for Engineering Education: *Electronic communication for educational and student organizations using the World Wide Web*. Columbia, MO.