

Bear Rides Documentation

Team TBD2

Abstract

Bear Rides is a software designed for Baylor members to share a ride. As a peer-to-peer ride sharing software, we encourage drivers who want to make friend (money) by sharing rides with other students. Bear Rides is a platform that links driver and passenger together.

Background story for introduction

Our ex-team leader – Leyao Yang wanted to make a software that will help all Chinese (international) students easy to find a cheaper ride, (Uber or Uberish Application are too expansive or sometime let us feel not safe enough) she came out the idea of building a software that can help students find a ride that is cheaper and safer, and that comes out our application – Bear Rides

Requirements

1. Login
2. Logout
3. Post Message
4. Add Trip
5. Delete Message
6. Manage Trip
7. Account setting change

Analysis

Based on Java

Swing graphical user interface design

Derby Database development

Controller handling frames

Design

To start our software design, it is better to understand its frame (structure) first, how this software work internally. We find out our application will have user communicate with another user, so we need a message object and a message board. For a single user he/she will need a place to store his/her information, so we implemented database. All the design should be flow smoothly.

Implementation

We build our base objects first, so we can have a frame of our software, at the same time we are building a graphical user interface with Swing. After its frame finished, we consider how to link its internal data altogether. We used Model Controller View (MVC) structure, with complicated database. This software add user input to its database and other user can view its data.

Evaluation/Testing

We thoroughly test the model's API, the collection of objects, the controller and the access of database, creating, inserting, updating and delete data from database.

Conclusion

Development a software requires a lot of preparation, and when developing software, it is better to fail early, because it is hard to fix a problem after you disperse it to everywhere, just like treat the disease, find the problem early will help a lot. Also testing is important to check if software work properly, more we test, less we fail. Also knowing design patterns will help you build software more effective, it is best tool for a beautiful software. Also, start planning early will make your life easier.