**TicketVault**

**Sprint 2 Retrospective**

**Group: TicketVault (Team 8)**

**Members: Chengkang Xu, Hanxun Huang, Kai Sun, Longding Zhang, Yijie Wu**

**1.What went well?**

Every user stories is implemented during this sprint and each one of them is functional.

1) As a user, I would like to view the details of an event

We create a database named event to store all the events. There is an event detail page, which will query database using the event’s name, and displays all the details (event name, organization name, price, location ...etc.) of the event. By using SESSION variable, the event detail page will display “edit event” button if the logged in user is organizer of the event, “cancel” or “reserve”, and “add to cart” buttons if the user is registered user, or no buttons if the user is public user. By query the ticket database, if the login user is registered user who has booked that event, the “cancel” button will appear. If he / she has not booked that event, it will display “reserve ” and “add to cart” buttons.

2) As a student user, I would like to reserve tickets for the events that I like

We create a database table store every ticket that user booked and assigned a TicketID. In the table we also stored the eventID and username for specific TicketID. The username and eventID will be used to identify the owner of the tickets. If user is logged in as a normal user, there will be a reserve button appears on the event detail page for every event that user has not booked tickets for. Whenever user clicks on the reserve button, the user will be redirected to the confirmation page. If user clicks on submit, the remaining tickets of the event will be deducted by one, which is reflected on the database and the event detail page.

3) As a student user, I would like to manage the tickets that I have reserved

We added a cancel button to event detail page, which will only appear if the user has booked the event. In profile page, there is a “view booked events” button, which will redirect user to the booked event page when clicked. In the booked event page, there is a list of events that have been booked by the user. This was accomplished by query username in the ticket table in the database. If user clicks on the event’s title, he will be redirected to the event detail page. The “cancel” button will only appear in the event detail page if user has booked the event.

4) As an organization user, I would to create and edit event information

We created another user database table called organizationAccount. The organization user uses same login UI as student user does. If we did not found the account information in the student table, we will check with organizationAccount. If there are also did not found match it will return to the error page. If the organization user is logged in, we will store it in the session. The organization user can get access to the event information belong to them and edit it. They can also create a new event.

5) As a student user, I would like my data to be perfectly secured

We are using hash function in php to hash the user’s data during sign up. When the user try to login we use the same method to hash it and compare with the one in the database. The hash method will generate a unique value.

6) Sign up page improvement

By using SESSION variables, whenever user has typed any error information in the sign up text box, it will display the error that user formerly typed in. In addition, if input information contains any error, it will display error below the textbox. This was also accomplished by using SESSION.

7) Base page improve the reusability

We created a base page allow us easily change the header banner of each page and it is also contains the connection to the database called database\_connect() returns a $dbhandle allow us to connect to MYSQL database. We change the database to the current web hosting company and we can only change the connection information in the base.php.

8) Extra feature: Sign up email notification and lost password recovery.

After sign up the user will receive an email confirmation. If the user lost the password, the user can request change password. The user will receive an email contains a temporary link allow the user to change password without enter current password.

**2. What did not go well?**

There is a bug in the createevent.php that the function cannot assign an unique id to the event causing some related bugs.

**3.How should you improve?**

1) Currently there is no date information for the event, we should add it in next sprint and add some related service.

2) The current web hosting company did not gave us full access on the shell and did not allow us using customized page not found. If we can get full access, we should create a shell script that delete temporary reocvery lost password link periodically to increase security.

3) We can continue increase the reusability. For example we store the specific method to hash user’s data as a variable in global scope. When we want to change the encryption method we could simpily change the variable instead of change in every php file.