Do poprawienia:

1. Wyskakujące błędy i zoptymalizowanie strony
2. Usunąć powtarzający się kod
3. Pozmieniać strukturę plików we frontendzie – wrzucić modele do oddzielnego pliku i tak samo serwisy
4. Odświeżanie list po aktualizacji/dodaniu danych – obecnie jest jak jest.
5. Destiations Create – usunąć błąd, który powstaje jak się zostawi któreś z pól jako puste
6. Destinations Create – usuwać duplikaty dodawanych magazynów albo nie pozwalać na dodawanie duplikatów.
7. Route Update – zaktualizowanie „destinations” danej trasy.
8. Route Details – poprawa algorytmu tak, by w przypadku statusu „NoRoute” trasa była dzielona na dwie i łączona linią i mimo wszystko pokazywana.
9. Destinations Create – drugi algorytm, który sprawdzałby długość każdej możliwej trasy. Złożoność – n!
10. Add Distance – ewentualnie stworzenie tabeli w BD i przerzucenie obliczeń odległości między magazynami do backendu

Funkcjonalności na teraz:

1. Resources Create – dodanie tabeli „resources”, gdzie byłyby przedstawione jakimi produktami dysponuje dany magazyn – możliwość dodawania/aktualizowania surowców oraz przedstawienie ich wraz ze szczegółami danego „warehouse’u” – dodanie Warehouse Details
2. Resources Delivery – możliwość wybrania magazynu i produktu, którego w nim potrzeba – algorytm powinien wyznaczać trasę do magazynów, do których jest najbliżej i posiadają dany produkt
3. Users Create – dodanie użytkowników i wykonanie rejestracji i logowania na stronie – Spring Security

Technologie:

1. Przejść z RxJS na NgRX
2. Protractor – end to end testing
3. Docker
4. Kubernetes
5. Jenkins
6. Wzorce projetkowe – poczytać, ale niekoniecznie się uczyć
7. SOLID
8. REST, SOAP
9. ORM
10. Git, Maven, Gradle
11. Spring Boot
12. Gerrit – code review
13. SonarQube, CheckStyle – analiza kodu do Jenkinsa (albo Mavena albo IntelliJ)
14. LESS, Sass, Stylus – CSS Preporcessor
15. Bootstrap
16. ngRX
17. Responsive Web Design
18. +Hadoop

Database:

1. Location
2. id – long
3. name - string
4. latitude – float
5. longitude – float
6. airport – boolean
7. seaport – boolean
8. Route
9. id – long
10. name – string
11. id\_first\_warehouse – long
12. Destination
13. Id - long
14. id\_route – long
15. id\_warehouse - long
16. order - long

Opis wykonywanych zadań:

1. Pierwszym zadaniem jakie sobie narzuciłem było przypomnienie sobie podstaw Angulara, Springa, Hibernate’a, MySQLa oraz tego jak mogę je połączyć. Zajęło to pierwszych kilkanaście godzin wraz z przerobieniem paru tutoriali i próbą rozwiązania paru problemów, ale całe szczęście etap ten zajął znacznie krócej niż przy tworzeniu aplikacji na Programowanie Zdarzeniowe. Dało mi to podstawowe klasy, które mogłem potem wykorzystać przy tworzeniu właściwej aplikacji. Właśnie stąd są wszelkie elementy związane z Pracownikiem. Będzie to do usunięcia, póki co zostawiam wszystko jakby co.
2. Drugim zadaniem jakie postawiłem przed sobą było wykonanie pierwszej funkcjonalności mojej aplikacji internetowej jaką była możliwość dodawania magazynów na mapie, a także wyświetlania ich listy na mapie
3. Kolejnym zadaniem było dodanie możliwości dodawania tras, a także wyświetlania ich na liście oraz wchodzenie w ich szczegóły, gdzie widać listę magazynów będących częścią danej trasy oraz mapę, na której widać te magazyny oraz trasę między nimi. Kolejność magazynów wyznacza algorytm – pierwszy magazyn jest ustawiony przez użytkownika, reszta zostaje wyliczana, choć będzie to trzeba poprawić.