Design of Internet Services

PROJECT REPORT

Tanvi Borkar | Romina Nayak | Mounika Nakkala | Mayur Kabra
Instructor: Professor Richard Martin
CS553 | December 16, 2015

Introduction

This project is an internet service for providing a carpooling platform to people associated with Rutgers University (Anyone having a Rutgers NetId)

MOTIVATION

- The New Brunswick campus of Rutgers University in itself is spread across a wide span of New Brunswick and Piscataway covering 4 campuses in itself.
- Although it is covered by a regular scheduled bus services, they come with their limitations. These limitations include area covered, limitation of number of buses, no regular services on weekends/holidays, etc.
- To avoid this and due to numerous other reasons, there are many cars on the roads used by a large amount of students. Over occupied parking spots, and traffic during peak lecture hours are evidence of this.
- Also this just covers a small part of a regular student's day to day travelling, apart from travelling inside campus, the students have to rely on expensive cab services to travel to nearby places.

PURPOSE

- Reduces travel expenses for its users
- Reduces driving stress and parking space crunches for its users
- Travel to regular places on the campus which are not covered by the bus services
- Promoting friendship building/ travel buddies, interaction between its users
- Carpooling cares for environment, less cars, less fuel usage, less pollution

TARGET AUDIENCE

Any person having a valid Rutgers NetId looking for rides to nearby places or looking to share rides when they have vacant seats to share.

LIMITATIONS & WORKAROUNDS

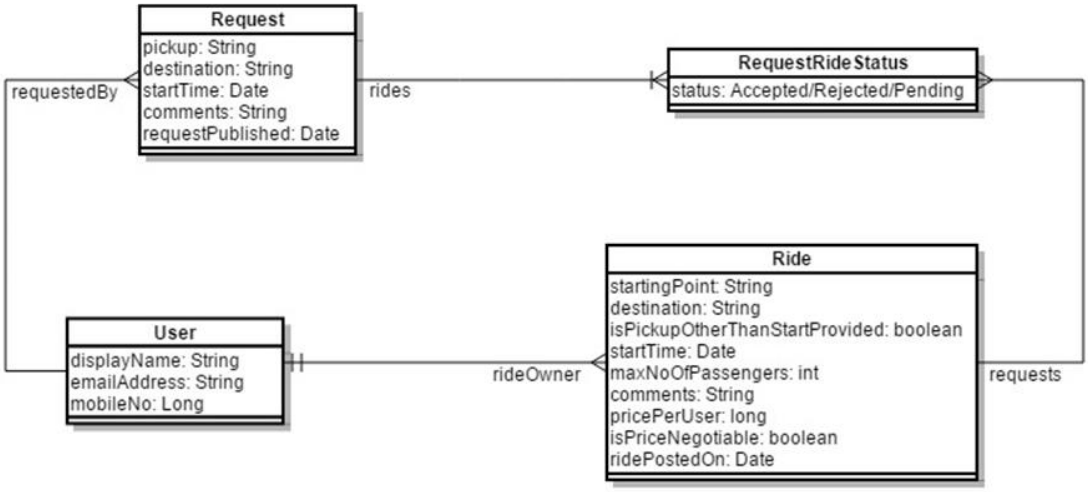
- Initially it was planned to use Rutgers CAS for authentication purpose of the user wanting to use the service.
- But since it could not be processed, Facebook login was used to demonstrate the same. So in this case, for demonstrative purpose, the user needs to have a Facebook login

Basic Business Requirements

- A user needs to be logged in to use any of the following services
- A user can post a request that he wishes to travel from a certain place to another
- A user can post a ride for which he/she can offer a defined number of pickups
- A user can view matching rides from the requests he/she made and request to pair them up, and vice versa
- Numerous validations follow including checking of number of passengers limit etc..

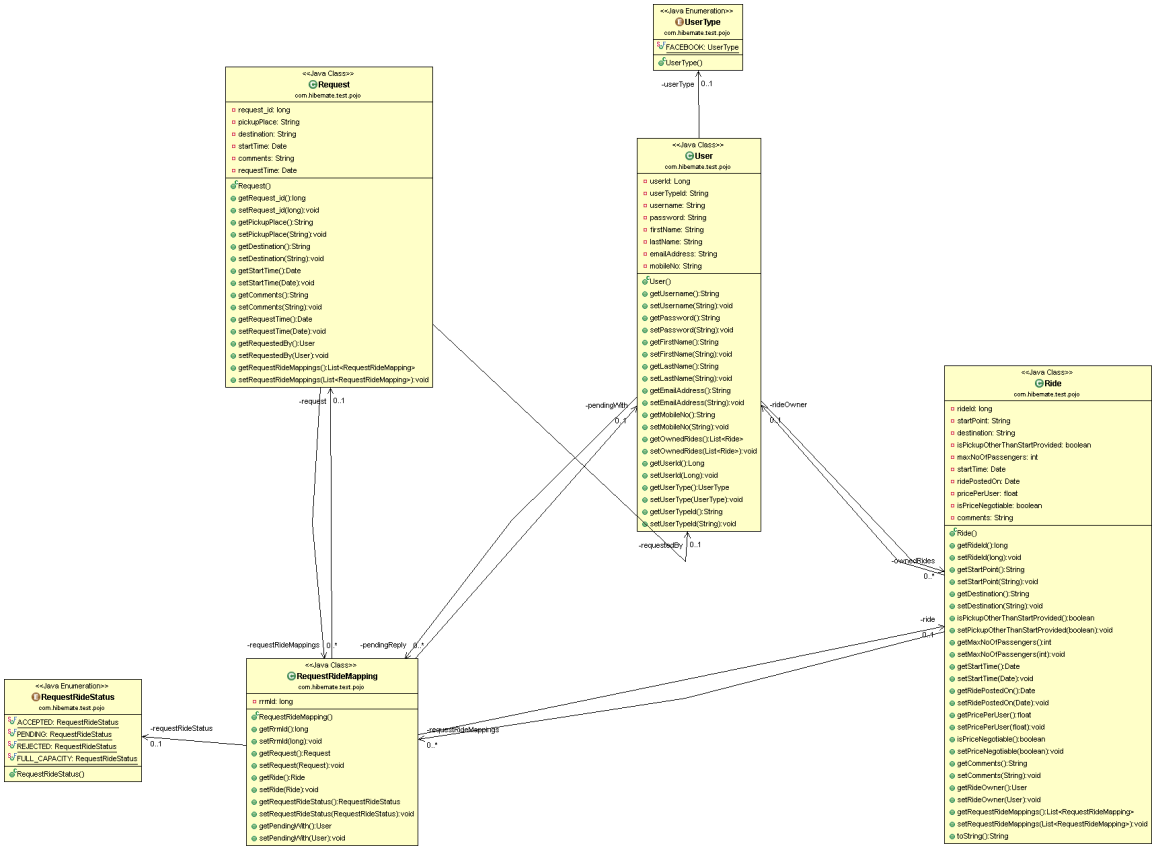
Modelling of Business Objects

BASIC



DETAILED

(The diagram has been committed in the repository as well)



Technologies Used

- Programming Language – Java (v 1.8)
- Frameworks – Hibernate ORM (v 4.3), Spring (v 4.2.2)
- Database – MySQL (v 5.7)
- Server – Apache Tomcat (v 7.0)
- Unit Testing – jUnit (v 4.11)
- Repository - github

WHY FRAMEWORKS?

- Hibernate
    - provides productivity (no native SQL queries to be directly written, helps concentrating on business logic)
    - provides maintainability (reduces lines of code, makes code more understandable)
    - provides portability (completely obscures the application from underlying database, can easily be ported to any other database)
    - is pretty much a standard accepted across many JavaEE using industries
    - is free
- Spring
    - helps in maintaining MVC (Model View Controller) architecture
    - provides IoC (Inversion of Control) capabilities, externalizes the creation and management of component dependencies by using DI (Dependency Injection)
    - is lightweight

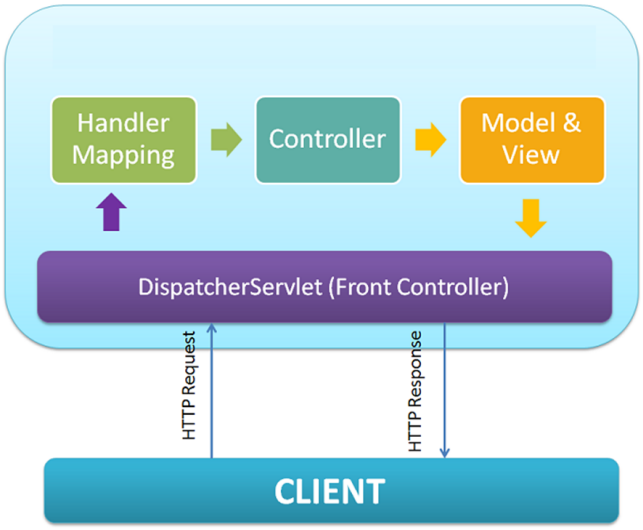
Technical Overview

VERTICAL MODULES

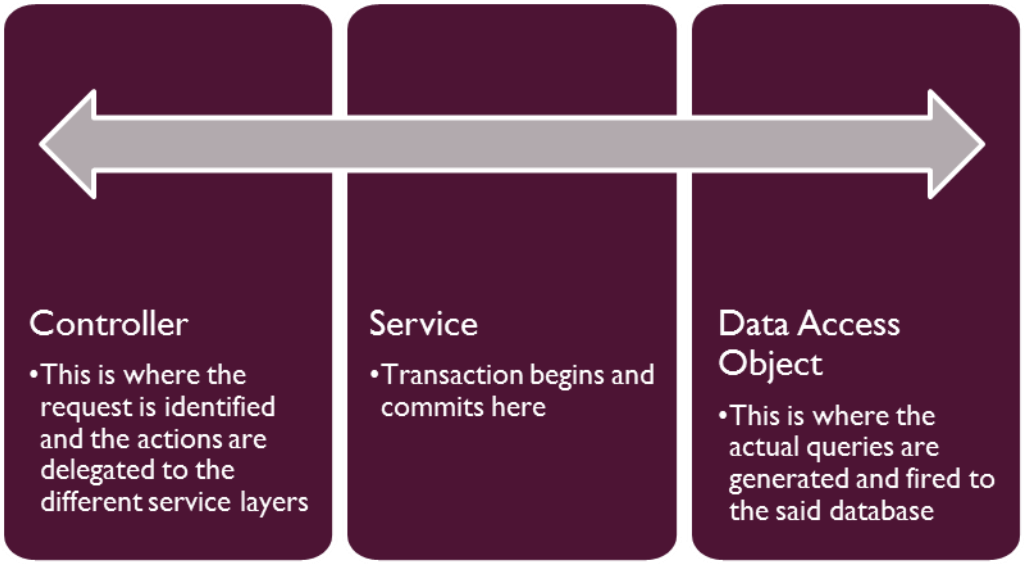
The project was separated into three verticals in terms of business objects

- Request
- Ride
- User

MVC ARCHITECTURE FLOW



The flow from controller goes through a set of different flows whenever a database related transaction is required, which can be summarised from the following diagram

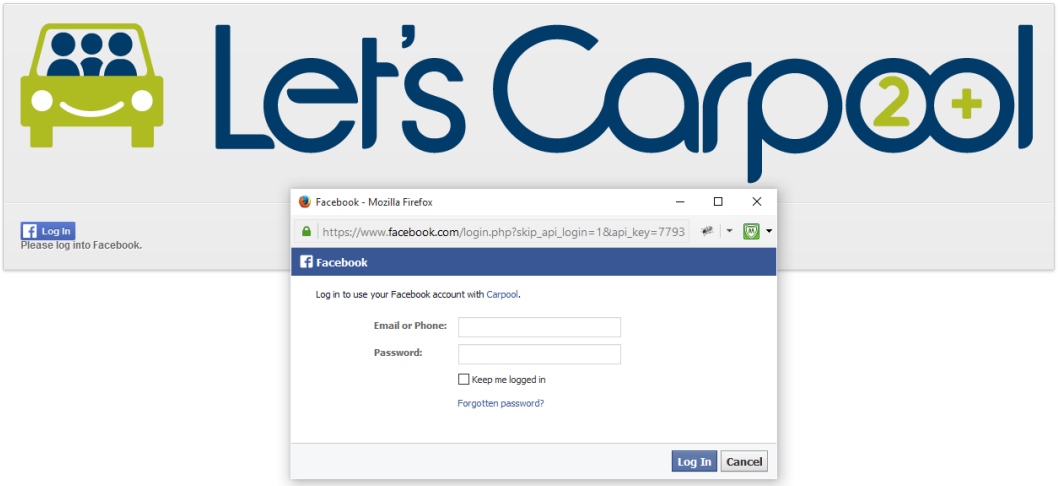


User Interaction Overview

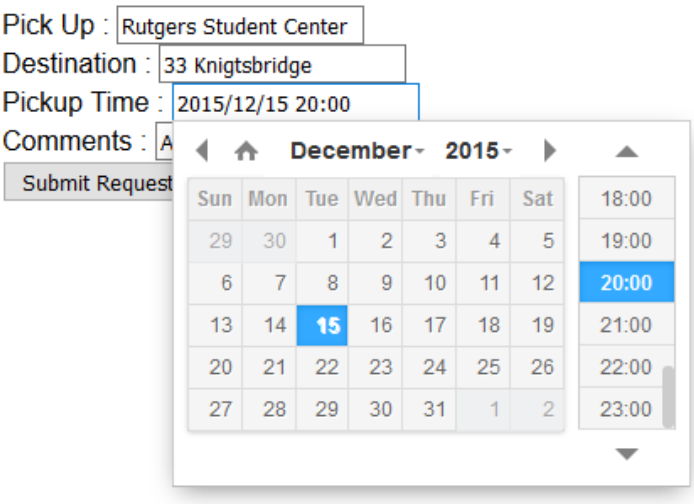
- This project is basically an internet service which can ideally be exposed and used from any available interface which can use this internet service. Eg: website, mobile application etc.
- For the purpose of demonstration of this service, we created a website using the created services

USER FLOW

Login using Facebook to avail all options



Post a new request



View requests in a listing on dashboard

Pickup	Destination	Pickup Time	Actions
Rutgers Student Center	33 Knightsbridge Road	2015-12-16 09:00:00.0	View All Rides

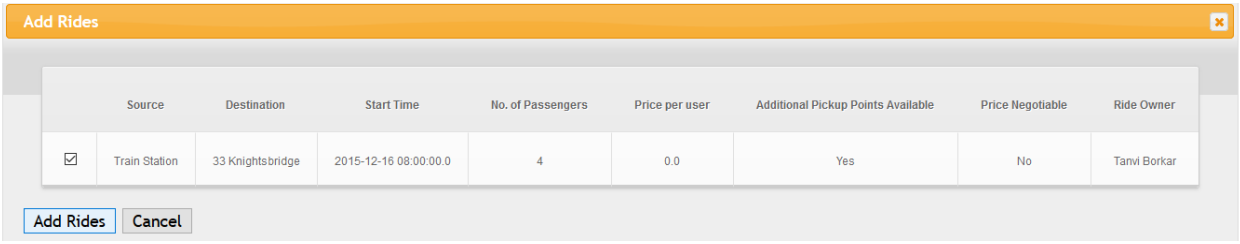
Post a new ride

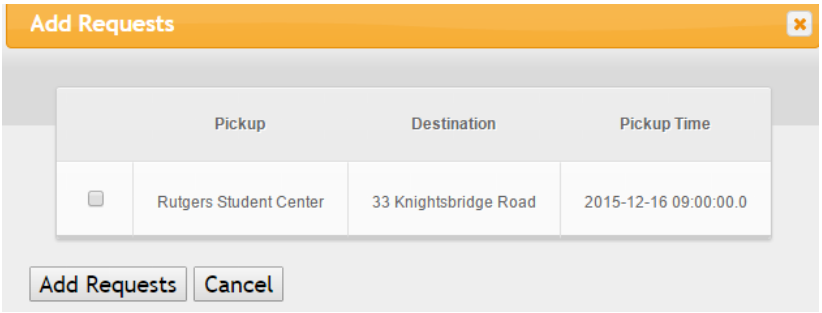
Source : New Brunswick Train Station
Destination : 33 Knightsbridge
Start Time : 2015/12/16 08:00
No. of Passengers : 4
Price per User : 0
Pickup other than start : ☑
Price Negotiable : ☐
Comments :
Post Ride

View rides in a listing

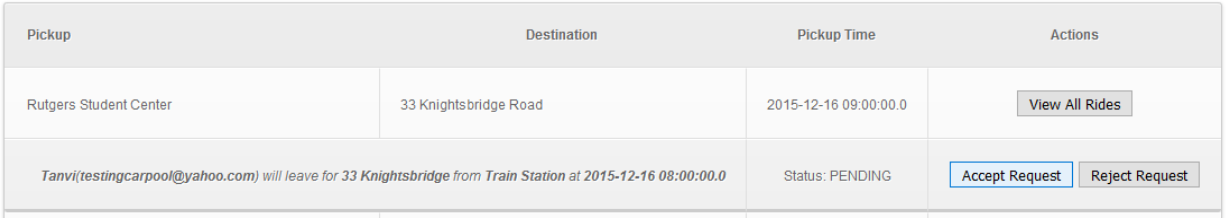
Source	Destination	Start Time	No. of Passengers	Price per user	Additional Pickup Points Available	Price Negotiable	Actions
aashq	ohooooo	2015-12-31 20:00:00.0	6	12.0	Yes	Yes	View All Requests
Mayur(mk_only4u91@rediffmail.com) needs to be picked up from C for F at 2015-12-31 20:00:00.0						Status: ACCEPTED	Accept Request / Reject Request
Mayur(mk_only4u91@rediffmail.com) needs to be picked up from P for L at 2015-12-31 20:00:00.0						Status: REJECTED	Accept Request / Reject Request
Train Station	33 Knightsbridge	2015-12-16 08:00:00.0	4	0.0	Yes	No	View All Requests

Requests and rides can view their matching counterparts





These internal pairing has to be accepted by the counterpart



Challenges Faced

- Synchronizing each member's work using the repository github. This came up with many conflicts which were getting difficult to solve. A lot of time had to put into the synchronization. We would suggest using subversion (SVN) instead of github, in case IDE used is eclipse.
- Spring and Hibernate integration did take a lot of time, but the outcomes are completely worth the challenges faced while integrating them.
- Finalizing the business logic took up some time due to the limitation of time and resources.

Workload Distribution

- Hibernate & Spring Integration – Mayur, Tanvi
- Database – Mounika, Romina
- Request Module – Mayur
- Ride Module – Tanvi
- User Module – Mounika
- Facebook Integration – Mayur
- UI & Presentation – Romina

Future Scope

- Rutgers CAS integration
- Exposure of services using web services which allows the service to be used by mobile apps and such other interfaces
- Option for recurring rides
- Integration with GPS and maps
- Vehicle tracking

Resources

The code for this can be checked out from: -

https://github.com/mayurkabra/HibernateMySQLTest.git