**HN University of Science and Technology**

**Software Engineering**

**Report for milestone 1 - Tinkerbell Garden v1.2**

**Team Name**: Team24

**Team ID:** 509

**Project Name**: Tinkerbell Garden

**Members**:

Vũ Đình Minh \_ 20194804

Nguyễn Thụ Hiếu \_ 20194761

Lê Đức Huy \_ 20194777

Mạc Đình Phú \_ 20194817

Nguyễn Khánh Linh \_ 20194789

Nguyễn Minh Tuấn \_ 20194876

Nguyễn Trường Giang \_ 20194751

Bùi Trần Hải Quân \_ 2019482

**Table of contents:**

1. Minimal Viable Product
2. List of functionalities
3. Flow of each functionality
4. Problems
5. Appendix
6. **Minimal viable product**

The product isn’t too specific but it does illustrate the essential requirements of the system including provisional design.

It clearly shows how the current version meets parts of the requirements. The report contains the list of functionalities that should be done, list of functionalities that have been done, the main or alternative flows, bugs and problems.

1. **List of functionalities**

Functionalities should be done in milestone 1 follow the feasibility report:

* Admin and staff can log in to the system and interact with the system through different pages based on their permission.
* Admin is the only one who can manage accounts, he can add more accounts for staff or delete accounts from the database or update the password of each account easily.
* Staff can sell tickets to create new tickets with Id for customers. They can also view the price of each ticket and pay for each ticket they want

Functionalities that have been done so far:

* The HomePage displays the basic information about the park and contains many references to another page as it is the introduction page.
* Admin can login to the system and edit accounts of its own or staff’s accounts. He can create a new account of staff or delete accounts of staff.
* Staff can login to the system and create new tickets for users. When customers get out of the park, staff can calculate the price of their ticket via the ticket ID and make a payment to save data on the database.
* Optionally, staff can save information of customers to the database when creating tickets for statistical purposes in the future.
* Other minor functionalities such as auto-login for staff and admin once they have logged in to the system before, the website is responsive and can be viewed normally on mobile or laptop,...

=> We have done every task in the first milestone. Some functionalities are not listed in the feasibility report because we add it during the development process to give the product better performance.

We have done the task very fast with 3 teams: design team, backend team, frontend team. Design and frontend are also testers at the end of each Scrum. We meet 3 times offline a week and work with the highest efficiency. Thanks to the leader of our team, who always manages the meeting and assigns tasks for each member in the right way.

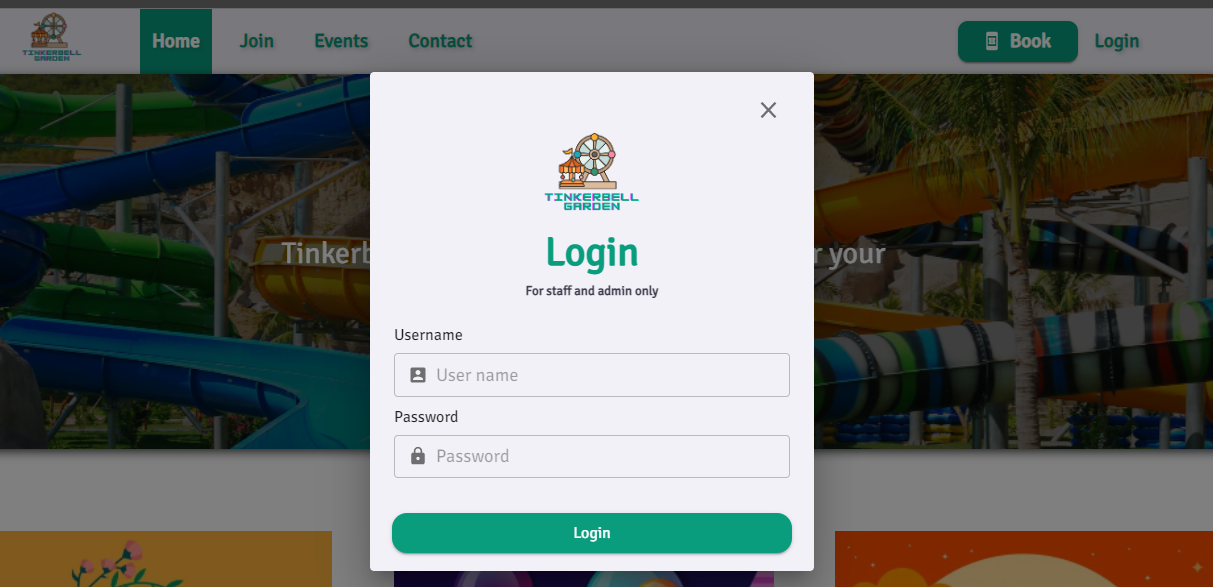
Starting from 8/5, in the first week, our team has finished the design for each page and unified everyone's opinion on database partitioning. In the next 1 week, the front end team divided into two parts, one for AccountManagementPage, one for PaymentManagementPage. These two tasks are complicated so we need to do them in parallel first. In the final week, we created a static web for HomePage which is very easy and tests all the functions, make sure it can be deployed for production.

=> The project v1.2 finishes totally in 3 weeks.

1. **Flow of each functionality**

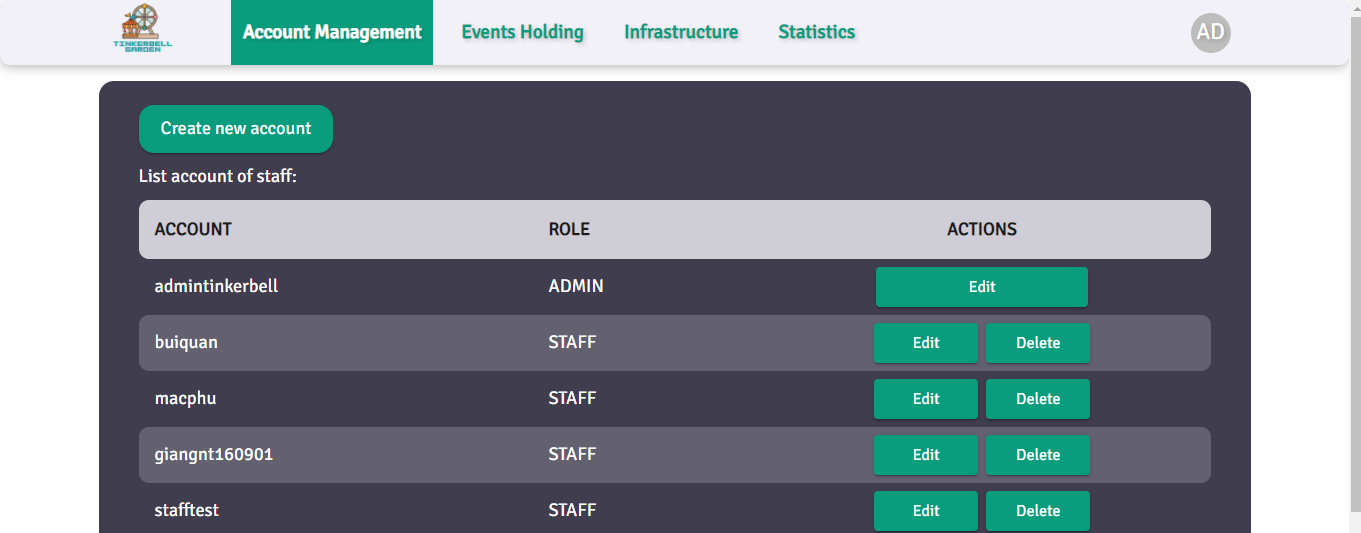
Flow of each functionalities:

1. Staff and Admin can login to the system by click to the login button on the to right of the website(this button is visible in every pages of the website), they can type their valid username and password and click button “Login”:

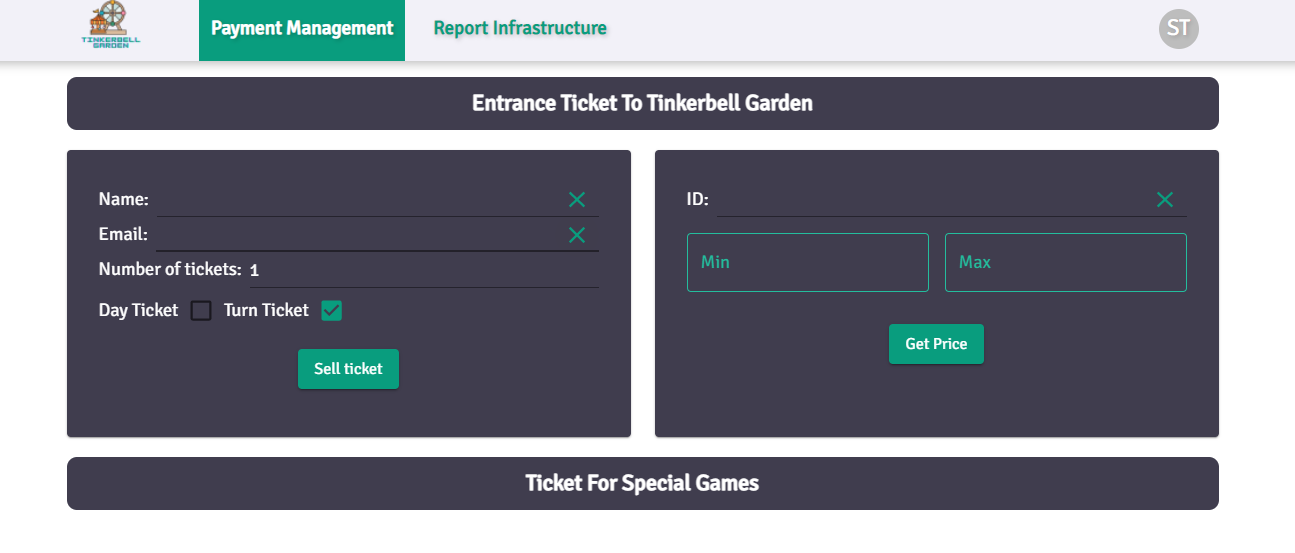


Depending on what their account role is, if login successfully, the page will be redirected to the appropriate router.

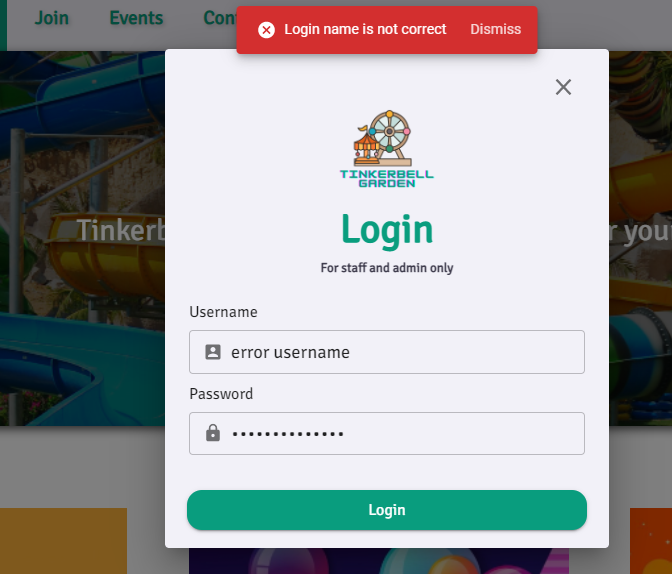
For admin:



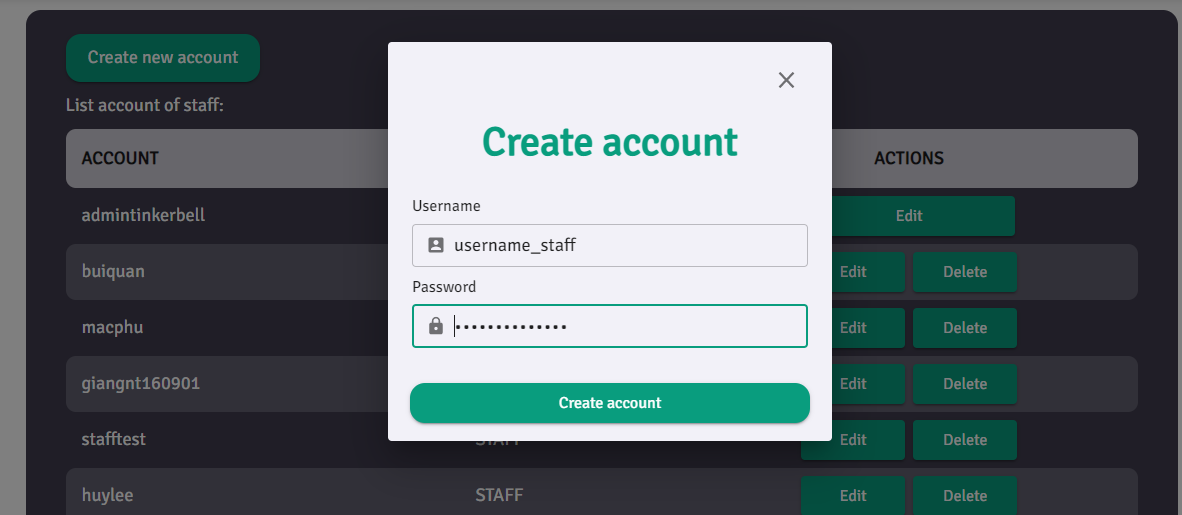
For staff:



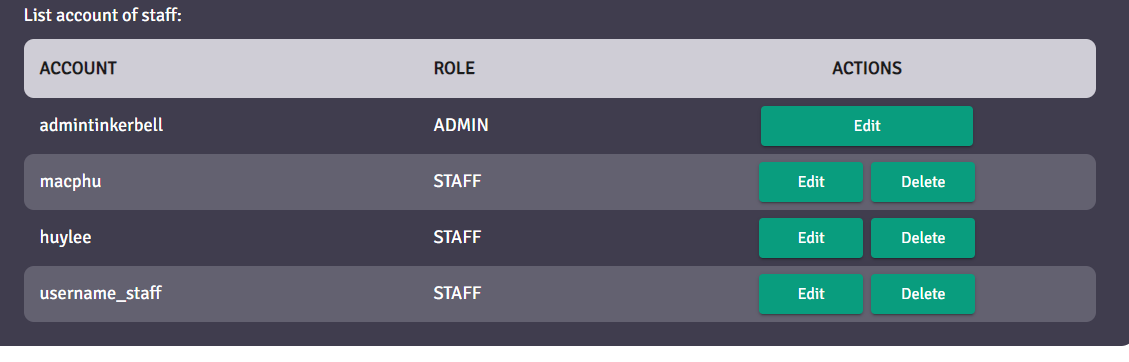
If there is a login error, the website will show the error message. Note that username contains at least 6 characters and password contains at least 8 character:



1. Admin can create accounts for staff by clicking the button “Create new account” on the Account Management Pages:

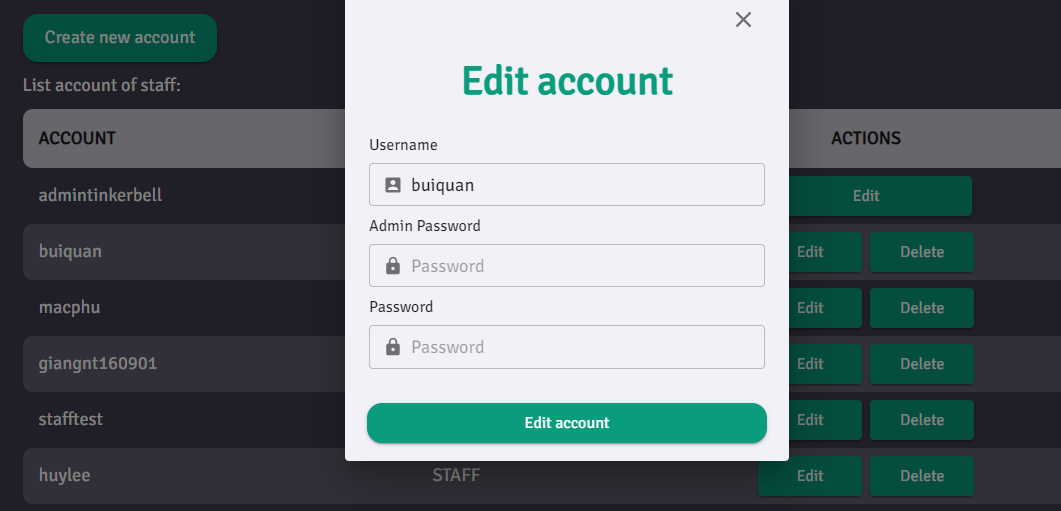


After admin created accounts, staff can login to the system with the new account. If the account fails to be created, the notification stack bar will show an error to the admin. If account is created successfully, it will be appear in the account list below:



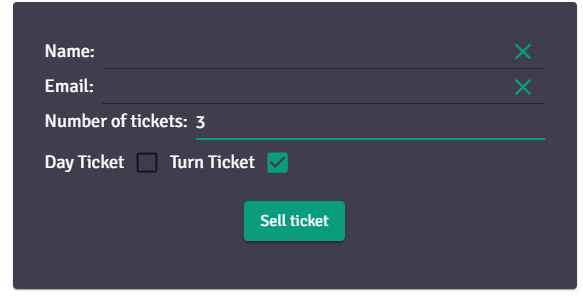
Admin can delete each account by click button delete in each row of the account or edit accounts of admin and other staff. After deleting, the account would disappear from the account list if deleting successfully and error will be shown if it fails.

To edit an account, the admin has to click on the edit button of each row of staff’s accounts and provide admin’s password for security:

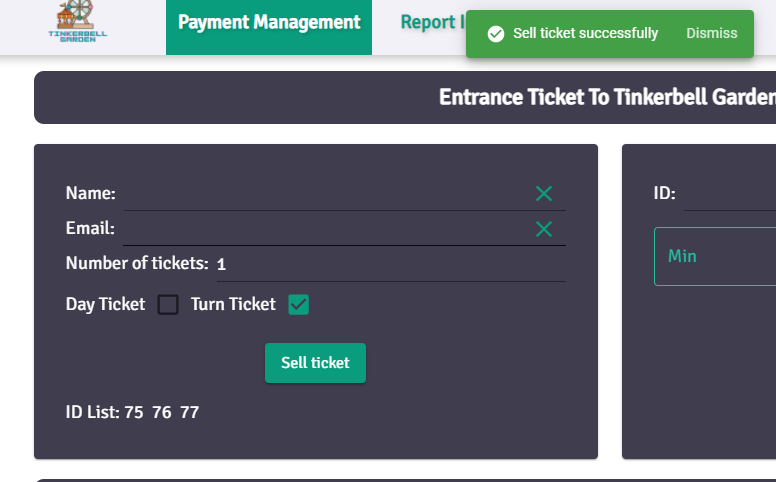


If updating accounts successfully, the system will pop up notification to users or throw an error if it fails. However, the admin can only edit the password of the account, not the username because changing both username and password is similar to delete and create a new user. After updating, staff can login to the system with the new password.

1. When customers come to the park and request to buy a ticket, staff will login to the system to create a new ticket on the “Payment Management” page. They have to input at least type of ticket and number of tickets:

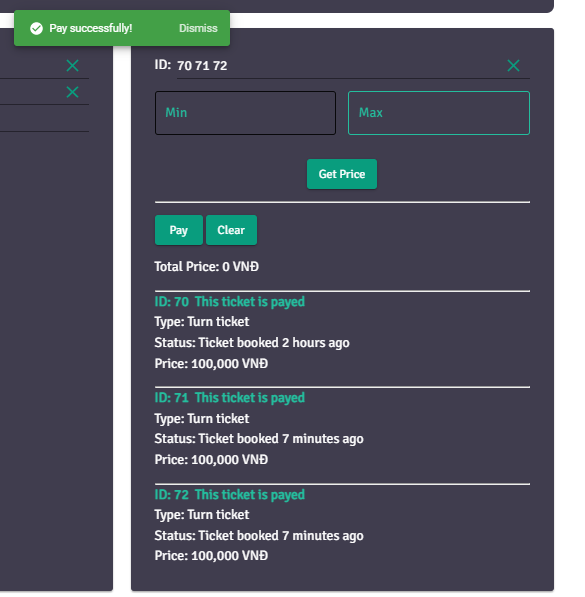


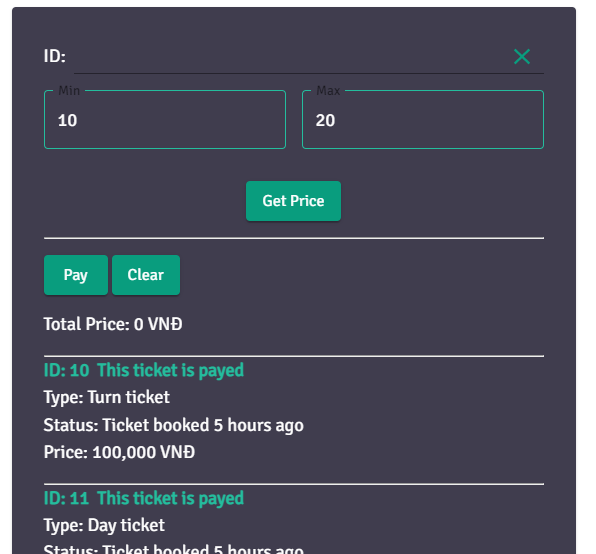
The field name and email of customers is optional for statistics purposes only. For example selling 3 turn ticket, if it has no error, the id list will be shown below the button immediately:



When customers want to go out of the park, they have to checkout and pay for the ticket. Depending on the type of ticket (day ticket or turn tickets) and the time they buy, the price is calculated automatically on server side and shows to the customer the total cost. Staff can input the id list of ticket and the detail information will appear immediately below the button:

If customers pay for those ticket, staff can simply click pay button to save these data to the database:



To make the payment easier for staff, we make the ID of the ticket auto increase by one. For example, if one person buys 10 tickets with id from 10 to 20, staff can input 10 and 20 instead of manually typing 10 ticket ids. 

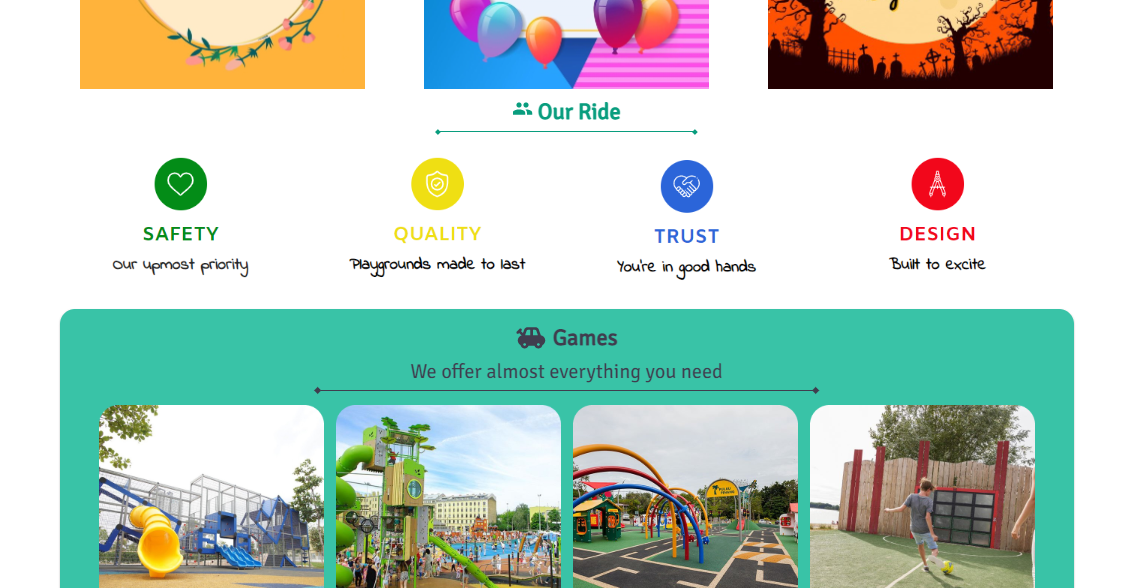
1. Other functionalities:

* All access pages are authorized. Admin cannot access staff’s pages and vice versa. HomePage can be access by both admin and staff and customer simply by click on the logo in the top left:

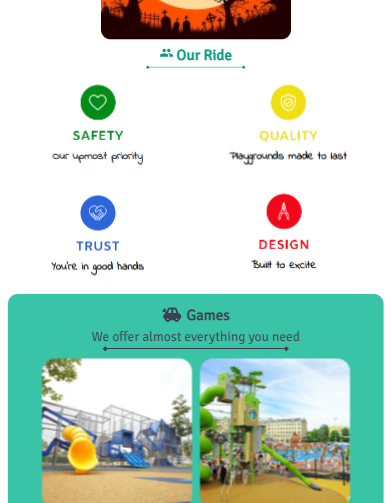
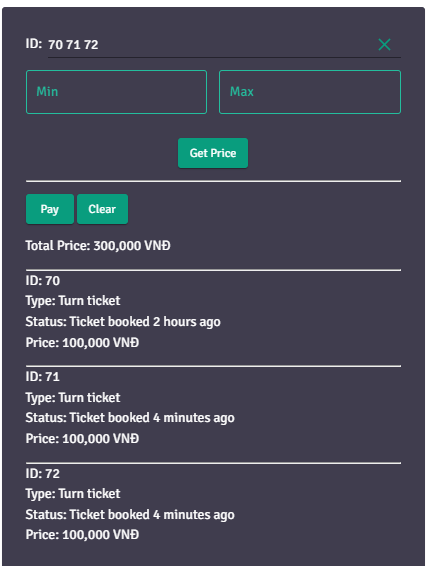


* The website is totally responsive and can be viewed on both mobile and laptop. For example:

The HomePage on laptop:



The HomePage on mobile:



1. We also do some functionalities such as manage report, manage event, manage games, sell ticket of paid games that you can see in more detail in the app and in the presentation
2. **Problems**

The project works pretty well. However, there are some minor problems we encountered:

* Press “Enter” on keyboard to login instead of click the button “Login”
* The color of input field is not correct follow the design
* Problems on fetching price of list id from client sending request to server.

=> Until now, our team has fixed all the bugs and there are no unresolved problems.

1. **Appendix**

Many **important documents** such as task division, overall use case diagram, database diagram, URL for website and github explanation is in the link below:

https://bit.ly/reportAttachment