**Gigs N’ Chill**

**Software Test Plan**

**CSCI-P465/565 (Software Engineering I)**

**Project Team**

**Andrew Gotts**

**Deepan Elangovan**

**Jackie Drs**

**Harsha Valiveti**

# Overview

This section provides an overview of the testing approach used to verify the software product.

**1.1 Test Objectives**

Provide a description of the objectives of the test. That is, what is it that is to be accomplished by the testing effort.

The objectives of the test are to confirm functionality of each feature through basic functionality testing, unit testing, and problem reporting.

## Test Environment

Describe the environment that will be used to test the software product. Include a description of the hardware and operating system of all machines involved in the test. Describe the networking between those machines (if your product requires network support).

Describe any support software packages that need to be installed/available/running on the machine for the software to operate.

Comment on whether the test environment is the same as the operational environment and describe any differences.

We will be running the server on a local machine to test the GigsN’Chill website by starting it through console. We will also be using the API (Application Programming Interface) development tool, Postman, and REST APIs (Application Programming Interface) (Application Programming Interface) to send GET/POST HTTP request, simulating various specific testing environments, and checking console output to verify proper integrations. Each team member will unit test their assigned features.

## Test Personnel

Describe the personnel involved in the testing effort, and the roles that they will fill. Describe the level of user involvement in the testing process (if any).

All developers will be involved in the testing effort. The client will be involved in biweekly demos to verify that our features meet their requirements and are functional in various cases.

## Acceptance Criteria

Under what conditions will the software be considered acceptable to deliver? For example, how many errors will you tolerate before the testing is abandoned to correct the errors and try again? Or how many of the requirements need to be verified before the software is considered deliverable. Identify in this section an acceptance criterion that is feasible for your development team and acceptable for your customer/user.

Software will be considered acceptable to deliver if the feature is functional and allows the functionality of pre-existing GigsN’Chill software. If the error with the current software disables the rest of the software and cannot be completed before the deliverable due date, it will be noted and put in the backlog.

## Noted Omissions

Identify any software specification statements that are not intended to be verified by this test and justify the exclusion of these specifications.

# Test Cases

The test cases are the partitioning of the verification of the software into manageable sections. Often these sections correspond to the set of active use case scenarios but can be organized as the test developer sees fit. Test cases should be in place to cover all the software verification methods. Even non-execution-based testing methods (i.e., inspection/analysis) may be detailed here. The intent is for the test case procedures to provide a repeatable verification of the software specification.

For each test case describe the following:

**Number:** Provide a unique number for the Test Case

**Name:** Provide a unique, descriptive name for the Test Case.

**Description:** Provide a summary description (1-3 sentences) of the Test Case, and what is intended to be verified by the test case.

**Initial Conditions:** Describe any unique conditions in the test environment that must be in place to run the test. Also, note whether another Test Case is intended to be run prior to this Test Case.

**Input Data:**  Describe any specific data inputs that are to be used in the Test Case (e.g., a mock data file, a test user ID, erroroneous inputs to stimulate error conditions, etc).

**Specifications:** List the Specification statements (by number) that are intended to be verified by this Test Case.

**Procedure:**  Provide a link to the Test Case Procedure for this Test Case.

You may represent this information as a table, or simply list it.

**NOTE:** If you are using the rigorous process, the Specifications described above are already captured in the RVM and may be omitted from this document.

|  |  |
| --- | --- |
| Number | 01 |
| Name | Register |
| Description | To test the register, we entered user information (name, email, mobile and password) and checked SQL database to see if the data was saved. |
| Initial Conditions | To test sign up, the user must enter a valid email and valid password. |
| Input Data | First name (String), last name (String), email (String), mobile(int), password (String) |
| Specifications | Users will have a GigsN’Chill account |
| Procedure | Pass the required value in the JSON format which return the response as “SUCCESS” if the data is stored and “FAILURE” in case of errors. |

|  |  |
| --- | --- |
| Number | 02 |
| Name | Login |
| Description | To test log in, we proceeded to enter in the username and password previously submitted to register and verified if the user was able to log in. |
| Initial Conditions | Initial conditions are that the user must already be stored in the database. Test case 01 (Register) must be performed before logging in. |
| Input Data | Username (String), password (String) |
| Specifications | User has access to GigsN’Chill page |
| Procedure | Pass the required value in the JSON format which returns “USER LOGGED IN” when the username and password matches. In case of mismatch or user not available it returns “FAILURE.” |

|  |  |
| --- | --- |
| Number | 03 |
| Name | Login - OAuth |
| Description | To test log in, we proceeded to enter in the username and password previously submitted to register and verified if the user was able to log in. |
| Initial Conditions | To test OAuth Login, the user must have a valid google account. |
| Input Data | Username (String), password (String). |
| Specifications | Users should be able to login to GigsN’Chill page with Google Account |
| Procedure | When user clicks on “Continue with Google” on Login page, the application should be redirected to google login page and when the user authenticates the google account, the user should be able to login successfully. |

|  |  |
| --- | --- |
| Number | 04 |
| Name | Password Recovery |
| Description | To test Password Recovery, we proceeded to enter the email which was previously submitted to register and verified if the user was able to log in. |
| Initial Conditions | Initial conditions are that the user must already be stored in the database. Test case 01 (Register) and Test case 02 (Login) must be performed before Password Recovery. |
| Input Data | Email (String) |
| Specifications | User forgot the password to GigsN’Chill webpage |
| Procedure | Pass the required value in the JSON format which returns “PASSWORD RESET SUCCESSFUL” after the password is successfully updated to Database. In case of mismatch or email not available it returns “PASSWORD RESET FAILED.” |

|  |  |
| --- | --- |
| Number | 05 |
| Name | Profile View |
| Description | When the user is logged in, the user should be able to see his profile details, and booked events |
| Initial Conditions | Initial conditions are that the user must already be stored in the database. |
| Input Data | User/Host profile details along with history of events and upcoming evnets mapped in to the calendar. |
| Specifications | Retrieve user details from database and display in profile page |
| Procedure | Pass the required value in the JSON format which returns the user details from the database. |

|  |  |
| --- | --- |
| Number | 06 |
| Name | Search Events |
| Description | When the user is logged in, the user should be able to see the events which are available to book. And the user should also be able to search the events by filtering desired fields |
| Initial Conditions | Initial conditions are that the user must already be stored in the database. |
| Input Data | Search by name, category and type of event from the available list of events. |
| Specifications | Retrieve event details from database based on the fields provided for filtering. |
| Procedure | Pass the required value in the JSON format which returns the event details from the database. |

|  |  |
| --- | --- |
| Number | 07 |
| Name | Adding new events |
| Description | When the host is logged in, the host should be able to add the new events which he/she is going to host so that the registered users would be able to book the events. |
| Initial Conditions | Initial conditions are that the details of the host must already be stored in the database. |
| Input Data | EventName(String), EventGenre (String), EventType (String), EventDate(Date), EventStartTime(Time), EventEndTime(Time), Performer (String), MaxNoOfSeats(Int), SeatsAvailable(Int), Price (Int), HostId (String), City (String), State(String), ImageUrl (String) |
| Specifications | Host will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 08 |
| Name | Registering for the events |
| Description | When the user is logged in, the user should be able to register the future events. |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database. |
| Input Data | UserID(String), NoOfSeats(Int), EventID(String) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 09 |
| Name | Bookmarking the events |
| Description | When the user is logged in, the user should be able to bookmark the future events. |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database. |
| Input Data | UserID(String), EventID(String), BookmarkStatus(Boolean) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 10 |
| Name | Chat – One to one |
| Description | When the user is logged in, the user should be able to chat with other users and the host of the event |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database. |
| Input Data | UserID(String), RecipientID(String) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 11 |
| Name | User Review |
| Description | When the user is logged in, the user should be able to give review for the events |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database. |
| Input Data | UserID(String), EventID(Int), Rating(Int) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 12 |
| Name | Cancelling the registered events (User) |
| Description | When the user is logged in and previously registered for an event, the user should be able to cancel the registered events |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database and the user has previously registered the events. |
| Input Data | BookingId(Int) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

|  |  |
| --- | --- |
| Number | 13 |
| Name | Cancelling the hosted events (Host) |
| Description | When the host is logged in and previously hosted an event, the host should be able to cancel the hosted events |
| Initial Conditions | Initial conditions are that the details of the user must already be stored in the database and the host has previously hosted the events. |
| Input Data | EventId(Int) |
| Specifications | Users will have a GigsN’Chill account. |
| Procedure | Pass the required value in the JSON format which returns the response as “SUCCESS” if the data is stored and “FAILURE” with error details in case of errors. |

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Change Description** |
| Initial Creation | 19-Feb-2022 | Added Testcases for Register, Login, Password Reset |
| Added Testcases | 06-Mar-2022 | Added Testcases for Profile View and Searching the events |
| Added Testcases | 27-Mar-2022 | Added Testcases for adding new events and registering the events |
| Added Testcases | 10-Apr-2022 | Added Testcases for bookmarks, Inviting friends and chat |
| Added Testcases | 24-Apr-2022 | Added Testcases for user review, cancelling events and bookings |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |