



# JagTrack

Elaboration Iteration 1

# Project Overview

- Vision
  - To create a mobile application that increase the utility of the JagTran system for users
  - To create a web-based application that provides usage data to JagTran administration
- High Level Features
  - Bus Tracking
    - Arrival Times
    - Locations
  - Administrative Information
    - Load balancing
    - Usage of all buses at a given time

# Elaboration Phase Plan

- Goals
  - Continue requirements gathering
  - Begin implementation of features that involve risky aspects of architecture
  - Develop a candidate architecture for the system
- Cost
  - No monetary cost yet

# Schedule

- Iteration 1

- Implement the standard path for "Get Passenger Count" and "Get Arrival Time" use cases
- Implement key architecture and components
  - Database
  - PHP application
- Define candidate architecture
- Continue requirements gathering

- Iteration 2

- Further refine architecture
- Begin carrying out all tests
- Realize another use case

# Team Configuration

## Team 1: Android Application

Christopher Johnson - Lead

Sumit Shrestha

Hao Wu

## Team 2: Desktop Application

Bradley Bittinger - Lead

Matthew Ngyuen

Weijian Jian

## Team 3: Deployment / Integration / Testing

Xingyu Wang - Lead

Shanna Keith

Leyue Wang

Rujie Yuan

## Team 4: Quality Assurance

Christopher Camp - Lead

Ray Bigelow

## Team 5: Hosting Setup / Persistence Layer Design / Database Design

Adam Moore

Hayden Chudy

## Team 6: Prototyping :

Robert Fornof

## Team 7 - Use Case Development / Requirements Analysis:

KD Wilson - Lead

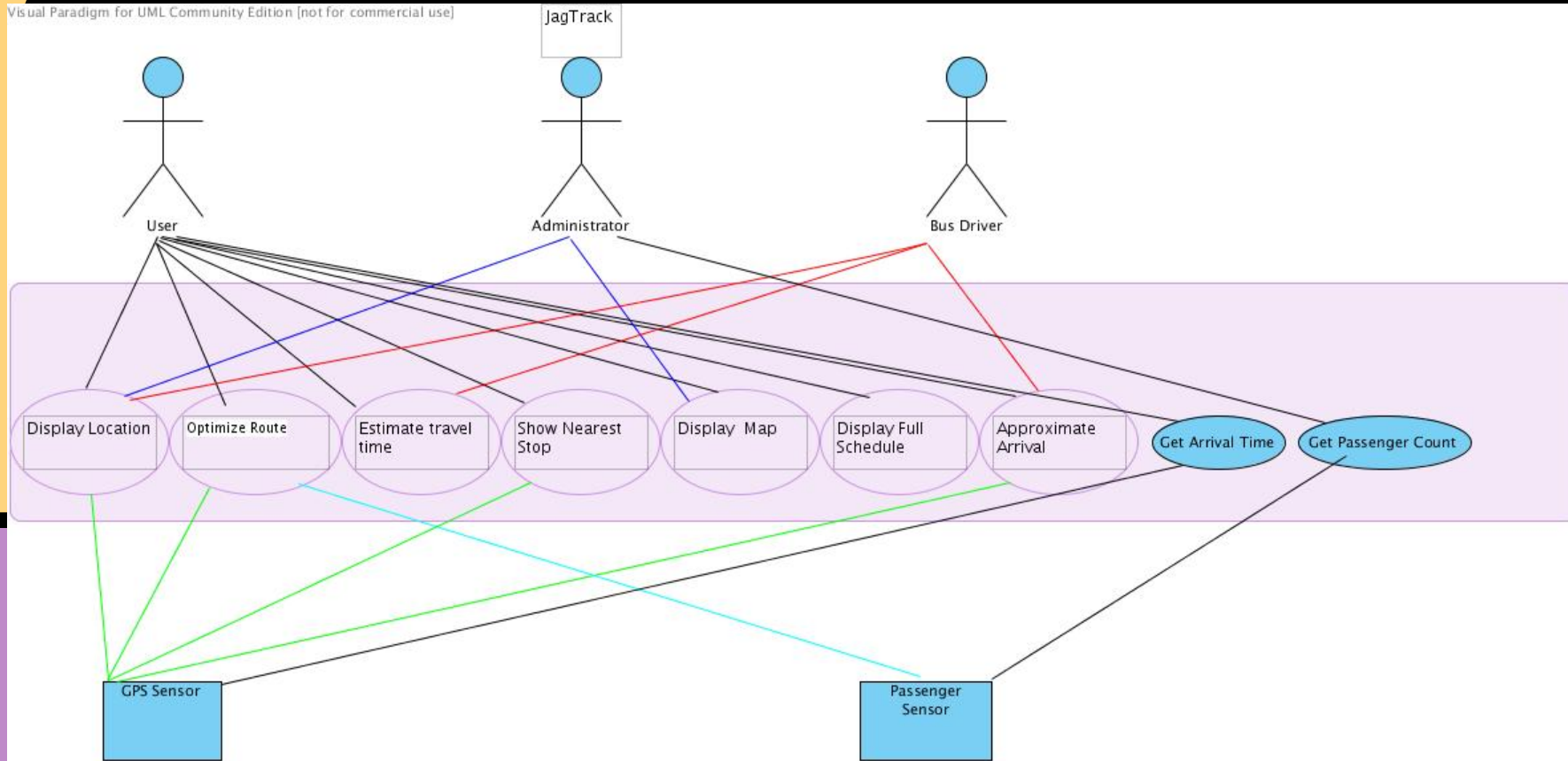
Chase Bryant

He Zhang

Jim Fletcher

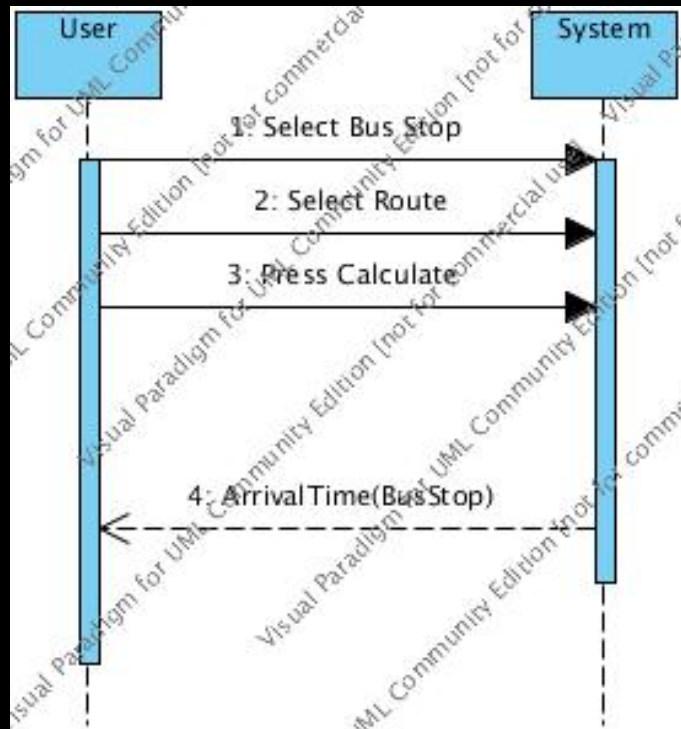
Caleb Hall

# Overall Use Case Diagram

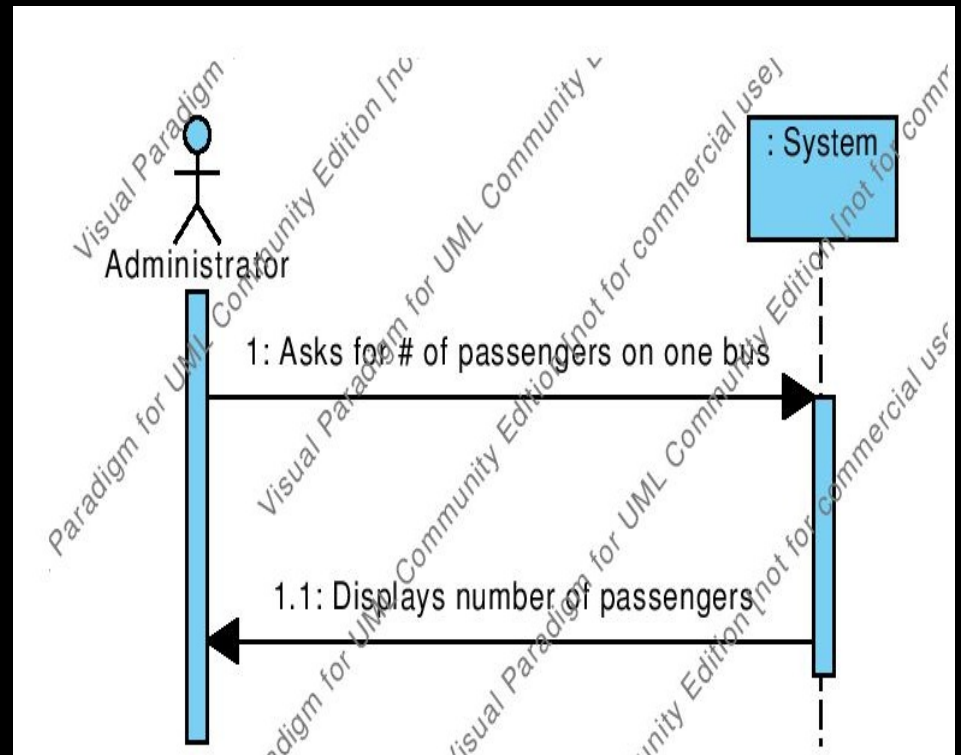


# Use Case Summary

## SSD: Get Arrival Time



## SSD: Get passenger count



# Use Case Summary

- Expected to further develop use cases
  - Communication issue



# Non-Functional Requirements

- Supportability
  - Android devices vary
  - Internet is not always available, sometimes smartphones malfunction
- Performance
  - Response Time
    - Android database calls must go through HTTP
  - Throughput
    - Scaling may be hard, prematurely fought with a cloud based host

# Risk Management

- Group Communication Lacking
  - Honestly, didn't do very well
  - Would do better by having more meetings and more clearly defined requirements
- Staff may be busy with other school work during design process
  - Didn't do well on this either
  - Could improve by being more explicit with staff about what is required

# Risk Management continued

- The server available for the project may not be capable of handling enough request for reading and writing data.
  - Didn't run into this as we only tested with a few devices
  - Would upgrade DotCloud plan if this became an issue

# Risk Management continued

- Android hardware may fail at critical times
  - This did happen.
  - Used AVD for development
  - Hayden's phone worked great!
  - Abundance of Android Phones and AVD's make this less risky



# **Architectural and Design Overview**

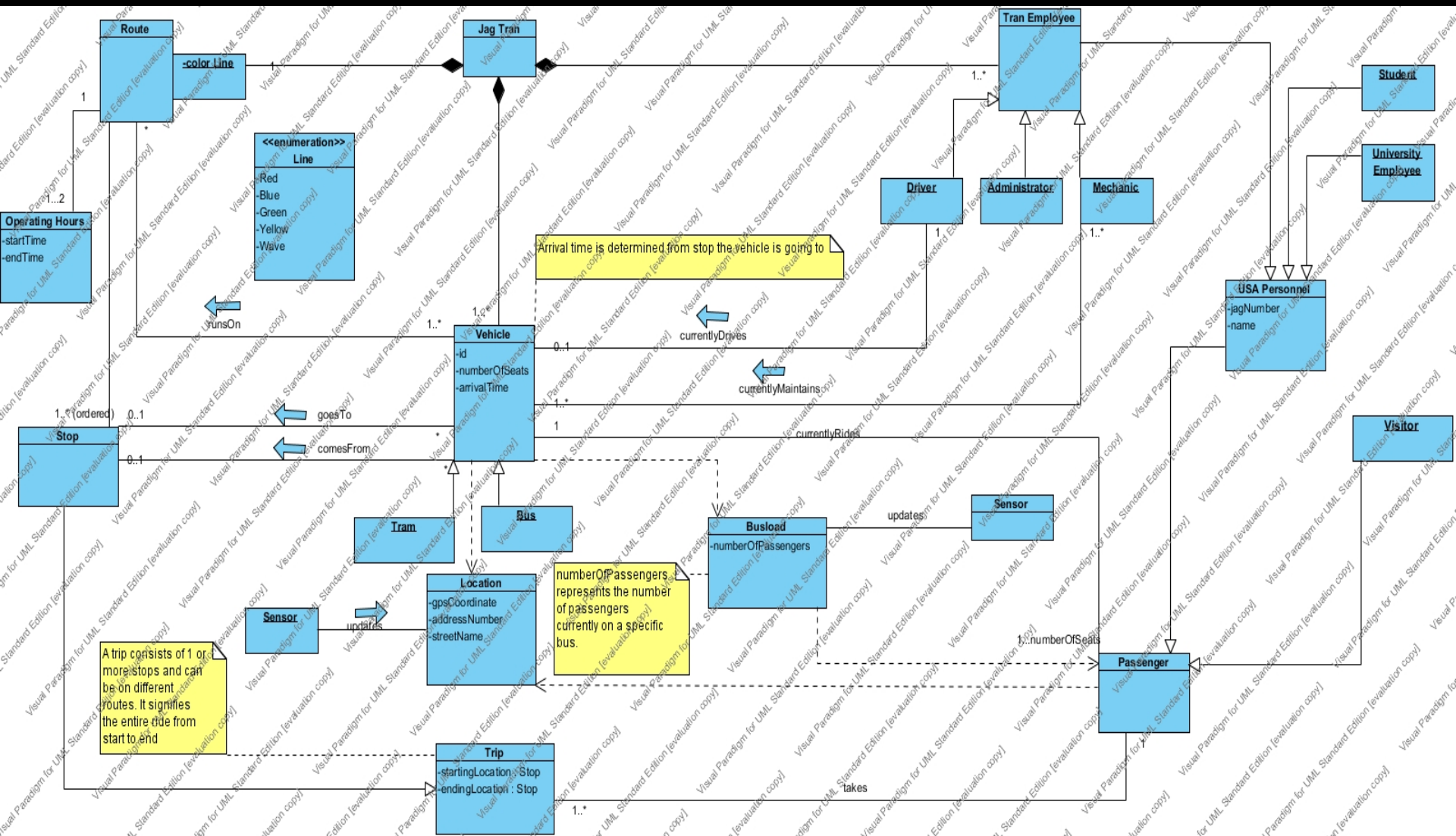
# Domain Model

The diagram is a UML Domain Model for a transit system. It includes the following elements:

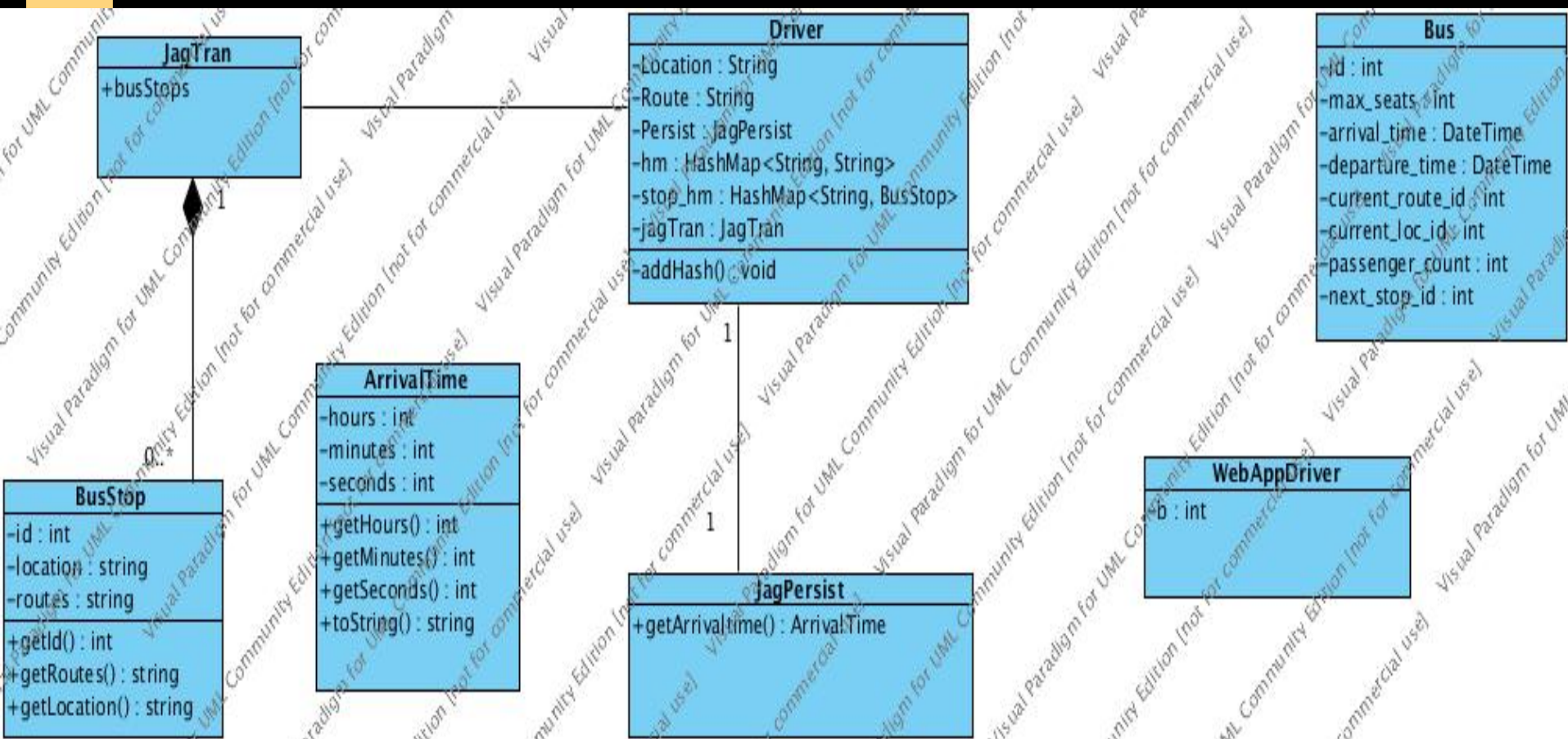
- Route**: Attributes include `start`, `end`, and `color`. It has a `RunsOn` association with `Stop`.
- Stop**: Attributes include `start` and `end`. It has a `comesFrom` association with `Vehicle`.
- Vehicle**: Base class for `Tram` and `Bus`. Attributes include `id`, `numberOfSeats`, and `arrivalTime`. It has a `currentlyDrives` association with `Driver` and a `currentlyMaintains` association with `Mechanic`.
- Tram**: Inherits from `Vehicle`.
- Bus**: Inherits from `Vehicle`. It has a `Busload` association with `Sensor`.
- Busload**: Attribute includes `numberOfPassengers`.
- Sensor**: Has an `updates` association with `Location`.
- Location**: Attributes include `gpsCoordinate`, `addressNumber`, and `streetName`.
- Trip**: Attributes include `startingLocation` and `endingLocation`. It has a `takes` association with `Passenger`.
- Passenger**: Has a `numberOfSeats` association with `Vehicle`.
- Driver**: Has a `currentlyDrives` association with `Vehicle`.
- Mechanic**: Has a `currentlyMaintains` association with `Vehicle`.
- Tran Employee**: Base class for `Administrator` and `Mechanic`.
- USA Personnel**: Base class for `JagNumber` and `name`.
- Student**: Inherits from `USA Personnel`.
- University Employee**: Inherits from `USA Personnel`.
- Visitor**: Inherits from `USA Personnel`.

Key relationships and notes:

- Route** has a `color` attribute and is associated with `Stop` via `RunsOn`.
- Vehicle** is a generalization of `Tram` and `Bus`. It has a `currentlyDrives` association with `Driver` and a `currentlyMaintains` association with `Mechanic`.
- Bus** has a `Busload` association with `Sensor`.
- Sensor** has an `updates` association with `Location`.
- Location** has attributes `gpsCoordinate`, `addressNumber`, and `streetName`.
- Trip** has attributes `startingLocation` and `endingLocation` and is associated with `Passenger` via `takes`.
- Passenger** has a `numberOfSeats` association with `Vehicle`.
- Tran Employee** is a generalization of `Administrator` and `Mechanic`.
- USA Personnel** is a generalization of `JagNumber` and `name`.
- Student**, **University Employee**, and **Visitor** are specializations of `USA Personnel`.

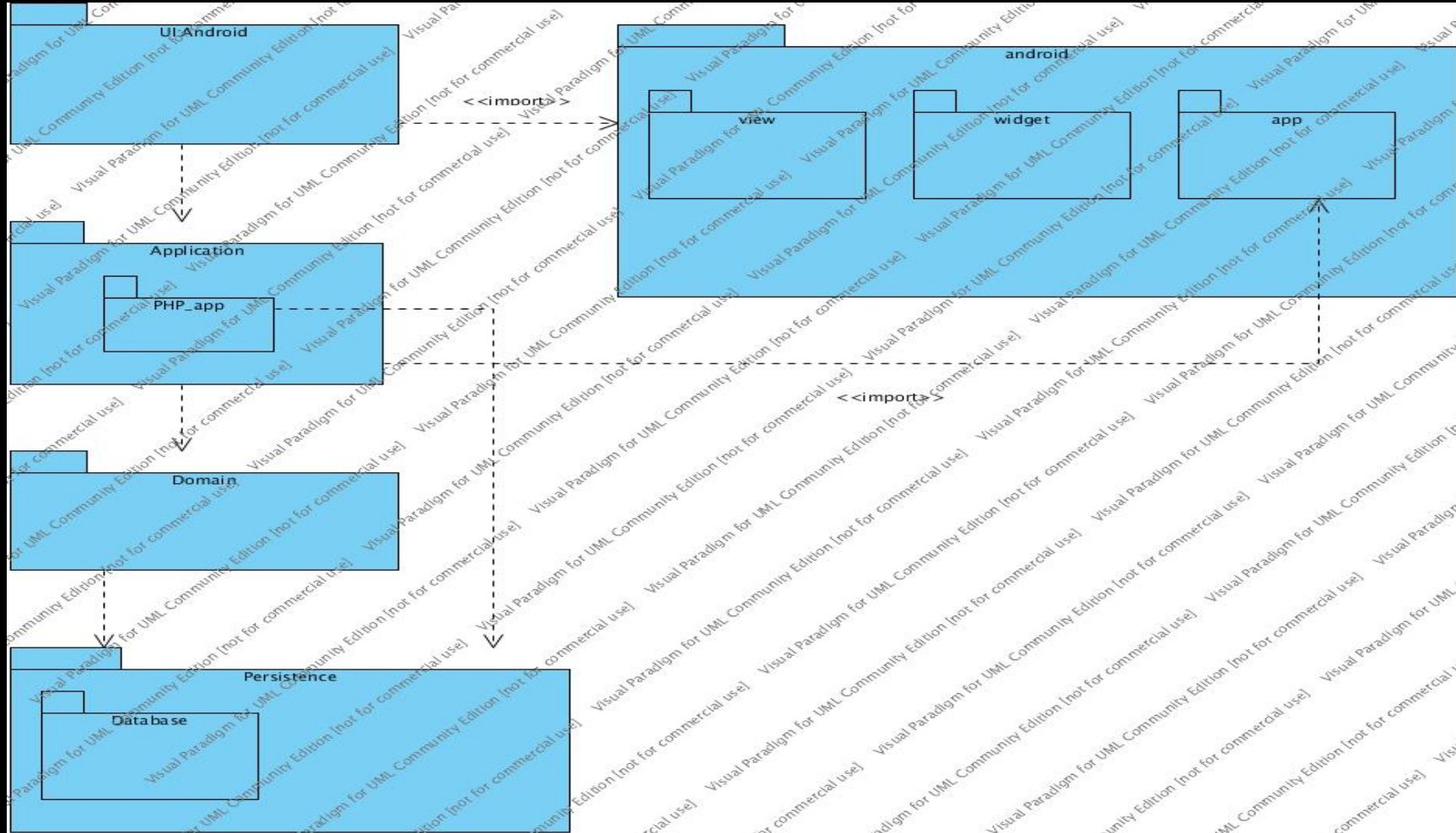


# Design Class Diagram



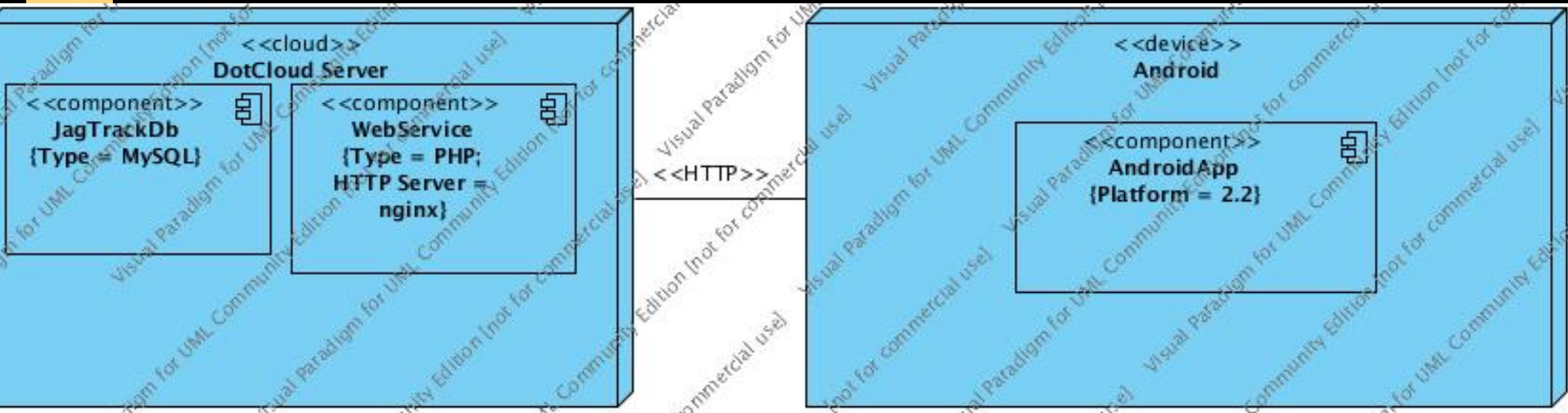


# Package Diagram





# Deployment Diagram



# Test Plan and Results

- Data and Database Integrity: Passed
- Function Testing: Passed so far
- UI Testing: Passed
- Performance Testing: Not yet tested
- Load Testing: Not yet tested
- Stress Testing: Not yet tested
- Volume Testing: Not yet tested
- Security and Access Control: Minimal testing on web app side
- Configuration Testing: Passed

# Demonstratons

- Prototype
- Production Code
  - Android app
  - Web interface

# Project Status

- Architecture issues were addressed
  - DotCloud was used and an error with it was fixed
  - The Java application turned into simple HTML, to prepare for a web interface
  - The database and early server side code were successfully setup
- Use Cases
  - Use cases were unified into one diagram
  - Use cases were partially realized

# Recommendations for Next Iteration

- Flesh out web interface
  - Pick a CMS or web framework to drive the web interface
    - This offers security, easier administration, and better looks
  - Allow administrators to see more info about buses
- Get a Google Map implementation started
  - Both on the Android and the web interface

# Why Should You Hire Us?

- As students, we are familiar with the JagTran and the campus
- We have an Android development team led by an experienced developer
- We have a web development team led by an experienced developer
- We're affordable