# Rideshare – EE464H Project Outline

**January 24, 2008**                                                                                              **Spring 2008**

Team Members:

| Name | UTEID | Phone # | Email | Course |
|------|-------|---------|-------|--------|
| Yoni Ben-Meshulam | JZB62 | (512) 633 0675 | YoniBen@mail.utexas.edu | 464H |
| Garrett Cooper | GC588 | (361) 563 9723 | GMCooper@mail.utexas.edu | 464H |
| Derreck Huhn | DSH58 | (210) 269 5678 | DerrickHuhn@mail.utexas.edu | 464R |
| Patrick Lowry | PML228 | (210) 861 3935 | Lowry@ece.utexas.edu | 464R |

Supervising faculty member: Dr. Constantine Caramanis

**Overview of Rideshare**

Rideshare aims to connect passengers and drivers seeking a "Rideshare" (or carpool) model of transportation, which generally optimizes the resources of travelers for mutual benefit. Rideshare will do this by minimizing travel time and travel distances, as well as taking into account the user preferences and constraints of both drivers and passengers. This Rideshare program positions itself as a service that takes the coordinating tasks and risks of seeking rides on bulletin boards (or similar models) by performing matching and route planning functions.

**Description and Design Content**

The objective of this semester-long project is to create the full rideshare web application, which can be used across the Internet to connect users interested in sharing rides. The application's service will be based on a matching algorithm that seeks a balance between minimizing route lengths and times, and maximizing the number of riders matched to drivers. This will be the most complex aspect of the project, since the underlying computational problem is NP-Complete. We will need to employ an approximation algorithm to find optimal results within the space of possible matchings. The application will be interfaced as a map-based web application, or "map mashup", through a designed website.

For the user interface, we will employ PHP for user management and a content framework, and MySQL as a database solution. We will integrate Google Maps to provide users with visual representations of results and routes. Users will enter their ride offers/requests using the map, and will review their results on a map. By clicking the results, they will receive additional information, also highlighted on the map.

A good testing platform will be necessary in order to evaluate each module of the application. To accomplish this goal we must design the software with testing in mind, i.e. design-for-test. Each testing requirement must be quantified in terms of our final software architecture.

Our final product will require a server that can support our projected bandwidth and storage needs. The server must have a web-application server installed on it; we plan to use Apache. Once we have the program running, and we are able to connect riders with drivers in an efficient and user-friendly manner, we will focus on ways to improve the application interface, quality of results, aesthetics, and distribution.

Because this project will carry over from EE364D, we have done much of the high-level design work already. However, we will need to figure out exactly which algorithms we need to implement for each of our features, as well as some technology testing of third-party programs (such as the Google API), which we have only recently begun working with.

**Deliverables**
Our foremost concern is to provide the functional rideshare web application, which users can utilize to find and offer rideshares. We will offer documentation of the high-level software architecture, as well as the low-level module implementation, and testing interfaces. The actual code will document the programming structures and functions used for implementing low-level modules. At the end of the semester we will also provide a comprehensive demonstration of this application.

**Testing**
The implementation of each software module will include private functions specifically designed for testing. These will test the quality of each module based on the following criteria.

1. Functionality, i.e., the module does what it is supposed to. This will be most challenging for the matching algorithm, so we will place a significant focus on constructing edge cases. One case which we wish to support is picking up a passenger at points A and B, which lie close to the line the driver is taking from C to D, but A != C and B != D.
2. Performance, in terms of how quickly the function acts. Though the matching problem can be interpreted as N-P complete, we will narrow the problem's scope to execute in a workable amount of time. We have already found evidence of algorithms which run in polynomial time, as can be seen in our System Design document.
3. Stress. As we presume that our program will increase in functionality with the higher numbers of users, we need it to be able to support as many users as possible. We will simulate users in testability functions, and determine if the breaking point is high enough.
4. Security. Each of our software modules must be safe from exploitation and misuse. The best we can do here short of hiring a security expert is to try to hack and exploit it ourselves, ensuring that the private functions are not accessible by outsiders, and that each user's personal information is private.
5. Usability. Every function that relates to user interactivity must be intuitive and simple. The user must not be hindered by complexity or dead-ends, and we should be able to test this by watching people who are unfamiliar with the website use it.

**Schedule**
Within a month, we expect to have the basic website and database functionality complete. In the next month, we should complete the actual matching application, as well as specific data entry and internal/external user communications and any other website functionality we will need. Though we will be testing the application upon completion of every module, the remaining time will be used for extensive testing and perfection of the matching algorithm.

**Individual Contribution**
Though we each plan on contributing to all major sections of the project, each team member will have an area which they will 'own,' so to speak. All team members will be involved in applying and implementing the algorithms used for the optimization. We aim to have several options available, and will implement and compare some of them, so there's plenty of work for each student in this regard. Yoni and Patrick will be in charge of generating ride requests and offers, to simulate real users. Derrick

and Garrett will develop the map-based user interface and map visualizations. Patrick and Garrett will design and implement the overall user interface of the application. Garrett will be in charge of any graphic design needed for the project.  All members of the team will be involved in designing the test framework and responsible for generating comprehensive test cases for the modules they are working on as well as for the entire system as a whole.

In addition to the design contributions, team members will each have a management role in the project. Patrick will manage the web server and MySQL database. Derrick will manage the version control system and overall code policy. Yoni will manage the wiki. Garrett will manage the overall documentation of the project.


Signed,

Yoni Ben-Meshulam                 Derreck Huhn

X_____               X_____

Patrick Lowry                          Garrett Cooper                          Dr. Constantine Caramanis

X_____               X_____               X_____