

# Transport App Case Study

1	Name of the Project	<b>Transport App</b>
2	Objective/ Vision	<p>Create an app to search and view routes for stations Dashboard view with three sections - Display Favorite station, Station statistics</p> <ul style="list-style-type: none"> <li>- This Dashboard is the default view to be shown.</li> <li>- The 3 sections are: <ul style="list-style-type: none"> <li>- Favorite station</li> <li>- Station statistics</li> </ul> </li> <li>- View All Favorite destination under Favorite section</li> <li>- Display all mode of transport under transport statistics section</li> <li>- View all stations information from 3rd party tracks service provider (<a href="https://developer.transportapi.com/">https://developer.transportapi.com/</a>)</li> </ul>
3	Users of the System	All Internet users
4	Functional Requirements	<ol style="list-style-type: none"> <li>1) Home Page should consist of Register page link through which a user can register himself. Upon registration, the user able to login into his account.</li> <li>2) User home page should also have options for to edit his profile and changing his password.</li> <li>3) Search for destination station – Shall provide all mode of transport to travel to the destination</li> <li>4) Nearby stops to destination – Nearby station of the destination</li> <li>5) Transport mode - View all transport modes to a destination from <a href="https://developer.transportapi.com/">https://developer.transportapi.com/</a></li> </ol>
5	Non-functional requirements	<ol style="list-style-type: none"> <li>a) App should be accessible from any location with access to the Internet.</li> <li>b) App should be responsive to display consistently across multiple device screens.</li> <li>c) App should have an intuitive UI that can be operated by novice-expert Internet users</li> </ol>
6	Tools and Technologies to be used	<ol style="list-style-type: none"> <li>1. VCS : Gitlab</li> <li>2. Middleware : Spring Boot</li> <li>3. Frond end : Angular/React</li> <li>4. Data Store : MongoDB / MySQL</li> <li>5. Testing : JUnit, Mocha, Chai, Jest, Protractor</li> <li>6. Container : Docker</li> <li>7. Bug Fix : Sonarlint</li> <li>8. CI : Gitlab</li> </ol>

# User Stories

1	As a user I should be able to register with the application so that I can login and use the functionalities of the application.
2	As a user, I should be able to login with my user name and password in order to access the functionalities of the application.
3	As a user, I should be able to login with my Gmail account in order to access the functionalities of the application.(optional requirement)
4	As a user I should be able to search resources to view their details
5	As a user, I should be able to save resources to a wishlist/favourite so that I can access them later
6	As a user, I should be able to access items saved to my wishlist/favourite

Notes:

- The application should be based on microservices architecture
- API Gateway pattern should be implemented using Spring Cloud Gateway
- Services should register themselves with Eureka Service Discovery server.
- All layers of microservices should be covered with automated unit and integration tests
- All microservice endpoints should have API documentation

## High Level Architecture Diagram

