

EECS 22L: Taxi Cab Alpha Release

Prepared by: Nistha Tandiya, Shang Ma, and Prof. Rainer Dömer

February 23, 2015

The third deliverable in the taxi project is the alpha version of the Taxi Management program for the user, as well as the corresponding alpha version of the program source code and documentation. While certain shortcomings are expected for an alpha version (e.g. routing is not optimal, accounting is inaccurate or absent, it sometimes crashes, and/or several of the optional features are incomplete), the deliverables should give a good impression of the state of the implementation and provide an early "preview" of what the final release will be.

For this submission, the client and taxi modules have the greatest weightage. A significant implementation of these two modules is expected for alpha release.

User/Customer Package (Taxi_Alpha.tar.gz)

For full credit (100%), the binary package for the user should have the following file hierarchy:

- **README**
 - author, version, date, general instructions ...
- **COPYRIGHT**
 - author and copyright
- **INSTALL**
 - installation instructions
- **bin/**
 - binary directory
 - *bin/taxi*: the executable Taxi program
- **doc/**
 - directory for documentation
 - *doc/Taxi_UserManual.pdf*

Source Code Package (Chess_Alpha_src.tar.gz)

For full credit (100%), the source code package for further developers should have the following file hierarchy:

- **README**
 - authors, version, date, general instructions...
- **COPYRIGHT**
 - authors and copyright
- **INSTALL**

- installation instructions (how to compile, install, and test)
- **bin/**
 - binary directory
- **doc/**
 - directory for documentation, report, etc.
 - *doc/Taxi_SoftwareSpec.pdf*
 - *doc/Taxi_UserManual.pdf*
- **inc/**
 - include directory, with all header file(s) or symbolic link(s) to header file(s)
 - *inc/Taxi_Server.h*
 - *inc/others.h*
- **src/**
 - source directory, with all source code files
 - *src/Makefile* - Makefile for sources
 - *src/Taxi_Server.c*
 - *src/others.c*
- **test/**
 - test directory with test setup for each of the functionality and their separate inputs and outputs
 - *test/test_communication.c* - check the communication between client and server for different types of requests
 - *test/test_routing.c* - check for optimality of routed path given certain number of customer requests
 - *test/test_accounting.c* - check functionality of accounting module
 - or similar

Note that the source code package should include all files needed to build, install, and test the software. In particular, the **Makefile** should support the following:

```
% make
% make test-all
% make test-gui / make test-comm / make test-xxx
% make clean
```