**Introduction**

This project is a Ticket Search and Recommendation Engine based on the usrs' Geo-location and likes.

Click [Demo](https://fanzhang4530com.files.wordpress.com/2017/09/myticket.mp4) to see the demo.

Click [link](http://13.58.80.248/MyTicket/) to visit the website.

The [code](https://github.com/Fan4530/MyTicket) is here.

[wpvideo plic9rEk]

**Requirements**

1. [JDBC](https://fanzhang4530com.files.wordpress.com/2017/09/jdbc.pdf)
2. [JASON Lib](https://fanzhang4530com.files.wordpress.com/2017/09/jason-lib.pdf)

**Installation**

1. Install Java 8
2. Install IntelliJ
3. [Setup Tomcat Server](https://fanzhang4530.com/2017/09/21/tomcat/)
4. Install Postman
5. [Ticket Master Key](http://developer.ticketmaster.com/products-and-docs/apis/getting-started/)
6. [Install MAMP](https://fanzhang4530com.files.wordpress.com/2017/09/install-mamp.pdf)

**Recommended modules**

You can change to MongoDB database if there are large amount of data.

You can change by: In Java Factory, change the default base from MySQL to MongoDB.

**Configuration**

Click the  link to enter the [login page.](http://13.58.80.248/MyTicket)  
Username: 1111   Password: 2222 (for test).

Click Login button to enter the search page. There are three selections: Nearby, My Favorites, Recommendation.

Nearby:

In this page, users are able to search the nearby tickets and add them to favorites by click the "heart" button.

My Favorites:

This page shows the users' favorites item. Users can also cancel the favorite item here.

Recommendation:

This page shows the recommended items based on the users' favorites and Geo-location.

src/

• rpc/

o The entry point of the project, handles rpc request/response, parsing, etc.

o SearchItem.java

o RecommendItem.java

o ItemHistory.java

o RpcHelper.java

o Login\_in.java

• db/

o The ‘backend’ of the project, connects to database.

o DBConnection.java is an interface. (implement it in mysql and mongodb)

o DbConnectionFactory.java is a factory class.

o mysql/

o mongodb/

• externel/

o Another ‘backend’ of theiq project, connects to public APIs.

o ExternalAPI.java is an interface. All supported backends should implement it.

o ExternalAPIFactory.java is a factory class.

o TicketMasterAPI.java. ‘Backend’ of our project, connects to TicketMaster API.

• entity/

o Handles creation/conversion/etc of object instances.

o Item.java

• algorithm/

o event recommendation algorithms.

o GeoRecommendation.java

o Recommendation.java

Let me show how this files work:

client --> call servlet : login.java ---------|

|

|

---------------------------------------------|

| | |

| | |

| | |

ItemHistory.java SearchItem.java RecommendItem.java <------- Recommendation.java <--- GeoRecommendation.java

^

|

DbConnectionFactory ---------------------------------------------

/ \

Connection(implemented by) ---> MysqlDBConnection MongoDbDBConnection <-----------------

^ ^ |

| | |

mysql(creation) mongodb(creation) |

|

|

|

ExternalAPIFactory(create db) -------|

^ ^ ^

/ | \

ExternalAPI interface -->(implement by) TicketMasterAPI .... ...