

# Best Tickets Parking Management System – Installation Guides

---

This document provides detailed installation instructions for different user roles involved in setting up the Best Tickets Parking Management System.

## Table of Contents

- [Administrator Guide: Complete System Setup](#)
  - [Hospital IT Guide: Payment Terminal Setup](#)
  - [Gate Operator Guide: Gate Software Setup](#)
- 

## Administrator Guide: Complete System Setup

---

This guide is intended for system administrators who need to set up the entire Best Tickets system, including the database, server, and admin software.

## Prerequisites

- Server
- MySQL Server 5.7+ or compatible
- .NET 6.0 SDK and Runtime
- .NET 9.0 SDK and Runtime
- Python 3.9+ with pip
- Administrative privileges on all systems
- Network connectivity between all components

# Step 1: Database Setup

## 1. Install MySQL Server:

- Download MySQL from [official website](#)
- Follow the installer instructions for your operating system
- Set a secure root password during installation

## 2. Create Database and User:

```
CREATE DATABASE best_tickets CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE USER 'best_tickets_user'@'%' IDENTIFIED BY 'strong-password-here';  
GRANT ALL PRIVILEGES ON best_tickets.* TO 'best_tickets_user'@'%';  
FLUSH PRIVILEGES;
```

## 3. Configure MySQL Settings:

- Edit the MySQL configuration file ( `my.cnf` or `my.ini` )
- Ensure the server accepts remote connections if needed:  

```
bind-address = 0.0.0.0
```
- Adjust other performance parameters as needed
- Restart MySQL service

## 4. Secure the MySQL Installation:

- Run the MySQL secure installation script

```
mysql_secure_installation
```

- Follow the prompts to secure your MySQL installation

# Step 2: Ticket Server Setup

## 1. Clone or Download the Repository:

```
git clone https://github.com/jere344/best-tickets.git
cd best-tickets/ticket-server
```

## 2. Create a Python Virtual Environment:

```
python -m venv venv
```

## 3. Activate the Virtual Environment:

- Windows:

```
venv\Scripts\activate
```

- Linux/Mac:

```
source venv/bin/activate
```

## 4. Install Required Dependencies:

```
pip install -r requirements.txt
```

## 5. Configure Database Connection:

- Open `main.py` and update the `DATABASE_URI` variable with your database connection details:

```
DATABASE_URI = 'mysql+pymysql://root:your-password-here@localhost:3306/'
# or for the school server:
# DATABASE_URI = 'mysql+pymysql://dev-2230460:[pass]@sql.decinfo-cchic.
```

## 6. Configure Server Network Settings:

- Open `main.py` and locate the `waitress.serve()` function near the bottom of the file
- Update the host parameter to specify which network interface to listen on:

```
waitress.serve(app, host='0.0.0.0', port=PORT, threads=4) # Listen on
# or
```

```
waitress.serve(app, host='127.0.0.1', port=PORT, threads=4) # Listen o
# or
waitress.serve(app, host='your.server.ip.address', port=PORT, threads=4
```

- Note: Using '0.0.0.0' allows connections from any network interface, which is typically needed for production

## 7. Initialize the Database:

- Ensure your MySQL server is running
- Create an empty database if it doesn't exist:

```
CREATE DATABASE IF NOT EXISTS `best_tickets` CHARACTER SET utf8mb4 COLL
```

- Navigate to the admin-software directory:

```
cd ../admin-software
```

- Replace the connection strings in the `appsettings.json` file with your MySQL server details:

```
{
  "ConnectionStrings": {
    "ReleaseConnection": "Server=votre-serveur-mysql;Port=3306;Databa
    "DefaultConnection": "Server=sql.deinfo-cchic.ca;Port=33306;Data
    "Test": ""
  }
}
```

- If you are building in Release mode, use the ReleaseConnection string
- If you are using the school server, use the DefaultConnection string
- When in doubt, just edit both
- Install the Entity Framework Core CLI tools if not already installed:

```
dotnet tool install --global dotnet-ef
```

- Apply the database migrations to create all tables and schema:

```
dotnet ef database update --connection "Server=localhost;Database=best_
```

- This will create all necessary database tables and relationships using the migrations defined in the admin panel project

## 8. Test the Server:

```
python main.py
```

The server should start and be accessible at <http://your-server-ip:your-port>

You should see a message indicating "Parking Ticket Server Starting" and "Waitress server running..."

## 9. (optional) Configure as a Service:

- Windows: Use NSSM or create a Windows Service
- Linux: Create a systemd service file

```
# /etc/systemd/system/best-tickets-server.service
[Unit]
Description=Best Tickets API Server
After=network.target mysql.service

[Service]
User=ubuntu
WorkingDirectory=/path/to/best-tickets/ticket-server
ExecStart=/path/to/best-tickets/ticket-server/venv/bin/python main.
Restart=always
RestartSec=5
Environment=PYTHONUNBUFFERED=1

[Install]
WantedBy=multi-user.target
```

Then enable and start the service:

```
sudo systemctl enable best-tickets-server
```

```
sudo systemctl start best-tickets-server
```

## 10. Configure Firewall:

- Allow incoming connections on the server port (default: 5000)
- Windows:

```
netsh advfirewall firewall add rule name="Best Tickets Server" dir=in a
```

- Linux(UFW):

```
sudo ufw allow 5000/tcp
```

# Step 3: Admin Software Setup

## 1. Prerequisites:

- Windows 10/11 with .NET 6.0 Runtime installed
- Network connectivity to the database and server

## 2. Installation:

- Navigate to the admin-software directory
- Build or obtain the admin software installation package

```
dotnet publish -c Release -r win-x64 --self-contained true
```

- Copy the published files to the administrator's computer

## 3. Configuration:

- Edit the `appsettings.json` file:

```
{
  "ConnectionStrings": {
    "ReleaseConnection": "Server=your-mysql-server;Port=3306;Database=
    "DefaultConnection": "Server=sql.decinfo-cchic.ca;Port=33306;Database=
    "Test": ""
  }
}
```

}

- If building in Release mode, use the ReleaseConnection string
- If using the school server, use the DefaultConnection string

When in doubt, just edit both

#### 4. First Launch and Initialization:

- Run the admin software ( `admintickets.exe` )
- Login with the default credentials:
  - Username: admin
  - Password: admin
- **IMPORTANT:** Immediately change the default password

#### 5. Configure your first Hospital in the system:

- Go to "Hospitals" section
- Click "Add New Hospital"
- Fill in all required details
- Set a secure gateway password for the hospital (write this down, you'll need it for gate setup)

#### 6. Create Administrator Accounts:

- Go to the "Admin" section
- Create new administrator accounts
- For each new account, copy the temporary password and provide it to the user
- Instruct users to change their password on first login
  - > Go to "Profile" with the profile icon in the top right corner
  - > fill in the "Old Password" field with the temporary password
  - > fill in the "New Password" field with the new password
  - > fill in the "Confirm Password" field with the new password
  - > click on the "Save" button

#### 7. Configure System Settings:

- Set up tax rates
- Configure price brackets
- Set up subscription tiers

- Configure discount codes if needed

## Step 4: System Testing


### 1. Verify Database Connectivity:

- Ensure the admin software can connect to the database
- Test creating, updating, and deleting records

### 2. Test API Server Functionality:

- Verify the server is running
- Test API endpoints using a tool like Postman
- Verify authentication is working

Test: Create a Ticket (Valid Credentials)

 **Expected:** Returns a JSON object with a new ticket and hospital details.

```
curl -X POST http://[server-address:port]/create_ticket ^  
  -H "Content-Type: application/json" ^  
  -d "{ \"hospital_id\": [id], \"password\": \"[correct-password]\" }"
```

### 3. Validate Hospital Configuration:

- Ensure hospital details are correctly stored
- Verify price brackets are properly applied

## Step 5: Backup and Maintenance Plan

### 1. Database Backup Strategy:

- Configure automated MySQL backups

```
mysqldump -u root -p best_tickets > backup_$(date +%Y%m%d).sql
```

- Store backups in a secure, off-site location
- Test backup restoration procedure



## 2. Update Strategy:

- Document the update procedure for all system components
- Create a test environment for validating updates before deployment

## 3. Monitoring Setup:

- Configure server monitoring tools
- Set up alerts for system issues
- Regularly review logs for potential problems

# Troubleshooting

## Database Connection Issues

- Verify MySQL service is running
- Check connection strings in appsettings.json
- Confirm firewall rules allow database traffic
- Verify the database user has proper permissions

## API Server Not Responding

- Check if Python service is running
- Verify network connectivity to the server
- Check server logs for errors
- Validate environment configuration

## Admin Software Crashes

- Check Windows Event Logs
  - Verify .NET Runtime is properly installed
  - Ensure all dependencies are available
  - Check for disk space issues
-

# Hospital IT Guide: Payment Terminal Setup

---

This guide is for hospital IT staff who need to set up payment terminals and configure their hospital in the Best Tickets system.

Hospital IT may want to use their own server or setup the whole architecture in local network for security and privacy reasons., to do that refer to the [Administrator Guide](#).

## Prerequisites

- Information from system administrator:
  - Database connection string
  - Server API address
  - Admin software credentials
- Computer hardware for payment terminal:
  - Windows 10/11 computer
  - Touch screen (recommended)
  - Receipt printer
  - Card reader (if using integrated payment)
  - QR code scanner
  - Network connectivity
- .NET 6.0 Runtime installed

## Step 1: Hospital Registration (via Admin Software)

### 1. Obtain Admin Software and Credentials:

- Get the admin software installation package or portable executable from the system administrator
- Receive login credentials with appropriate permissions

### 2. Login to Admin Software:

- Launch the admin software
- Enter provided credentials

### 3. **Configure Hospital:**

- Navigate to "Hospitals" section
- Click "Add New Hospital" if your hospital isn't already registered
- Complete all required fields:
  - Hospital Name
  - Address
  - logo
- Set a secure gateway password (write this down, you'll need it for gate setup)
- Save the configuration

### 4. **Configure Price Brackets:**

- Go to "Price Brackets" section
- Set up pricing structure based on hospital requirements
  - If you don't setup the price brackets, the global price brackets will apply
- Configure different rates for different durations
- Save the pricing configuration

### 5. **Set Up Discount Codes** (if needed):

- Navigate to "Discount Codes" section
- Create codes for special rates (e.g., for patients, staff)
- Configure discount percentages or fixed amounts
- Set expiration dates if applicable

### 6. **Configure Subscription Tiers** (if offering subscriptions):

- Go to "Subscription Tiers" section
- Create different subscription levels
- Set pricing, durations and number of uses per day
  - If you don't setup the subscription tiers, the global subscription tiers will apply

## Step 2: Payment Terminal Installation

### 1. **Prepare the Terminal Hardware:**

- Set up the computer in the designated payment kiosk location
- Connect the receipt printer, card reader, and QR scanner

- Ensure all devices are working properly with their respective drivers
- Connect to the hospital network

## 2. Install Payment Terminal Software:

- Get the payment terminal installation package
- Extract or run the installer

## 3. Configure Payment Terminal:

- Edit the `appsettings.json` file:

```
{
  "ConnectionStrings": {
    "ReleaseConnection": "Server=your-mysql-server;Port=3306;Database=your-database;User=root;Password=your-password",
    "DefaultConnection": "Server=mysql.deinfo-cchic.ca;Port=33306;Database=your-database;User=root;Password=your-password",
    "Test": ""
  }
}
```

- If building in Release mode, use the ReleaseConnection string
- If using the school server, use the DefaultConnection string

When in doubt, just edit both

- The payment terminal connects directly to the database using the connection string. It is important to ensure the payment terminal is physically secure and that the connection string is not exposed.

## 4. Test the Configuration:

- Launch the payment terminal application
- Verify it connects to the database
- Confirm it can retrieve hospital-specific pricing
- Test the printer functionality by crediting a test ticket

## 5. Configure Hardware Peripherals:

- Test QR code scanner functionality
- Configure receipt printer settings
- Set up card reader integration if applicable

## Step 3: Test Payment Workflow

### 1. Create Test Tickets:

- Use an entry gate to generate a real test tickets

### 2. Process Test Payments:

- Scan the test ticket at the payment terminal
- Verify correct pricing is applied
- Test payment processing
- Confirm receipt printing

### 3. Test Subscription Management:

- Create a test subscription
- Issue a subscription card
- Test subscription validation on a new test ticket

## Step 4: Prepare Gate Software Configuration

### 1. Obtain Gate Software Package:

- Download the gate software installation package
- Extract the package to a temporary location for configuration

### 2. Configure API Connection:

- Locate the `.env` file in the gate software package
- Edit the `API_BASE_URL` parameter to point to your server:

```
API_BASE_URL=http://your-server-address:port
```

- Replace "your-server-address" with the actual IP address or hostname of your API server
- Replace "port" with the port number the API server is running on
- Save the `.env` file

### 3. Prepare Installation Package for Gate Operators:

- Create a package containing:
  - The pre-configured gate software with updated `.env` file
  - Installation instructions for gate operators
  - Hospital gateway password (securely communicated)
  - Contact information for support

#### 4. **Documentation for Gate Operators:**

- Provide gate operators with:
  - Hospital name and gateway password
  - package containing the gate software with updated `.env` file
  - Hardware requirements
  - Installation checklist
  - Contact information for technical support

## Step 5: Staff Training

#### 1. **Conduct Training Sessions:**

- Create quick reference guides for common tasks
- Train front desk staff on basic operation
- Train technical staff on troubleshooting
- Document support contact information

## Admin Software Settings

After installation, administrators can customize the application experience:

#### 1. **Change Language and Theme:**

- Click on the settings cog icon in the top right corner
- Select your preferred language from the dropdown menu
- Choose between light or dark theme
- Save your preferences

#### 2. **User Profile Settings:**

- Access profile settings using the profile icon in the top right corner
- Change password and other user-specific settings

# Troubleshooting

## Terminal Cannot Connect to the Database

- Check network connectivity
- Verify connection string in configuration
- Ensure MySQL server is running

## Printing Issues

- Verify printer is properly connected and has paper
- Check printer driver installation

## Scanner Not Reading QR Codes

- Check scanner connection
  - Verify scanner drivers are installed
  - Test with different QR codes
  - Adjust ambient lighting if needed
  - Try to scan the QR code with a phone to verify the code is valid
- 

# Gate Operator Guide: Gate Software Setup

---

This guide is for technicians responsible for setting up the entry and exit gates with the Best Tickets system.

## Prerequisites

- Information from hospital IT:
  - Pre-configured gate software package with API connection settings
  - Hospital gateway password
- Hardware components:
  - Windows computer with .NET 9.0 Runtime

- QR code scanner/reader
- Barrier gate controller hardware
- Ticket printer (for entry gates)
- Network connectivity
- Optional: Display screen for status information

## Step 1: Hardware Installation

### 1. Set Up Gate Computer:

- Install Windows on the gate computer
- Install .NET 9.0 Runtime
- Connect to power and network
- Position in a weather-protected location

### 2. Connect Peripheral Devices:

- Install the QR code scanner
- Connect the barrier gate controller
- Connect the ticket printer (entry gates only)
- Wire all connections securely
- Label all cables for future maintenance

### 3. Test Hardware Components:

- Verify scanner can read QR codes
- Test printer functionality (entry gates)
- Confirm barrier gate can be controlled

## Step 2: Gate Software Installation

### 1. Install Gate Software:

- Obtain the pre-configured gate software package from hospital IT
- Run the installer or extract files to a dedicated directory
- Create a desktop shortcut for easy access
- Note: The API connection has already been configured in the `.env` file by hospital IT



## 2. Initial Configuration:

- Launch the gate software for the first time
- The software will present the First Setup wizard
- Enter the following information:
  - Select your hospital from the dropdown list
  - Enter the hospital gateway password
  - Select gate type (entry or exit)
- Complete the setup wizard by restarting the software

## Step 3: Test Gate Operation

### 1. Entry Gate Testing:

- Trigger a new ticket request
- Verify ticket prints with correct QR code
- Confirm gate opens
- Validate the ticket was recorded in the system

### 2. Exit Gate Testing:

- Create and pay for a test ticket using payment terminal
- Scan the paid ticket at exit gate
- Confirm gate opens
- Verify ticket is marked as used in the system

## Step 4: Physical Gate Setup

### 1. Position Physical Components:

- Mount QR scanner at vehicle window height
- Ensure ticket dispenser is easily accessible (entry gates)
- Position any instructional signage

### 2. Install Safety Measures:

- Configure safety sensors to prevent gate closure on vehicles
- Test all safety mechanisms thoroughly

- Ensure all wiring is properly protected

### 3. **Weather Protection:**

- Install weather protection for all components
- Ensure adequate drainage
- Provide heating for cold environments if necessary

## Troubleshooting

### Resting the Gate Software after selecting the wrong hospital :

- Close the gate software
- Delete the `settings.json` file in the gate software directory
- Restart the gate software to re-run the First Setup wizard

### Gate Software Cannot Connect to Server

- Call the hospital IT guy who should :
  - Verify network connectivity
  - Ensure the `.env` file contains the correct `API_BASE_URL`
  - Check that the server is running and accessible
  - Ensure firewall allows connection

### Scanner Not Reading QR Codes

- Test the QR code validity with your phone or another scanner
- Check scanner connection

### Gate Not Responding to Commands

- Check physical connections to gate controller
- Verify controller has power

### Ticket Printer Issues (Entry Gates)

- Verify printer has paper
- Check printer connection
- Confirm printer driver is properly installed
- Test with Windows print dialog

# Emergency Procedures

## 1. System Restoration:

- Restart gate software
- If issues persist, reboot the computer
- delete the `settings.json` file in the gate software directory to re-run the first setup wizard
- For persistent problems, contact system administrator

---

# Additional Resources

## Network Requirements

Component	Port	Protocol	Description
MySQL Database	3306	TCP	Database communication
API Server	5000	TCP	HTTP API communications
Admin Software	N/A	TCP	Client to server/database
Payment Terminal	N/A	TCP	Client to server/database
Gate Software	N/A	TCP	Client to server

## Maintenance Procedures

### Regular Database Maintenance

- Schedule regular database backups

- Monitor database size and performance

## Support Contact Information

For assistance with installation or troubleshooting:

- **Technical Support Email:** [jeremy.guerin34@yahoo.com](mailto:jeremy.guerin34@yahoo.com)
- **Documentation Website:** <https://github.com/jere344/parking-management-suite>

---

Document Version: 1.0

Last Updated: March 2025