# User's Guide

## Prerequisites:

To Run correctly the application, you must have this list of programs installed on your device:

- IDE (like IntelliJ or Eclipse)
- PostgreSQL
- PostMan (for demo)

## Run the BackEnd:

1. Clone the application code from the link https://github.com/gabriele22/it.uniroma2.progISSSR.git (you must create an account GitHub to do this) using the IDE
2. Then you have to create a DB on postgres to save the data of the application.
3. Config the file src.main.resource.application.properties modifying the line
    a. spring.datasource.url="URLOfDataBasePostgres";
    b. spring.datasource.username="usernamePostgres";
    c. spring.datasource.password="passwordPostgres";
4. Run the file src.main.java.it.uniroma2.TicketingSystemApplication.java

## Setting the FrontEnd

1. Clone the application code from the link https://github.com/Melissa95/it.uniroma2.frontEnd.git

(you must create an account GitHub to do this) using the IDE

## Run the demo

You can test your application running a postman run that populate the DB. You must do:

1. Open Postman application (you must create an account PostMan)
2. Click on Import --> Import File --> Choose File and select the file ticketingsystem.postman_collection.json
3. Click on runner in the main postman interface and in the new window open click on ticketingsystem --> run ticketingsystem

If all tests are passed, the DB is correctly populated. The next page synthetize the new data insert in the DB.

- Users put into the system with related access' credential:

| Username | Password |
|---|---|
| admin | admin123 |
| customer | customer |
| teamLeader1 | teamLeader1 |
| teamLeader2 | teamLeader2 |
| teamCoordinator1 | teamCoordinator1 |
| teamCoordinator2 | teamCoordinator2 |
| teamMember1 | teamMember1 |
| teamMember2 | teamMember2 |
| teamMember3 | teamMember3 |
| teamMember4 | teamMember4 |
| teamMember5 | teamMember5 |

- Team put into the system with related elements:

| Name | Leader | Coordinator | Members |
|---|---|---|---|
| team1 | teamLeader1 | teamCoordinator1 | teamMember1<br>teamMember2 |
| team2 | teamLeader2 | teamCoordinator2 | teamMember3<br>teamMember4<br>teamMember5 |

- Targets put into the system:
    - Product1
    - Product2
    - Service1
- Tickets put into the system with its related attributes:

| Title | Status | Creator |
|---|---|---|
| ticket1 | pending | admin |
| ticket2 | pending | admin |
| ticket3 | pending | customer |
| ticket4 | pending | teamLeader1 |
| ticket5 | pending | customer |
| ticket6 | pending | customer |
| ticket7 | pending | admin |
| ticket8 | pending | teamLeader2 |
| ticket9 | closed | customer |
| ticket10 | closed | teamLeader1 |
| ticket11 | closed | admin |
| ticket12 | closed | admin |
| ticket13 | execution | admin |
| ticket14 | execution | customer |

- The application starts with an escalation 0.2 * customerPriority + 0.5 * teamPriority + 0.3 * pendingTime.
- The simulation starts with the equality relation between:

    - ticket12 == ticket10
    - ticket13 == ticket1
- The simulation starts with the dependent relation between (where the relation A --> B means that to solve B I must first had solve A):
    - ticket1 --> ticket2 --> ticket3
    - ticket4 --> ticket5
    - ticket4 --> ticket6