

## E-ACE: SEAT-BOOKING E-ASSISTANT FOR CULTURAL

Angelos Vardakis<sup>1</sup>, Evi Papaioannou<sup>2\*</sup>, Christos Kaklamanis<sup>2</sup>

<sup>1</sup>University of Patras, Greece, vardaki@ceid.upatras.gr

<sup>2</sup>University of Patras and CTI "Diophantus, Greece, papaioan@ceid.upatras.gr,  
kakl@ceid.upatras.gr

\*Corresponding author

### Abstract

We present e-ACE, an online e-assistant for seat booking in the context of cultural events (concerts, performances, festivals, etc). e-ACE is an online platform built on PHP, HTML, CSS, JavaScript, jQuery and MySQL. It is an open platform which can be accessed by registered users, i.e., cultural event organizers and participants/attendees. Cultural event organizers - or Managers - create events providing as input event details like event name, a plan of the hall where the event will be hosted, attendee list, start time and duration. In addition, a seat-editing functionality allows Managers to classify seats into categories like VIP, average, upper, unavailable, etc. Then, participants/attendees owing a valid ticket for a particular event can simply register using their ticket identifier and book their seat according to restrictions imposed by their ticket category. Results appear online but can also be exported and saved locally in .xls format. e-ACE follows a responsive web design which allows it to automatically resize, hide, shrink or enlarge in order to automatically fit on the screens of different devices (desktop computers, tablets, smartphones). e-ACE suggests a very useful management tool for entities (individuals, groups, institutes) organizing cultural events. It can significantly facilitate the event management process offering flexibility and efficiency. Furthermore, e-ACE could be also used for seat booking or allocation purposes in the context of scientific, social or educational events.

**Keywords:** smart service; online application; management of cultural events; online seat-booking; CentOS; hmtl5.

## 1. INTRODUCTION

During cultural events like concerts, performances, festivals, etc, the need for allocating seats in the event hall to participants is demanding process. Despite the fact that some sort of classification is made according to the particular type of purchased ticket (VIP, regular, discount, etc), the final allocation of seats remains a process mainly completed on-site. In cases of major, highly-populated events, the seating procedure usually turns out to be an effort- and time-consuming task, causing nervousness to the organizers and the involved audience.

There are several online applications and services for ticket purchase. However, existing approaches mainly focus on availability check and ticket purchase rather than online seat booking in particular halls . To the best of our knowledge, there is still lack of online services for seat booking in halls hosting cultural events.

In this work, we suggest e-ACE, an online e-assistant for facilitating the seat booking procedure for cultural events hosted in halls or sites involving seats. e-ACE is available online at <https://goo.gl/5UpNhF>. Exploiting the rapid advances in communication (i.e., Internet connectivity) and technology (smart devices widely available), e-ACE can make a very useful management tool for modern cultural institutes (Dix, Finlay, Abowd, Beale and Abowd, 2004). It can significantly facilitate the event management process offering flexibility and efficiency (Coulter, Robbins and Decenzo, 2014).

e-ACE can also become “an ace up the sleeves” of event participants who - most likely - have at least one modern smart device connected to the Internet. e-ACE features a responsive web design and can automatically adapt on different devices (i.e., desktop computers, tablets, smartphones) offering a unique experience and valuable help to its users.

The rest of the paper is structured as follows. The rest of the paper is structured as follows. In section “Technical description”, we provide a detailed technical description of e-ACE addressing design and implementation issues and approaches. In section “Demonstration”, we demonstrate e-ACE functionalities via comprehensive screenshots. We discuss future plans in section “Conclusions”.

## **2. TECHNICAL DESCRIPTION**

In this section, we provide technical details for e-ACE, which is an online platform built on PHP, HTML, CSS, JavaScript, jQuery and MySQL (Caya, 2018, Welling and Thomson, 2017). e-ACE was developed using the Linux-based CentOS distribution (<https://www.centos.org/>). Additional necessary software components include Web Server (e.g., Apache HTTP Server), Database (e.g., MySQL or MariaDB), PHP (version 7 or higher), Composer (latest version) and PhpSpreadsheet (<https://phpspreadsheet.readthedocs.io/>), which is usually installed via the Composer (Meloni and Kyrnin, 2018). Different technologies were used for the development of the platform which is essentially composed of 3 basic parts: Database, Server part, Client part. An orchestrated interaction of these parts is crucial for the correct function of e-ACE as well as for a successful user experience (Deitel, Deitel and Deitel, 2012). Below, we describe these parts in detail.

### **2.1. Database**

The DBMS used for e-ACE is MySQL. MariaDB makes a good alternative since the two systems are interchangeable. e-ACE database structure consists of 7 basic tables. Table “manager” contains managers and information like username, encrypted password, the IP of the most recent user connection, user full name. Table “room” contains a unique name for the event hall as well as the username of the manager who first stored this particular hall. Table “event” contains all data concerning the events. For each event, the following information is maintained: event title, manager in charge of the organization, event hall, start and end time, event status (active/inactive) as well as a ticket list specifying who can attend the event. Table “category” indicates different types of seats in a hall hosting an event; seats of different type can be selected and booked only by persons having an appropriate ticket (i.e., ticket purchased at a particular price). Categories can be added by managers. Table “ticketlist” contains all the unique Serial Numbers (S/N) of tickets registered for particular events as well as event title and the seat category. More specifically, there is a 1:N relation between the event title and the ticket S/N. Table “seat” contains all seats for all events and includes information about the category, reservation status (booked/available) and position in the hall via x, y coordinates. Table “aspect” is essentially the hall plan. This table provides the guide for the creation of elements in table “seat”. The entity-relationship diagram of the e-ACE database is depicted in Fig. 1. A green background indicates a primary key while arrows indicate dependencies.

Furthermore, the following conditions hold. Deletion of an event from the table “event” implies the deletion of all information affiliated to this particular event such as ticket records (“ticketlist” table), and seats related to that event. Also, deletion of a hall implies the deletion of its plan, seats and affiliated events.

### **2.1. Server side**

For the implementation of the server side part of e-ACE, object-oriented PHP was used since it offers flexibility, code reusability and efficient grouping of functionalities and data. The general communication model “Database <=> Server Memory <=> User Interface” has been adopted because it supports grouping of functionalities and data and also facilitates management and scalability of the platform. This scheme implies that data stored in the e-ACE database may also reside to the server memory. New entries as well as updates or deletions of existing entries related to halls, events and users take place both in the server memory (e.g., during an active user session) and in the system database so that such data is immediately available to users.

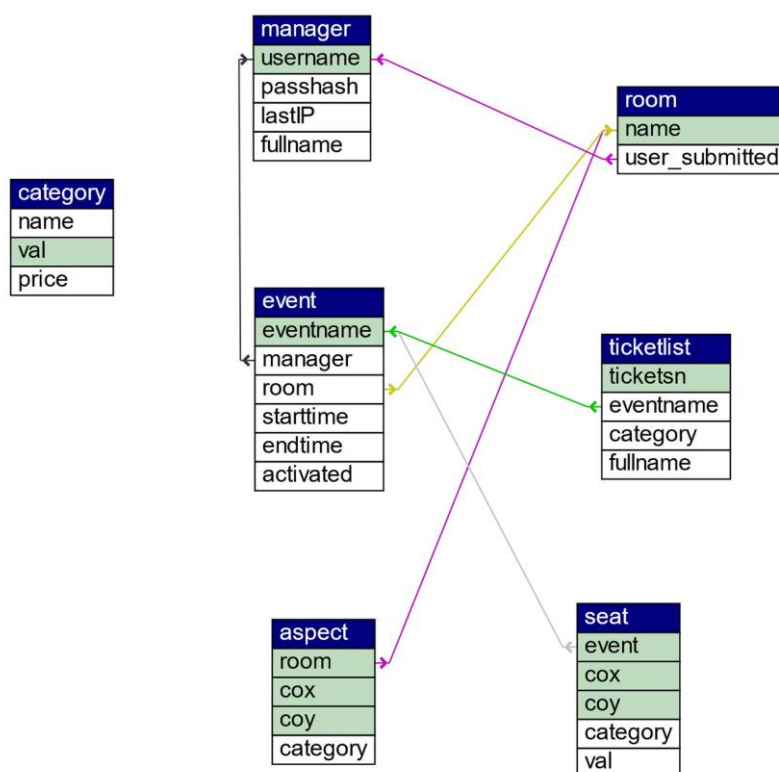


Fig. 1. The entity-relationship diagram of e-ACE database. A green background indicates a primary key while arrows indicate dependencies.

e-ACE is based on three prerequisites files: (a) core.php which contains the core code for e-ACE including classes, their members and methods, (b) consts.php where all constants are defined thus making e-ACE a highly parameterized system and (c) support.php which contains auxiliary functionalities frequently used within the system code. More precisely, core.php, which is actually the cornerstone of the application, has been built following a bottom-up approach. It contains 4 major classes namely, “seat”, “formation”, “user” and “attendee”. Objects of the class “seat” can represent either a position in an empty hall plan or a position in an actual hall plan for an upcoming event. The class “seat” has 4 members, namely coordinates x and y, category, and a value. The combination of the last two properties indicates whether a particular seat is reserved (value: 1), non-available (value: 0) or available (value > 0 and equal to a ticket S/N). The class “formation” is actually the basis where the event seating plan in a particular hall is built. It contains information regarding the event tile, hall name, event and hall seats, maximum number of seats horizontally and vertically for each hall plan. It contains 2 major subclasses, namely “event” and “room”. Objects of the class “user” represent a user who can currently be only a manager; however, the platform can be easily extended to accommodate also attendees as registered “users”. The class “user” currently contains 1 subclass named “manager”. In addition to user information like username, password, full name and the most recent IP by which the user connected to the system, a timestamp of the last log activity of the user is maintained (\$lastSeen) which is also used by session functions for security purposes. Objects of the class “attendee” represent attendees and contain information on ticket S/N, the event they are registered for, and a timestamp for log activity.

In addition to these three basic components, there are also several PHP scripts which handle user requests and respond to them. PHP scripts should also be considered important components of e-ACE since they play a significant role as intermediaries between the user interface and the server. PHP scripts of e-ACE perform tasks like initialization of the system database (1streg.php), listing and deletion of halls (delroom.php), login details (details.php), user management (keepalive.php, loginatd.php, loginmgr.php, logout.php, newusr.php), event hall plan creation and export (excelroom.php, generateexcel.php), event management for managers and attendees (getevents.php, new\_event.php), room management (newroom.php).

### *2.1.1 Data update and synchronization*

The activity of the user (whether active or not) is recorded at regular intervals. In addition, the activity of managers is updated after each request they make to the server. Active user logging is very important. It increases the security of the session as there is client-server communication and, therefore, agreement for the duration, renewal and termination of the session; furthermore, user data is continuously updated, since it is loaded directly the Database, thus providing up-to-date information on the data a user views and processes. Attendee data updates can take place less frequently (than after every request), so as to avoid mass queries to the database.

### *2.1.2 Platform security*

Claiming that a platform, application or website is generally sufficiently secure is extremely risky. Issues that need to be taken into account in order to fill all possible safety gaps are really numerous. However, the following tasks can significantly increase security. Server operating system and software certainly need to be updated as often as possible. Additionally, configuring the server to run applications on the HTTPS service with the necessary signed certificates certainly adds an additional level of protection.

Furthermore, it would be very useful to install a Web Application Firewall such as ModSecurity, which via appropriate rules can protect the system from malicious requests and attacks of various types.

User passwords are stored in the Database using the bcrypt encryption algorithm. The security it provides is satisfactory given the purpose of the platform. However, it is relatively easy to update to the Argon2 encryption algorithm (for PHP versions 7.2 or higher).

At the session level, appropriate functions have been defined to reduce the risk of session fixation / hijacking. More specifically, only one IP per session is allowed. So, in the case of a request made by a client connected to an already open session, the system deletes the session and logs out the client. Also, the id of a session is regenerated regularly so that session ID interception and reuse is avoided.

## **2.2 Client side**

The user interface forms the last part of the platform communication chain containing all data that is to be processed, displayed and stored. The server files accessible via hyperlinks include `admin_panel.php`, `events.php`, `first_reg.php`, `home.php`, `index.php`, `newevent.php`. At the beginning of most of these files there is a PHP script that checks whether the user is logged in and is a manager or an attendee before deciding whether or not to display the page. Also, all files include `support.js` that contains auxiliary functions. Additionally, `admin_panel.php` includes `adminpanel.js` where admin-specific utilities are implemented.

### *2.2.1 Layout Design*

The site has been built on Bootstrap 4 (<https://getbootstrap.com/>), providing a responsive design and user-friendly graphics environment. At the same time, several components of Bootstrap have been used, like for example alerts, buttons, dropdown menus, forms, input groups, list groups, modals, navbar, progress, tooltips.

Of course, HTML and CSS have been also extensively used. Custom items and parts were also created where appropriate. For instance, a CSS specifically formatted for cells was developed for the creation of seats. In this way, space on the user's screen is saved since table elements occupy the minimum possible space in relation to other elements.

### *2.2.2 Interaction via the user interface*

Users can modify the position of seats in an exam hall so that they meet their requirements. Furthermore, administrators can create exam halls from scratch and modify all of their parameters. Such functionalities require scripts to dynamically modify and add DOM document elements. For this reason, jQuery and JavaScript have been used in all pages accessed by users. jQuery - via the use of few lines of code - facilitates the definition of actions that must be performed if an event takes place. In general, it is a powerful tool for batch selection and modification of several elements for which sophisticated conditions have been set.

Additionally, 2 plugins have been added to facilitate data entry while providing a user-friendly visual result:

(i) jQuery Date and Time picker allowing users to select date and time to be automatically entered in the input field,

(ii) Bootstrap Toggle, which is actually a user-friendly checkbox with the graphics of an on-off switch.

### 2.2.3 Communication with the server

The interactivity of the platform is largely due to asynchronous communication via AJAX. In this way, users are not required to switch from page to page to complete an action or operation.

JSON has been selected to perform data send and receive due to simple syntax and structure. Each client request is sent either in JSON format or through the API FormData (when files must be uploaded together with data). In the latter case, feedback is provided on the progress of the file upload through the progress bar. The request method is always POST when referring to sending data to the server since it is safer than e.g., GET which sends data through the request URL. However, GET is used when just a request without data must be sent. For example, to regularly send an active session request to the user via the keepalive function specified in support.js, the method GET is used.

In general, all requests (other than those mentioned above) to the server are accompanied by a server response, augmented - when needed – by a success/failure message.

## 3 USING E-ACE

e-ACE is available online at <https://goo.gl/5UpNhF>. Below, we provide guidelines for the initial set up together with a comprehensive demo.

### 3.1 e-ACE initial setup

e-ACE initial setup can be performed via the Administration Panel where the following functionalities are offered: New Manager, View Managers, New Room, Delete Room, Manage Category. “New Manager” can be used for the registration of a new user of type Manager. Registered Managers can be viewed using “View Managers”. Using “New Room” allows the addition of an event hall so that it is available when a new event is created. Optionally, a spreadsheet file containing the hall structure and plan can be uploaded. An available seat corresponds to a cell with top, bottom, right and left border in the spreadsheet file. In any other case the cells are considered unavailable. Each newly created hall is named according to the corresponding spreadsheet name. Additionally, dimensions and seats can be modified. “Delete Room” can be used to view and delete saved halls. “Manage Category” allows the addition of a new seat category by entering name and ticket price as well as view and deletion of existing seat categories.

To initialize the platform, the first user must be registered. A user can be either a Manager or an attendee.

Users registered as Managers can manage existing events or create new events. Existing events appear as “Saved Events”. For existing events, Managers can view event details as well as the plan of the hall hosting each event. In particular, a map legend lists all available categories, each corresponding to a different color. Right clicking on a seat provides information for this seat. Event Status toggle button enables or disables the event for registration. Delete Event button deletes an event. For creating “New Events”, Managers must provide all required details by completing appropriate fields in a corresponding form. Event date and time must indicate a future time. Duration of an event must be at least 1 hour and at most 10 hours. Attendee list must be a spreadsheet file (xls, xlsx or ods) where data starts from the second row. In each row of such a spreadsheet, column B must contain the ticket Serial Number and column C must contain the ticket Category id. Columns A, D are optional. Column A indicates an index number and column D can contain the full name of the ticket holder. An attendee spreadsheet is depicted in Fig. 2.

	A	B	C	D	E
1	A/A	S/N	Category		
2		1	501	9	Lannie Schumaker
3		2	502	8	Jamal Whatley
4		3	503	7	Zulema Anspach
5		4	504	6	Luella Willingham
6		5	505	5	Vance Manry
7		6	506	4	Venetta Manges
8		7	507	3	Colette Hollmann
9		8	508	2	Cassi Siems
10					
11					
12					
13					
14					

Fig. 2. The structure of attendee spreadsheets.

Attendee users can view event details as long as they provide a valid ticket serial number (S/N). In particular, attendees can view on a map legend information about existing seat categories and the seat category of the current ticket (in bold). Right clicking on a seat notifies the user about details for this specific seat. Left clicking on a seat registers the current ticket to the selected seat (after a user confirmation is submitted).

During the setup of e-ACE the following issues must receive particular attention. Ticket S/Ns are unique. Attendee lists containing already existing ticket S/Ns should not be uploaded (in case this happens an error message shows up). Categories and halls can be used by all managers. Deleting a room or category may have consequences on scheduled events that use them. As far as categories are concerned, managers must provide ticket lists consistent with current existing categories. However, deleting a hall causes a chain effect, since all events scheduled for that hall are also deleted and, consequently, all tickets saved for this event are also deleted.

### 3.2 Demonstration

In this section, we provide a detailed description of several of the functionalities offered by e-ACE. When visiting e-ACE, users are prompted to log in to the system either as Managers or as attendees for Ticket Registration purposes (Fig. 3).

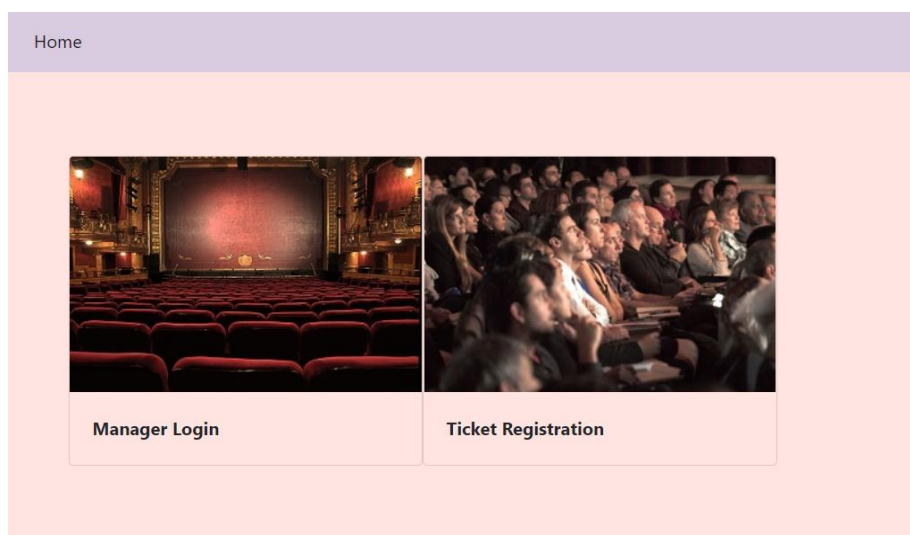


Fig. 3. e-ACE log-in screen.

A user wishing to connect as a manager is prompted to provide a Username and a Password. In case wrong credentials are provided, an appropriate message is returned.

Successful login is confirmed via a message appearing for 1.5 sec. Then, the user is automatically redirected to the e-ACE home page (Fig. 4) with links to “Saved Events”, “New Event”, “Admin Panel” and a “Log Out” option.

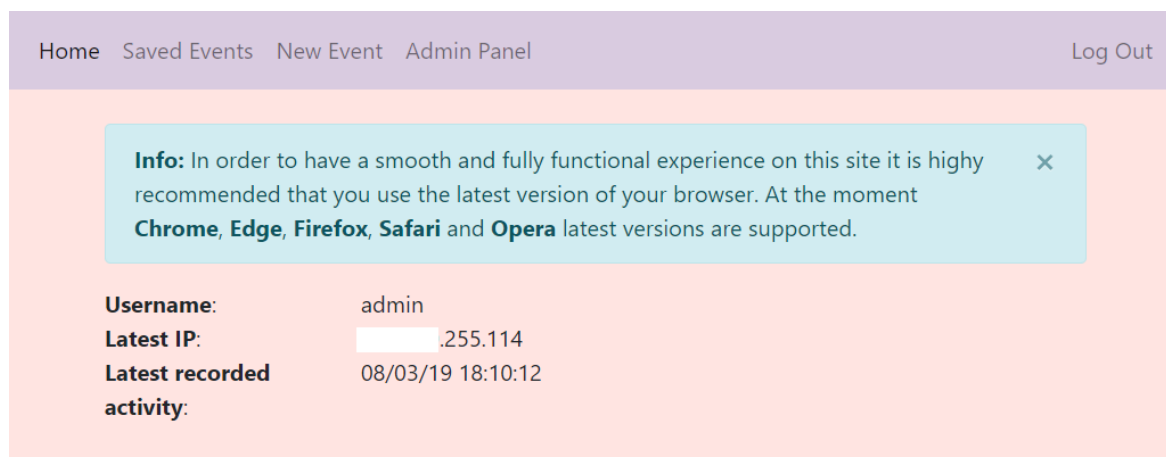


Fig. 4. e-ACE manager home page

A manager can organize an event by selecting the option “New Event” from the horizontal menu. For the creation of a “New Event” the following information is required: event name, hall (from an existing list), event start date and time, event duration and status (open/closed). In addition, the manager must upload an .xls file containing the ticket list (attendees’ ticket Serial Numbers and optionally full name). Furthermore, a manager can modify the status of seats in an event hall. Seat status is visually indicated via a legend. The Map Legend contains all categories added by any manager. Notice that the status of all seats can be reset (all available/non-available) by clicking on the # at the beginning of the table. Event start date and time can be chosen via a dropdown calendar.

Once all details are set on the corresponding form, clicking on the blue button “Save Event” (Fig. 5) completes the process.

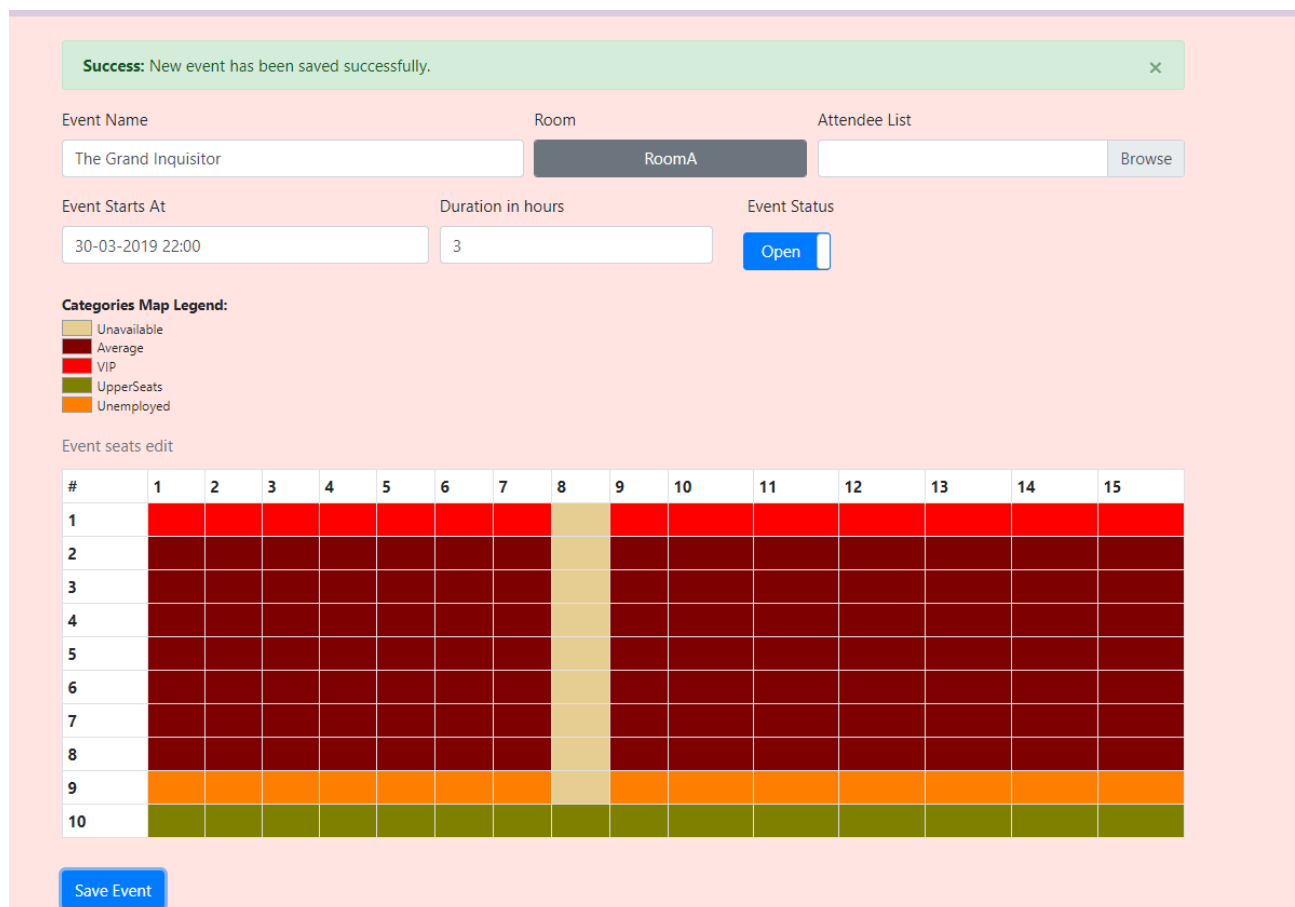


Fig. 5. Successful creation of a “New Event”

Via the option “Saved Events” in the horizontal menu, a manager can (i) retrieve or modify information about all scheduled events or delete events (Fig. 6), (ii) export event details including room seat allocation in an .xls file.

Also, by right-clicking on a cell, a manager can identify the ticket number for occupied seats.

The option “Admin Panel” allows a manager to perform administrative tasks like adding a new manager, adding and deleting categories, as well as adding a new room or removing an existing one (Fig. 7).

For the addition of a new hall, parameters like name, dimensions and availability of seats can be set. Furthermore, the hall plan can be directly imported as an .xls file.

A user wishing to use e-ACE to select a seat is prompted to provide a unique ticket Serial Number (see Fig. 8). Then, the event for which the attendee is registered appears.

For each available event, attendees can retrieve details like event title, manager, event hall, starting date, time and duration. Also, right clicking on a seat will provide any relevant information.

The screenshot shows the 'Management of event details' interface. At the top, there are navigation links: Home, Saved Events, New Event, Admin Panel, and Log Out. The event name 'The Grand Inquisitor' is highlighted in a blue box, with other options 'test event' and 'Some Shakspeare Play' visible. Below this, there are input fields for 'Manager' (Administrative Administrator) and 'Room' (RoomA). The 'Event Starts At' is set to 30-03-2019 22:00 and 'Event Ends At' is 31-03-2019 01:00. A 'Categories Map Legend' is provided with color-coded boxes: Unavailable (light yellow), Registered Seats (dark blue), Average (dark red), VIP (red), UpperSeats (green), and Unemployed (orange). An 'Event Status' dropdown is set to 'Open', and a 'Delete Event' button is present. The 'Course seats view' is shown as a grid with 10 rows and 15 columns. Row 1 contains various seat categories: red, dark red, dark red, dark red, dark red, dark red, dark red, light yellow, dark blue, red, red, red, red, red, red. Row 2 is dark red. Row 3 is dark red. Row 4 is dark red. Row 5 is dark red. Row 6 is dark red. Row 7 is dark red. Row 8 is dark red. Row 9 is orange. Row 10 is green.

Fig. 6. Management of event details

The screenshot shows the 'Available administrative tasks' menu. At the top, there are navigation links: Home, Saved Events, New Event, Admin Panel, and Log Out. The menu contains five teal buttons: 'New Manager', 'View Managers', 'New Room', 'Delete Room', and 'Manage Category'.

Fig. 7. Available administrative tasks





Fig. 8. Attendee login

Attendees can book any available seat matching their category, provided that they are not trying to register for more than once with the same ticket (Fig. 9). All information can be exported to an .xls file.

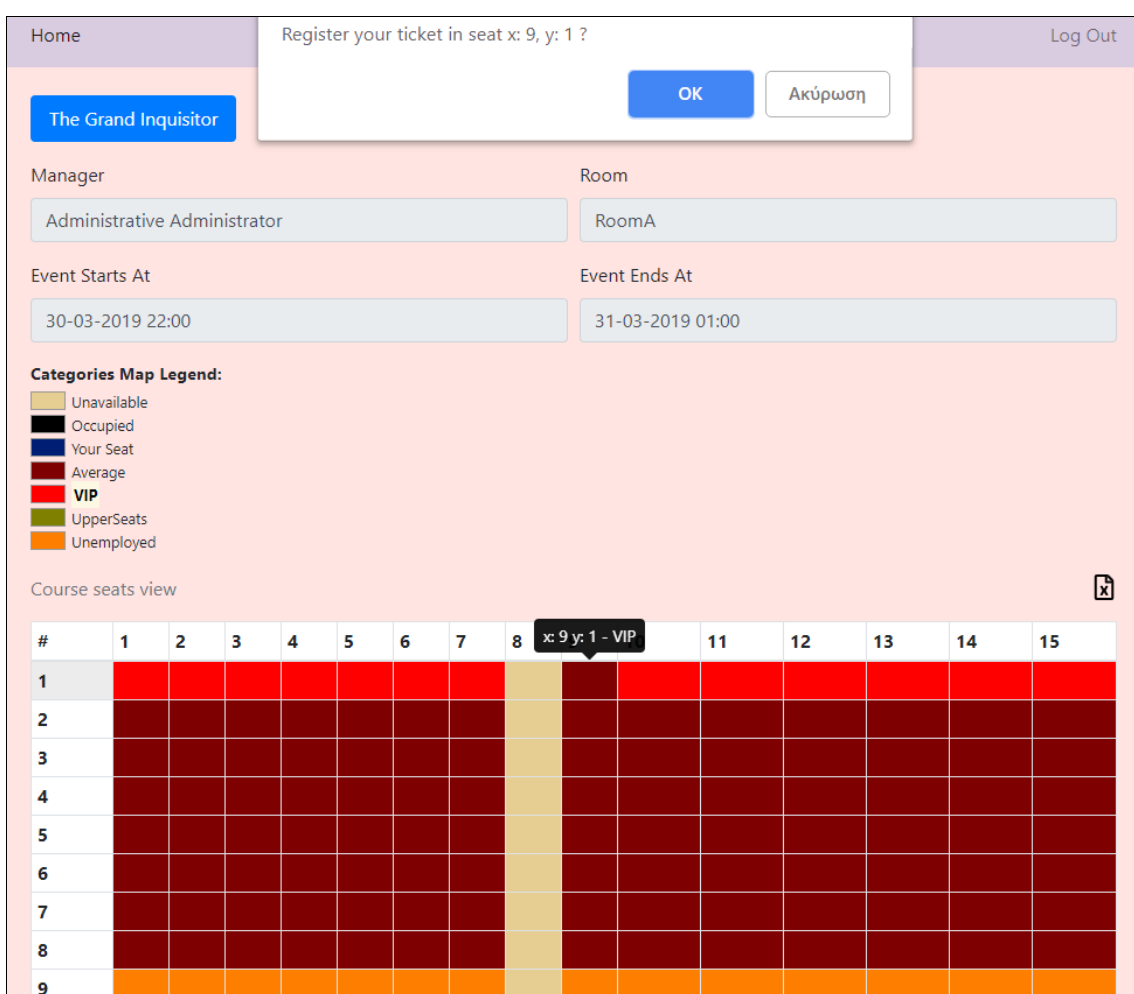


Fig. 9. Seat booking for a valid ticket

#### 4 CONCLUSIONS

Exploiting advances in internet and web technology and also the popularity and spread of smart devices, like smartphones and tablets, we designed and implemented e-ACE, an online platform for seat booking in halls hosting cultural events (concerts, performances, festivals, etc).

e-ACE suggests a very useful management tool for entities (individuals, groups, institutes) organizing cultural events. It can significantly facilitate the event management process offering flexibility and efficiency. e-ACE can also become “an ace up the sleeves” of event participants who - most likely - have at least one

modern smart device connected to the Internet. e-ACE features a responsive web design and can automatically adapt on different devices (i.e., desktop computers, tablets, smartphones) offering a unique experience and valuable help to its users.

It is within our future plans to further improve our platform via the incorporation of a barcode or a QR code reader for minimizing the required user actions for a ticket registration.

e-ACE, although initially intended for facilitating cultural event management process, could be also used for seat booking or allocation in other similar contexts like scientific, social or educational events.

## **REFERENCE LIST**

- Caya, A. (2018). *Mastering the Faster Web with PHP, MySQL and JavaScript: Develop state of the art Web applications using the latest Web technologies*. Packt Publishing Ltd, UK.
- Coulter, M., Robbins, S. P. and Decenzo, D. A. (2014). *Fundamentals of Management*. Pearson.
- Deitel, A., Deitel, P. J. and Deitel, H. M. (2012). *Internet & World Wide Web How to Program, 5<sup>th</sup> edition*. Pearson Education Limited.
- Dix, A., Finlay, J. E., Abowd, G. D., Beale, R. and Abowd, G. D. (2004). *Human-Computer Interaction*. Pearson Education Limited.
- Meloni, J. C. and Kyrnin, J. (2018). *HTML, CSS, and JavaScript All in One*. Sams Publishing.
- Thomson, L. and Welling, L. (2017). *PHP and MySQL Web Development (5th Edition)*. Addison-Wesley.
- CentOS Project. <https://www.centos.org/>
- PhpSpreadsheet's documentation. <https://phpspreadsheet.readthedocs.io/>
- Bootstrap. <https://getbootstrap.com/>