Université
Nice
Sophia Antipolis

BlablaMove Architecture - Mobile tracking app

Members :
Couvreur Alexis
Spinelli Aurélien
Swiderska Joanna
Wilhelm Andreina

Tutor :
M. Guilhem

September - February 2018/2019

Contents

1 System Description

BlablaMove is an application that it is meant to help students move their goods or furniture for a much lower price by using the free space in other people's cars who are going to the same destination or doing part of the path.

2 Mobile Tracking App

As part of the entire system this app will allow students follow their goods along the way from start to finish and track all the possibles changes in between.

2.1 Scope

The general architecture of the app is shown in the Use Case Diagram of Figure 1 where there will be two main users: Student and Driver.

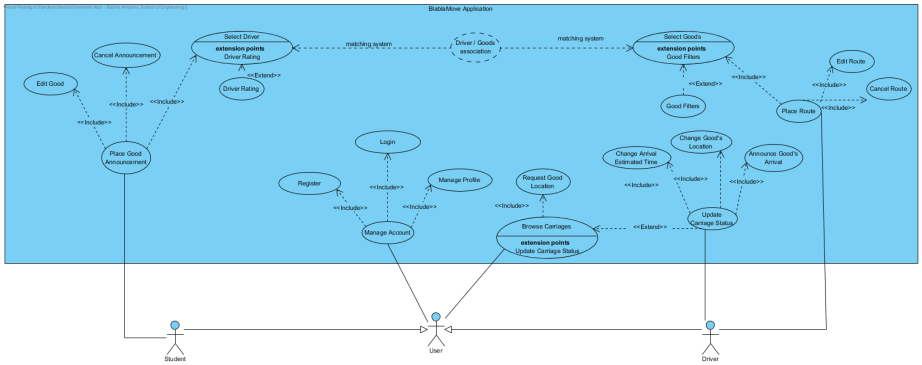

Figure 1: General architecture of the app

For the scope of this project the following functionalities of the app will be developed:

- **Account management:** Create, edit and delete an account. Log in and log out.
- **Announcement management:** Create, edit and delete the different types of announcement. For *students* it will be an announcement for moving goods and for *drivers* it will be a free space for transportation kind of announcement.
- **Tracking:**
  - Notification on received
  - Notification on start
  - Notification on checkpoints

  - Notification when there is a change of drivers (route combination)
  - Notification on incidents
  - Notification on arrival
  - Notification on delivered

On the other hand, the following functionalities will be mocked as they are not part of the scope, but are needed for the development of the app:

- **Matching system goods/routes:** Assign a driver to a good delivery.
- **Billing:** Distribute points according to service.
- **Volume assessment:** Estimate the volume of the goods to be moved.

2.2 Scenarios

Personas:

- *Lucas* is a student living in **Sophia** that needs to send his bike to his brother *Charles* in **Paris**
- *Charles* is *Lucas'* brother living in **Paris**
- *Austin* is a mechanic living in **Nice** who's traveling to **Lyon** soon
- *Mila* is a dancer living in **Lyon** who's traveling to **Paris** soon
- *Hope* is a student living in **Nice** who's traveling to **Paris** soon

2.2.1 Tracking with direct route

This scenario describes the tracking of the goods in a direct route (only one driver involved) with no incidents recorded.

1. *Lucas* create an announcement stating:
   - Bike 2 wheels, 8kgs
   - Departure: Sophia, 10km radius, before October the 12th
   - Arrival: Paris 10th arrondissement, before December the 24th
   - Picked up by: *Charles*
2. System finds a matching route with *Hope*
3. *Lucas* and *Hope* are notified and they agree. All the announcements change their status.
4. *Lucas* meets *Hope* and gives her the bike. She notifies on the app that she's received it.

5. *Hope* leaves the following day. She notifies when her trip begins.

6. *Hope* meets *Charles* and proceeds to give him the bike

    - *Hope* notifies that she has completed the delivery.
    - *Charles* notifies he has received the bike.
    - The system transfer points to *Hope's* account

2.2.2 Tracking with route combination

This scenario describes the tracking of the goods in a combination of routes (more than one driver involved) with no incidents recorded.

1. *Lucas* create an announcement stating:

    - Bike 2 wheels, 8kgs
    - Departure: Sophia, 10km radius, before October the 12th
    - Arrival: Paris 10th arrondissement, before December the 24th
    - Picked up by: *Charles*

2. System finds a matching route combination with *Austin* and *Mila*.

3. *Lucas*, *Austin* and *Mila* are notified and they agree. All the announcements change their status.

4. *Lucas* meets *Austin* and gives him the bike. He notifies on the app that he's received it.

5. *Austin* leaves the following day. He notifies when his trip begins.

6. *Austin* meets *Mila* and proceeds to give her the bike.

    - *Austin* notifies that he has completed his part of the delivery.
    - *Mila* notifies that she has received the bike.
    - The system transfer points to *Austin's* account

7. *Mila* leaves the following day. She notifies when her trip begins.

8. *Mila* drives through **Dijon** and notifies the checkpoint.

9. *Mila* arrives to **Paris**. She meets *Charles* and proceeds to give him the bike

    - *Mila* notifies that she has completed the delivery.
    - *Charles* notifies that he has received the bike.
    - The system transfer points to *Mila's* account

2.2.3 Component Diagram

For a better understanding of the system the Component Diagram on Figure 2 has been created. The components were separated server side and client side where most of our system will be on the server side.

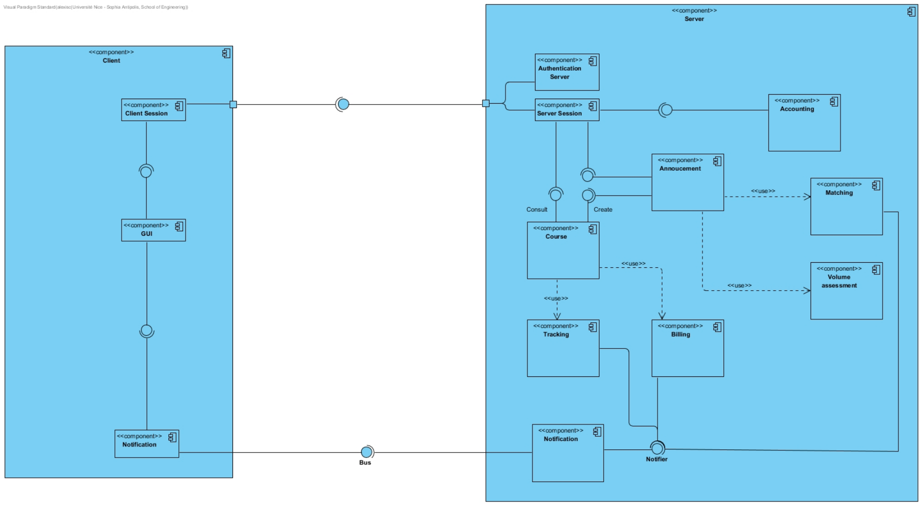

Figure 2:

2.2.4 Technological Stack

- Service Development:
  - Server side: Java Spring Boot
  - Client side: Web page Node.js
- Storage:
  - Database: TBD
- Deployment:
  - Docker Community Engine
  - Docker Compose
- Testing:
  - Acceptance testing: Cucumber
  - Stress testing: Gatling

2.2.5 Roadmap

Different roles will be assigned, such as software architect, tester and dev. Each role will not be permanent and will change every week so everyone has its word to say. It will be easy to assign a small team to a task, since components and tasks are quite in the same scope. The development of the app will be done in small iterations of one weak each. The weeks are planned as follow:

- **Week 41:**
  - Choose technologies to be used
  - External and internal interfaces
  - Mock external systems
- **Week 42:**
  - Continuous integration
  - Walking skeleton
    - Student / Driver entities
    - Announcement creation
    - Matching system *(mock)*
    - Course creation
    - Basic notifications
- **Week 43:**
  - Main risk mitigated
  - Verification / tests of the system
  - Billing system *(mock)*
  - Account management
  - Initiate front interfaces
- **Week 44:**
  - Coding enough of the rest for the POC
  - User Interface
  - Notification front to back
  - Front tests
- **Week 45:**
  - POC complete