Criterion B: Design

The starting page for the user is the login page which includes a simple panel for the login. When the user inputs login information, it is checked with the information stored in the database. If these two pieces of information match, the user will be redirected to the main page. Otherwise, he will not be allowed to enter.

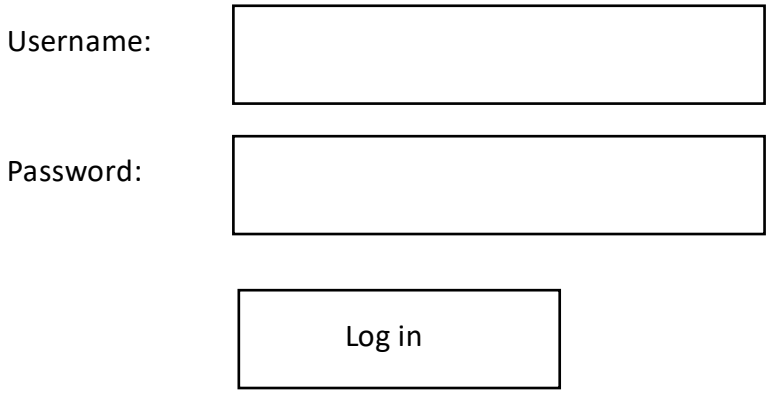

Figure 1. Scheme of the login page

When the user fills in incorrect login username or password, he will be redirected to the page, from which he can easily get back to the login page.

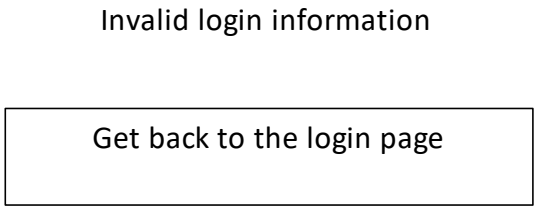

Figure 2. When user fills in wrong login information

In the main page, the user can upload a chosen .csv file to the database, download the last-updated .csv file from the database, publish the debate schedule table and send notification which will be stored in the database. He can also log out of the panel.

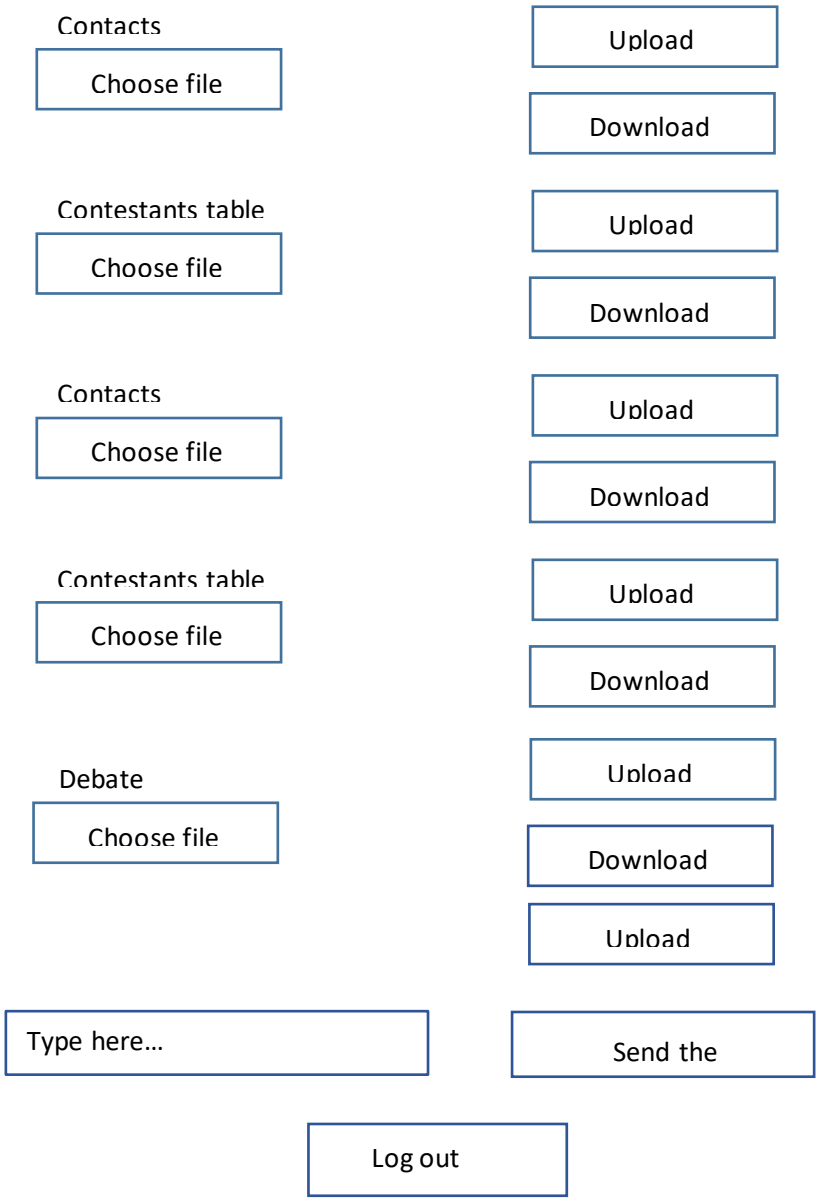

Figure 3. Main menu page which includes user interface for all functions that the user might want to perform.

Illustrated workflow

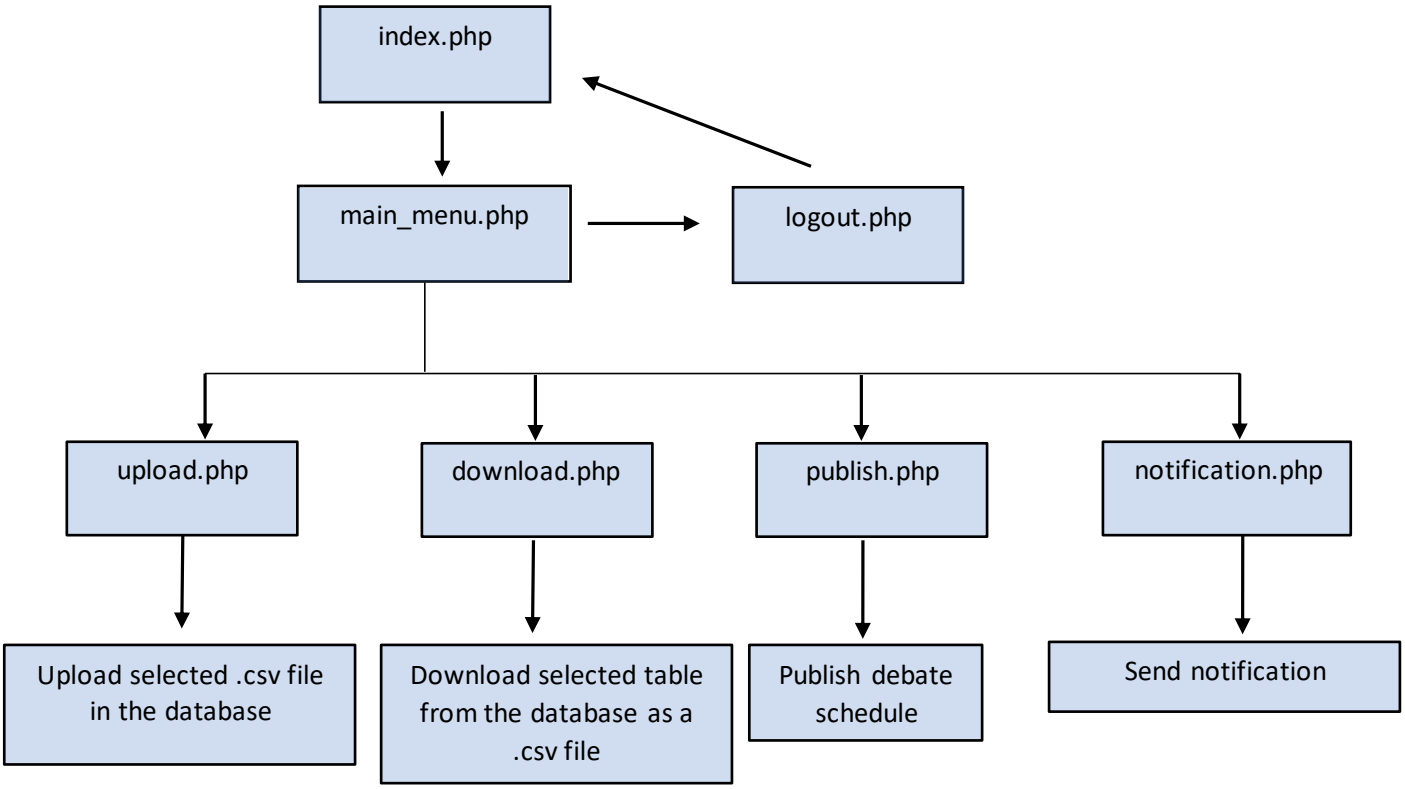

Figure 4. Illustrated workflow of the functions that user can perform in the administrator panel.

Structure of the website

- index.php – login page
- connect.php – connects to the database
- login_parse.php – enables log in to the administrator panel
- logout_parse.php – enables log out of the administrator panel
- main_menu.php – graphical portfolio for the user where he can perform functions: upload & download a .csv file, send notification and publish debate schedule
- upload.php – uploads selected .csv file to the database
- download.php – downloads selected table from the database as .csv file
- publish.php – transforms content of unpublished debate schedule table into published debate schedule table
- notification_panel.php – enables user to send notification to the database

Structure of communication with the application

- login.php – provides login of a contestant to the application
- confirmation.php – verifies contestant's login using email
- personalData.php – sends name and team of the contestant based on his device id
- updated.php – handles communicates with the application using the database
- notification_app.php – sends new notifications to the application

Graphical user interface

- style.css – provides design for the main menu page

Use of database

Data insertion into the database

Data insertion means adding new data to the database. For instance, this will be requested when the user wants to send a notification or simply upload a new .csv file into the database.

Data deletion from the database

Data deletion will be request when a contestant logs out of the application.

Data selection from the database

Data selection will be used when the data in the database needs to be downloaded or just compared to the data from the application.

Data update in the database

Data update will be used every time when the user uploads a file or sends notification to update the time of the latest upload. When time from the application is then received, communication between the database and the application will be successful only if the time in the database is 'newer' than time of latest update in the application.

Program flowchart: Table update

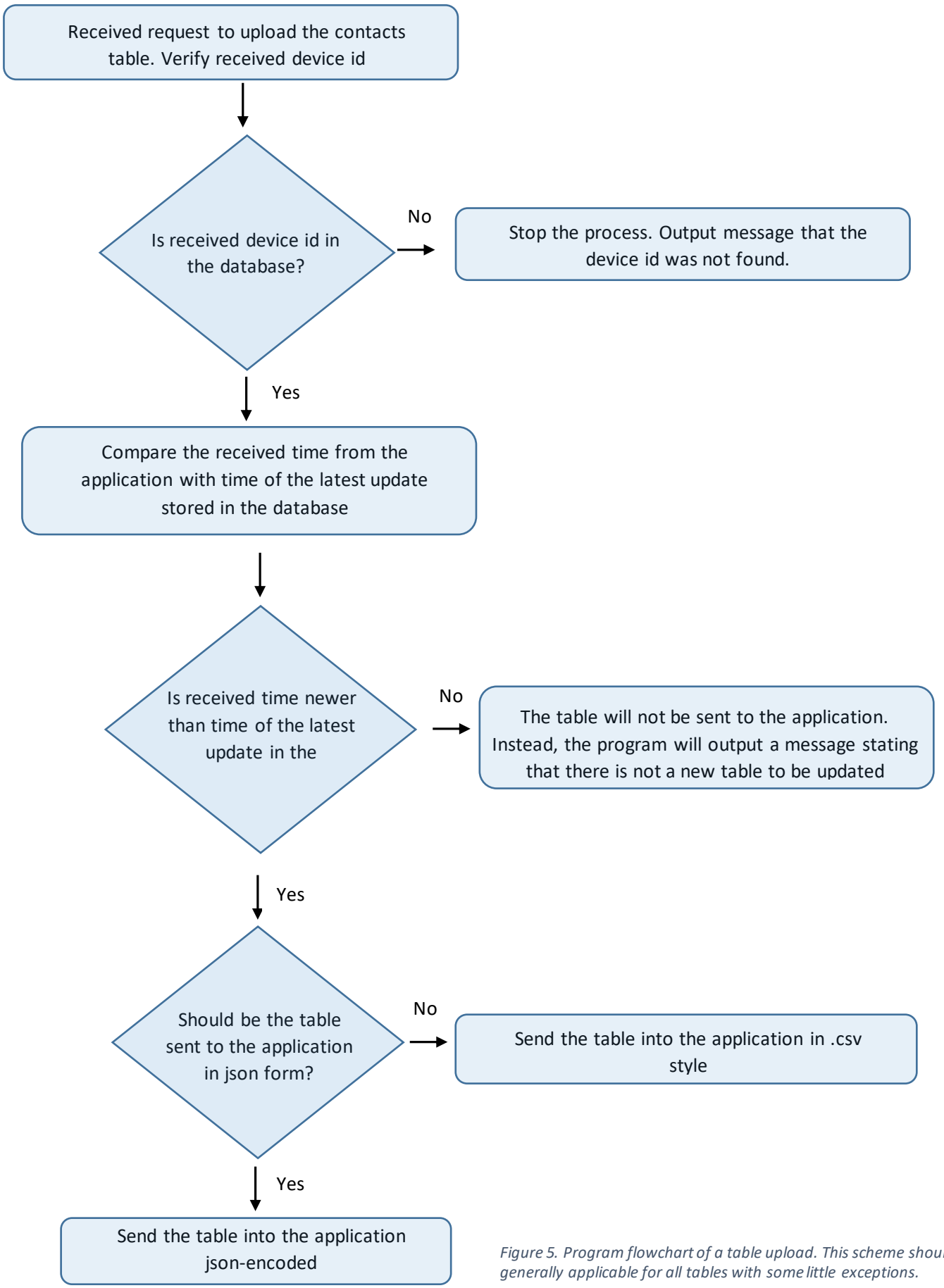

Figure 5. Program flowchart of a table upload. This scheme should be generally applicable for all tables with some little exceptions.

Program flowchart: Login to the application

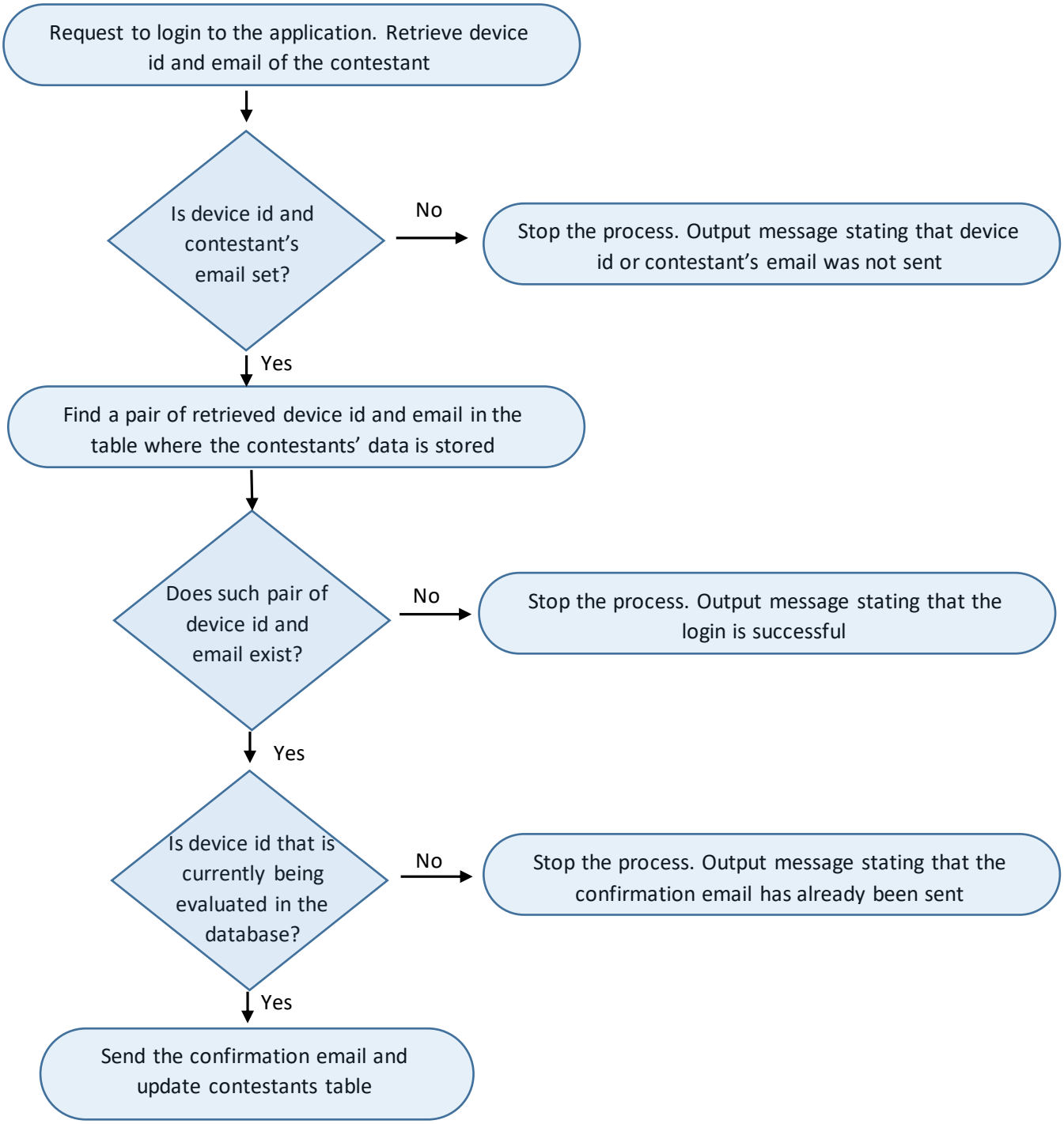

Figure 6. Program flowchart of a contestant's login into the application.

Program flowchart: Confirmation of the login to the application

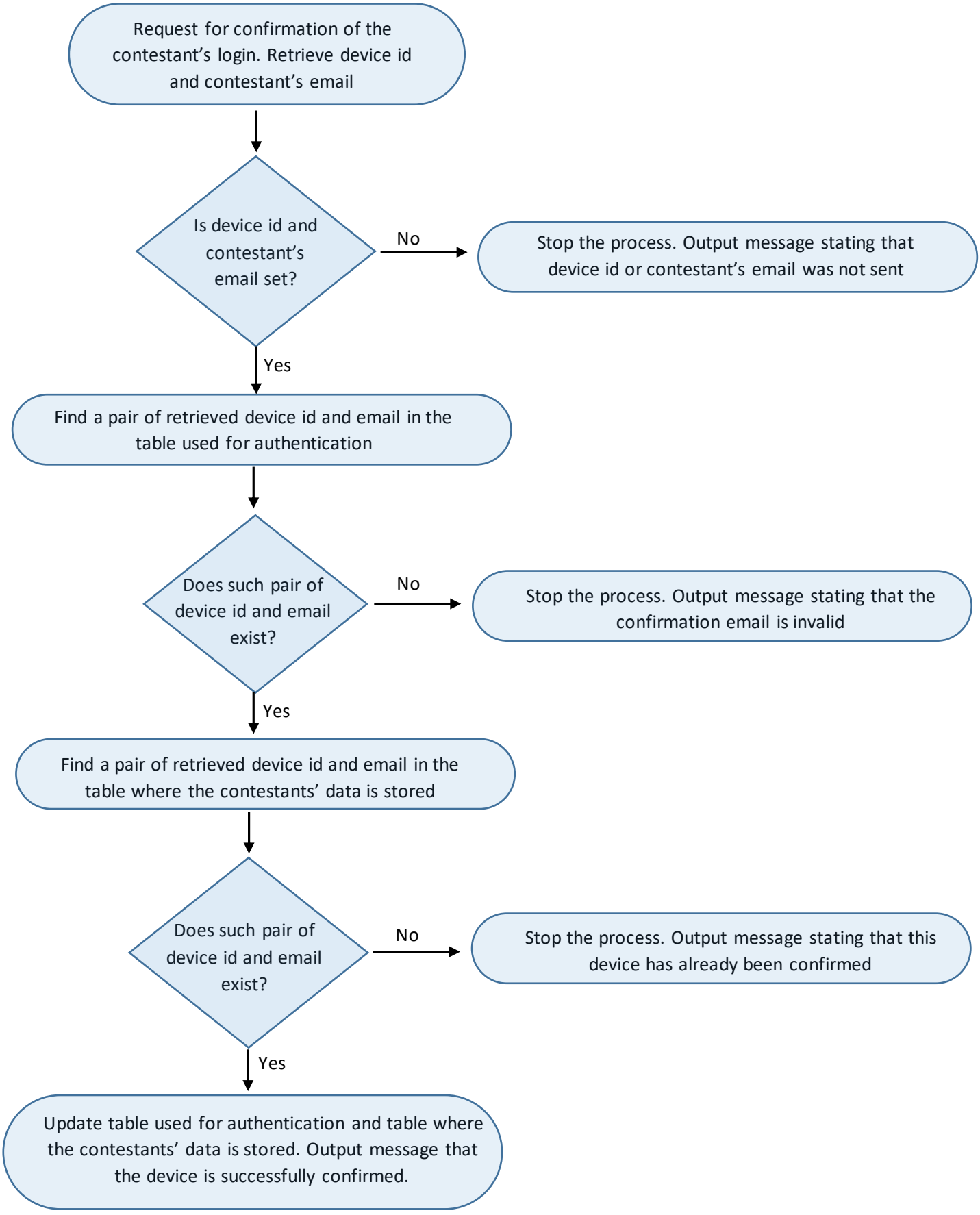

Figure 7. Program flowchart of a contestant's confirmation during the login process.

Tables

Table 'Login' stores username and password needed for the login process.

Login		
Name	**Data type**	**Description**
Username	Text	Username required for the login
Password	Text	Password required for the login

Table 1. Structure of login table

The 'Contacts' table is needed to store important information about the judges. It needs to be accessed by the administrator panel as the organizer may wish to add or change judges' information during the competition.

Contacts		
Name	**Data type**	**Description**
Name	Text	Name of the judge
Position	Text	Position of the judge
Phone_number	Text	Phone number of the judge
Email	Text	Email of the judge
Profile_picture	Text	Url address of the judge's profile pic

Table 2. Structure of contacts table.

The following table stores information about the debates and needs to be accessed by the administrator panel for appropriate update. This table does not communicate with the application.

Debate Schedule Unpublished		
Name	**Data type**	**Description**
Time	Text	Time when the debate takes place
Classroom	Text	Classroom in which the debate takes place
Team A	Text	Name of the first team
Team B	Text	Name of the second team
Judge 1	Text	Name of the first judge
Judge 2	Text	Name of the second judge
Judge 3	Text	Name of the third judge

Table 3. Structure of unpublished debate schedule table

The following table is the same as table 3, but this table is 'published'. Publishing in this context means that the application can communicate with the table. Administrator publishes the table when he is sure it is applicable for the following round and the contestants may see it.

Debate Schedule		
Name	**Data type**	**Description**
Time	Text	Time when the debate takes place
Classroom	Text	Classroom in which the debate takes place
Team A	Text	Name of the first team
Team B	Text	Name of the second team
Judge 1	Text	Name of the first judge
Judge 2	Text	Name of the second judge
Judge 3	Text	Name of the third judge

Table 4. Structure of debate schedule table

Table 'Schedule' stores information about the program during the conference. It needs to be accessible by the administrator panel in case there would be a sudden change in the program.

Schedule		
Name	**Data type**	**Description**
Date	Text	Stores the date of an event
Time	Text	Stores time of the event
Description	Text	Describes the event

Table 5. Structure of schedule table

Table 'Map' stores information needed for the communication with the application and also needs to be accessed by the administrator panel.

Map		
Name	**Data type**	**Description**
ID	Text	Stores contestant's device id
Name	Text	Stores contestant's name
Description	Text	
URL	Text	
Tags	Text	

Table 6. Structure of table for map

Table 'Notification' table stores all notifications and time of their update which were sent by the administrator.

Notification		
Name	**Data type**	**Description**
Time	Text	Time when the notification was uploaded
Description	Text	Message of the notification

Table 7. Structure of table for notifications

Table 'Pairs' stores data about the contestants' email and device ID and is used for the login process of the contestants.

Pairs		
Name	**Data type**	**Description**
Email	Text	Email of the contestant
ID	Text	ID of the contestant's device

Figure 8. Structure of table 'Pairs'

Table 'Authentication' has the same structure as table 'Pairs', but it is used for the process of confirmation. This is required for the contestant to login to the application.

Authentication		
Name	**Data type**	**Description**
Email	Text	Email of the contestant
ID	Text	ID of the contestant's device

Table 9. Structure of table 'Authentication'

Table 'Times' will be necessary in the communication with the application. It stores information about the time of the latest upload of each table that is accessible by the administrator panel.

Times		
Name	**Data type**	**Description**
File Name	Text	Contains 5 rows, specifying every file with appropriate name ('contacts', 'contestants'…)
Last Updated	Text	Specifies time of the latest update of the file

Table 10. Structure of table where time data of latest update is stored

Table ‘Contestants’ is a large table with an easy principle. It stores information about the contestants, their teams and their results in the debates throughout the whole competition. It needs to be accessible by the administrator panel as it will be updated after every round of the debates. Its structure is strictly given by the user.

Contestants		
Name	**Data type**	**Description**
Number	Text	Number of the contestant
Team	Text	Team in which the contestant is
Speaker	Text	Name of the contestant
Email	Text	Email address of the contestant
Win_Round_1	INT	Stores value ‘1’ in case the team has won, ‘0’ in case the team has lost
Ballots_Round_1	Text	Stores number of ballots the team was given from the judges in the first round
Points_Round1_Judge1	Text	Number of points the contestant was given from the first judge in the first round
Points_Round1_Judge2	Text	Number of points the contestant was given from the second judge in the first round
Points_Round1_Judge3	Text	Number of points the contestant was given from the third judge in the second round
Win_Round_2	Text	‘1’ in case the team has won, ‘0’ in case the team has lost
Ballots_Round_2	Text	Number of ballots the team was given from the judges in the first round
Points_Round2_Judge1	Text	Number of points the contestant was given from the first judge in the second round
Points_Round2_Judge2	Text	Number of points the contestant was given from the second judge in the second round

Points_Round2_Judge3	Text	Number of points the contestant was given from the third judge in the second round
Win_Round_3	Text	'1' in case the team has won, '0' in case the team has lost
Ballots_Round_3	Text	Number of ballots the team was given from the judges in the third round
Points_Round3_Judge1	Text	Number of points the contestant was given from the first judge in the third round
Points_Round3_Judge2	Text	Number of points the contestant was given from the second judge in the third round
Points_Round3_Judge3	Text	Number of points the contestant was given from the third judge in the third round
Win_Round_4	Text	'1' in case the team has won, '0' in case the team has lost
Ballots_Round_4	Text	Number of ballots the team was given from the judges in the fourth round
Points_Round4_Judge1	Text	Number of points the contestant was given from the first judge in the fourth round
Points_Round4_Judge2	Text	Number of points the contestant was given from the second judge in the fourth round
Points_Round4_Judge3	Text	Number of points the contestant was given from the third judge in the fourth round
Win_Round_5	Text	'1' in case the team has won, '0' in case the team has lost
Ballots_Round_5	Text	Number of ballots the team was given from the judges in the fifth round
Points_Round5_Judge1	Text	Number of points the contestant was given from the first judge in the fifth round

Points_Round5_Judge2	Text	Number of points the contestant was given from the second judge in the fifth round
Points_Round5_Judge3	Text	Number of points the contestant was given from the third judge in the fifth round
Win_Round_6	Text	'1' in case the team has won, '0' in case the team has lost
Ballots_Round_6	Text	Number of ballots the team was given from the judges in the sixth round
Points_Round6_Judge1	Text	Number of points the contestant was given from the first judge in the sixth round
Points_Round6_Judge2	Text	Number of points the contestant was given from the second judge in the sixth round
Points_Round6_Judge3	Text	Number of points the contestant was given from the third judge in the sixth round
Win overall	Text	Defines how many times the contestant's team won
Ballots overall	Text	Defines how many ballots the contestant's team received
Points team	Text	Defines how many points overall the contestant's team has received
Points individual	Text	Defines how many points overall the contestant received
Points average	Text	Defines the average number of points the contestant was given in one debate
Number of debated	Text	Defines in how many debates the contestant participated

Table 1. Structure of table where necessary data about the contestants and their results is stored

Proposed testing

Object of testing	Description of the test
Login into the administrator panel	The main menu page will be accessed before the login into the panel and therefore should not be successful. Afterwards, correct and incorrect login information will be used to test if the panel is password protected.
File upload	A chosen .csv file will be selected and uploaded to the database. Consequently, the table where the data should be stored will be controlled.
File download	A certain file from the options will be attempted to download.
Send notification	A message will be typed and the table where it should be afterwards stored will be controlled.
Publish the debate schedule	A button 'Publish' will be clicked and the database will be controlled whether the publish button transformed content between the two tables.
Automatic logout of the webpage after a certain time period of inactivity has elapsed.	The given time period will be set to a few seconds so that this criterion can be tested in the video.
Update time of the latest upload	Every time a table is uploaded in the database, in the main menu the time defining the latest upload of the table should be updated. This will be tested.

Table 2. Proposed testing

Approximate word count: 558