

Roller Coaster Tycoon First Iteration

Roller Coaster Team

Part I

<https://github.com/4156Team/4156TeamRepo>

Part II

MVP User Stories:

1. As a visitor, I can register my own account via my personal information and since then I can log in my account via my phone number and my password when I need to use this application.
2. As a visitor, I want to make my reservations and orders and observe them easily and modify them if I need so that I can book reservations and orders online and cancel them if I change my mind. My conditions of satisfaction are-
 - a. My ongoing orders and appointments should always have a striking display.
 - b. I can cancel my orders and appointments if I want.
 - c. All my orders and appointments will display according to the time.
 - d. Users can make appointments and orders, and they must be consistent.
 - e. If the server crashes and restarts, users should be able to observe their orders and appointments which have been made.
3. As a visitor currently in the park, I want to know the facilities' and events' current conditions so that I can make my in-park plan. My conditions of satisfaction are-
 - a. The time when the events are available.
 - b. Some description of events
 - c. Some introductions about facilities
 - d. The time when the facilities are available
 - e. How many people are in the queue now.
4. As a park manager, I want to have permission to change the status of events, add/delete events, check the current reservations for all visitors, and tickets' information so that I can manage the amusement park. My conditions of satisfaction are-
 - a. I can set an event to be open/close.
 - b. I can add/delete events.
 - c. I can modify the description of events.

- d. For a certain event, I know the number of appointments within a specific time slot.
- e. For a certain visitor, I know whether he/she has a ticket.
- f. For a certain visitor and event, I know whether he/she has a reservation for the event.

Acceptance test for MVP user stories:

1. As a visitor, I can register a new account when I first use this application. If I have an account, I will log in my account.
 - a. Given input of telephone number, name, gender, age, password, I can register an account. Otherwise, if one of the information is dismissed, the input will be invalid. For example, if I input an empty string of an telephone number, I cannot register an account successfully.
 - b. Once I have an account, I can log in my account via telephone number and password. Given an invalid input with an empty telephone number or empty password, I will fail to log in my account.
2. As a visitor, I want to make my reservations and orders and observe them easily and modify them if I need so that I can book reservations and orders online and cancel them if I change my mind.
 - a. Given the user account information, if the user can find their tickets and appointments, the test is passed and vice versa.
 - b. Given the user account information and the ticketId he wants to cancel, if the user can cancel the orders, the test is passed and vice versa.
 - c. Given the user account information and the id of appointment he wants to cancel, if the user can cancel the appointment, the test is passed and vice versa.
 - d. After a user makes an order or reservations, if there exists a related record in the database, the test is passed and vice versa.
 - e. Give the situation that a user wants to buy a ticket and if the ticket has been bought successfully, the test is passed and vice versa.
 - f. Give the situation that a user wants to reserve an appointment and if the appointment has been reserved successfully, the test is passed and vice versa.
 - g. Given the wrong ticketId which does not belong to the user, and if the user cannot cancel that order, the test is passed and vice versa.
3. As a visitor currently in the park, I want to know the facilities' and events' current conditions so that I can make my in-park plan.
 - a. Given an event name, if users can find the start and close time of the event, the test is passed and vice versa.
 - b. Given an event name, if users can find the description of the event, the test is passed and vice versa.

- c. Given a facility name, if users can find the introduction of the facility, the test is passed and vice versa.
 - d. Given a facility name, if users can find its open and close time, the test is passed and vice versa.
 - e. Given a facility name, if users can find how many people are in the queue, the test is passed and vice versa.
 - f. Given an event name which does not exist (invalid input), if the users can not see the event information, the test is passed and vice versa.
4. As a park manager, I want to have permission to change the status of events, add/delete events, check the current reservations for all visitors, and tickets' information so that I can manage the amusement park.
- a. Given an event name, if the manager can open and close the event, the test is passed and vice versa.
 - b. Given an event name and other information, if the manager can add and delete the event, the test is passed and vice versa.
 - c. Given an event name, if the manager can change the description of the event, the test is passed and vice versa.
 - d. Given an event name, if the manager can change the remaining number of appointments, the test is passed and vice versa.
 - e. Given a ticket id with valid date, if the manager can check that the ticket is valid the test is passed and vice versa.
 - f. Given a valid appointment id, if the manager can check that the appointment is valid the test is passed and vice versa.
 - g. Given a ticket id with invalid date, if the manager can check that the ticket is invalid the test is passed and vice versa.
 - h. Given a invalid appointment id, if the manager can check that the appointment is invalid the test is passed and vice versa.

Discussion of Acceptance Test:

In the acceptance test for user register and login, we found out that we need to design the database table carefully. In order to generate the unique userID while we can only get user information without an ID, we choose to set the user ID column auto increment to get the primary key. And when a user registers an account, the user must input all the important information such as telephone number and password which are used for login, so we must verify the inputs are not empty. Besides, in order to find the unique user, we need to set the column telephone number unique.

In the acceptance test for adding an appointment or ticket, we found that although there is a corresponding record generated in the database we could not know the corresponding appointment/ticket id which is automatically generated when adding a new appointment/ticket. That is we found a bug in design which makes us harder or cannot retrieve the appointment/ticket just created. Hence in current implementation, instead of simply return status messages we also return the automated generated appointment/ticket id back to the requester.

In the acceptance test for the manager part, after the check in ticket, I forget to change the state of ticket to state of used, therefore, I can check in the same ticket multiple times successfully. And thus I found the bug by this acceptance test and fixed it in my checkInTicket service.

Part III

For unit tests, we don't cover all the methods for some reason. Firstly, for those methods which are generated by mybatis framework, I think we don't need to test them in unit testing. Secondly, for the method of calling google verification api, we also don't test it. Thirdly, for some part of the code for debugging when we develop our application, such as TestController, we don't test them. Lastly, for those trivial methods such as getter and setter, we don't test them.

Part IV

We use Checkstyle plugin for style checker and spotbugs for finding bugs in our application. For both Spotbugs and checkStyle, I write some xml configuration to omit the scan for the files which are generated by the mybatis framework. And I use the google_checks for Checkstyle whose version is 6.18 and I modify its basicOffset of indentation to 4.

According to the Spotbugs.html and Checkstyle.html, we have fixed all the problems.