Photo Dispatcher Application Document

# Index

1. Introduction
2. General Description
3. System Architecture
4. Workflow
5. Installation and Configuration
6. Using the Application
7. Security and Privacy
8. Maintenance and Updates
9. Troubleshooting
10. Contacts

# 1. Introduction

This documentation describes the operation and implementation of an application designed to automatically send photos taken by photographers at events to the email addresses of attendees. Each photo is named with a progressive number corresponding to the pass number of the person depicted. Event organizers provide a CSV file containing the mapping between pass numbers and associated emails.

# 2. General Description

The application allows photographers to upload photos taken during an event. Each photo must be named with the pass number of the person depicted. The application uses a CSV file provided by event organizers to map pass numbers to email addresses and automatically sends the photos to the corresponding addresses. Additionally, the application allows specifying the local path of the folder containing all numbered photos from which they will be taken for sending.

## Main Features:

* Photo upload
* Using a CSV file for mapping pass numbers to emails
* Specifying the local path of the folder with photos
* Sending personalized emails with attached photos
* User interface for uploading and managing photos

**3. System Architecture**

The application consists of several key components:

* User Interface (UI): for specifying the folder path and managing operations.
* Backend: for photo processing and email sending.
* Database: for storing information about sent emails and users.
* Email Service: for sending emails with attached photos.

**4. Workflow**

1. **Photo Upload**: The photographer renames the photos with the pass numbers and saves them in a local folder.
2. **CSV File Upload**: Organizers upload the CSV file containing the mapping between pass numbers and email addresses.
3. **Folder Path Specification**: The user specifies the local path of the folder containing the photos through the user interface.
4. **Email Extraction**: The application uses the CSV file to map the pass number extracted from the file name to the corresponding email address.
5. **Email Sending**: The backend retrieves the photos from the specified folder and sends an email to the extracted address with the photo attached.
6. **Send Confirmation**: The system records the send in the database and notifies the user of success or error.

**5. Installation and Configuration**

## Email Configuration:

* Configure SMTP credentials in the configuration file for sending emails.

## CSV File Configuration:

* Ensure that the CSV file contains two columns: PassNumber and Email.
* Upload the CSV file through the application's user interface.

## Folder Path Configuration:

* Specify the local path of the folder containing the numbered photos through the application's user interface.

## Compile the Project:

* Open the solution with a compatible IDE (Visual Studio, Visual Studio Code, etc.).
* Compile the Photo Dispatcher and PhotoDispatcherView projects.

## Configure the Settings:

* Ensure that the appsettings.json file is located in the Photo Dispatcher folder and is configured correctly (it must have “Copy Always” set in the properties).

# 6. Using the Application

* Run the executable PhotoDispatcherView.exe to open the graphical interface.

## Configure the Settings:

* **Photos Directory**: Enter the path of the folder where the photos are stored.
* **CSV File Path**: Enter the path of the CSV file containing the mappings between pass numbers and email addresses.
* **Email Settings**: Configure the following parameters:
  + **From Name**: The displayed name of the sender.
  + **SMTP Server**: The address of the SMTP server (e.g., smtp.office365.com).
  + **SMTP Port**: The port used for the SMTP server (e.g., 587).
  + **SMTP User**: The email address of the account.
  + **SMTP Pass**: The password of the email account.
  + **Email Subject**: The subject of the email.
  + **Email Body**: The body of the email.

## Save the Settings:

* Click the **Save** button to save the settings in the appsettings.json file. The PhotoDispatcher.exe application will automatically start, which will retrieve the photos from the specified folder and automatically send them to the corresponding email addresses.

## Monitor the Process:

* The application will log status information in the console and in the report.txt file located in the same directory as the executable.

## Log Report

After each execution, a report.txt file will be generated with the following details:

* **Info Logs**: General information about the execution.
* **Warning Logs**: Warnings generated during the execution.
* **Error Logs**: Errors encountered during the execution.

**7. Security and Privacy**

* **Data Protection**: Use security protocols to protect user data.
* **Photo Retention**: Photos should be retained only for the necessary time for sending.
* **Data Access**: Limit access to sensitive data to authorized users only.

**8. Maintenance and Updates**

* **Software Updates**: Ensure the application is kept up to date with the latest security patches.
* **Data Backup**: Regularly back up the database and photos.

**9. Troubleshooting**

## Emails Not Sent:

* Check that the SMTP credentials are correct.
* Ensure the SMTP server is reachable.
* Verify the CSV format and ensure the emails are valid.

## Connection Errors:

* Verify the internet connection.
* Check firewall settings and ensure necessary ports are open.

# 10. Contacts

For further information or support, contact the system administrator at: acanfora\_7200@hotmail.com.