# Report No. 5 System Implementation & Test

## Introduction

### Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of GBTS system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

### Test Approach

* Goal: Test all features in the whole GBTS system based on the core flow.
* Method: black-box testing
* Technique: check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

* Integration testing would be performed by all member of team and approved by team leader.

System testing is focused on assessing the system’s reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

## Database Relationship Diagram

### Physical Diagram



### Data Dictionary

Figure 62: <Database Relation Diagram> Physical Diagram

|  |  |  |
| --- | --- | --- |
| Data dictionary: describe content of all tables | | |
| Table Name | **Mapping column with Conceptual diagram** | **Description** |
| User | User | Contain the user information. |
| Role | N/A | Contain the role information. |
| BusRoute | BusRoute | Contain the bus route information |
| Card | Card | Contain the card information |
| Credit Plan | Credit Plan | Contain the credit plan which is used to add credit to card in the system. |
| Offer Subscription | Subscription | Contain the offer subscription information. |
| Payment Transaction | Payment Transaction | Contain the payment transaction information. |
| Promotion | Promotion | Contain the promotion information. |
| ScratchCard | ScratchCard | Contain the scratch card information. |
| Ticket | Ticket | Contain the bought ticket information for a card on a bus route or via cash. |
| Ticket Type | Ticket Type | Contain the type of ticket information |
| Token | N/A | Contain the server token information for verifying ticket bought by phone. |
| UserSubscription | Subscription | Contain the subscription information of user who subscribe an offer. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| BusRoute | ID{PK} | Unique identifier of a bus route | Integer | No |
| Code | Unique code for each bus route | String | No |
| Name | Name of a bus route | String | No |
| Card | Id{PK} | Unique identifier of a card | Integer | No |
| UniqueIdentifier | Manufacturer ID of a card | String | No |
| CardName | Name of a card | String | Yes |
| RegistrationDate | Registration date of a card | Date Time | No |
| Balance | Balance of a card | Float | No |
| Status | Status of a card | String | No |
| UserId | Unique identifier of an user | Integer | Yes |
| DataVersion | Data version of a card | Big integer | Yes |
| CreditPlan | Id{PK} | Unique identifier of a credit plan | Integer | No |
| Name | Name of a credit plan | String | No |
| Price | Price of a credit plan | Integer | No |
| Status | Status of a credit plan | Integer | No |
| PaymentTransaction | Id{PK} | Unique identifier of a payment transaction | Integer | No |
| CardId | Unique identifier of a card | Integer | No |
| CreditPlanId | Unique identifier of a credit plan | Integer | No |
| TransactionCode | Unique code of a transaction | String | No |
| Total | Total payment amount of a transaction | Integer | No |
| PaymentDate | Date and time of a transaction | Date Time | No |
| Promotion | Id{PK} | Unique identifier of a promotion | Integer | No |
| Name | Name of a promotion | String | No |
| Status | Status of a description | Integer | No |
| CreateDate | Created date of a promotion | Date Time | No |
| ExpireDate | Expired date of a promotion | Date Time | No |
| Role | Id{PK} | Unique identifier of a role | Integer | No |
| Name | Name of a role | String | No |
| ScratchCards | Id{PK} | Unique identifier of a scratch card | Integer | No |
| Code | Code of a scratch card | String | No |
| Price | Price of a scratch card | Integer | No |
| Status | Status of a scratch card | Integer | No |
| ExpiredDate | Expired date of a scratch card | Date Time | No |
| Subscription | Id | Unique identifier of a subscription | Integer | No |
| Name | Name of a subscription | String | No |
| Code | Code of a subscription | String | No |
| Price | Price of a subscription | Integer | No |
| DiscountPercent | Discount percent to ticket price of a subscription | Float | No |
| TicketNumber | Number of ticket can be use in a subscription | Integer | No |
| Status | Status of a subscription | Integer | No |
| Ticket | Id{PK} | Unique identifier of a ticket | Integer | No |
| CardId | Unique identifier of a card | Integer | No |
| BusRouteId | Unique identifier of a bus route | Integer | No |
| Total | Total payment amount for a ticket | Integer | No |
| BoughtDated | Date of ticket bought | Date Time | No |
| IsNoCard | Bought ticket using phone | Boolean | No |
| TicketType | Id{PK} | Unique identifier of a ticket | Integer | No |
| Name | Name of a ticket type | String | No |
| Description | Description of a ticket type | String | Yes |
| Price | Price of a ticket type | Integer | No |
| Status | Status of a ticket type | Integer | No |
| User | UserId{PK} | Unique identifier of a user | Integer | No |
| PhoneNumber | Phone number of a user’s account | String | No |
| Password | Password for a user’s account | String | No |
| Fullname | Fullname of a user | String | Yes |
| Status | Status of a user | Integer | No |
| RoleId | Unique identifier of a user’s role | Integer | No |
| DeviceToken | Device token of user’s device | String | Yes |
| MinBalance | Minimum balance to get notification | Integer | Yes |
| UserSubScription | Id | Unique identifier of a user subscription | Integer | No |
| UserId | Unique identifier of a user | Integer | No |
| SubscriptionId | Unique identifier of a subscription | Integer | No |
| ExpiredDate | Expired date of user subscription | Date Time | No |
| IsActive | Activity of user subscription | Boolean | No |
| TicketRemaining | Number of ticket remain of user subscription | Integer | No |

Table 73: Data dictionary: Describe content of all tables

## Performance Measures

### 3.1 Web Application Loading Speed

* + 1. Definition

This section tests the speed of loading some page of the web application.

* + 1. Test Environment
    - Server: Shared Windows Hosting on server:
      * Operating System: Windows Server 2012 R2
      * Network: 100 Mbps
    1. Test Cases

|  |  |  |
| --- | --- | --- |
| Test case | Role | Page |
| 1 | Unauthorize User | * Login |
| 2 | Passenger | * Get card list |
| 3 | Passenger | * Get outcome report |
| 4 | Staff | * Publish scratch card |
| 5 | Manager | * Get income report |

* + 1. Test Results

The test is run 100 times. Each time we run all the listed test cases and record the average time of web services responding.

|  |  |
| --- | --- |
| Test case | Average Loading Time (second) |
| 1 | 1.15 |
| 2 | 1.38 |
| 3 | 1.96 |
| 4 | 2.16 |
| 5 | 2.23 |

Test report: <https://github.com/dnhoang/GBTS/blob/master/Documents/Test_Report/GBTS_Performance_Test_Report.xlsx>

### Mobile API Speed

1. Definition

This section shows the testing speed results of some mobile application functions when connection to server through API

1. Test Environment
   * + Server: Shared Windows Hosting on server:
       - Operating System: Windows Server 2012 R2
       - Network: 100 Mbps
2. Test Cases

|  |  |  |
| --- | --- | --- |
| Test case | Role | Function |
| 1 | Unauthorize User | * Login |
| 2 | Passenger | * Get card list |
| 3 | Passenger | * Get outcome report |
| 4 | Passenger | * Topup card |

1. Test Results

The test is run 100 times. Each time we run all the listed test cases and record the average time of web services responding.

|  |  |
| --- | --- |
| Test case | Average Response Time (second) |
| 1 | 1.44 |
| 2 | 1.95 |
| 3 | 2.05 |
| 4 | 1.34 |

Test report: <https://github.com/dnhoang/GBTS/blob/master/Documents/Test_Report/GBTS_Performance_Test_Report.xlsx>

## Test Plans

### Features to be tested

* 1. **Manager**:
     + Add/update credit plans
     + Add/update ticket types
     + Add/update promotion packages
     + Add/delete promotions
  2. **Staff**:
     + Add/update passengers’ cards
     + Update passengers’ information
     + Publish scratch cards
  3. **Passenger**:
     + Top up card
     + Update card’s name
  4. **Admin**:
     + Insert account
     + Activate/deactivate account
  5. **Auto handler**:
     + Predict usage
     + Generate token
     + Crawl bus route data
     + Auto extend subcription
  6. Login, logout, update profile
  7. **Mobile application**:
     + Login, logout
     + Top up card
     + Add credit
     + Activate card
     + Update passenger’s profile
  8. **Emulator application**:
     + Verify ticket
     + Top up card
     + Publish card

### Features not to be tested

1. **Manager**:

* View income report

1. **Passenger**:

* View cards’ information
* Search bus routes by bus code
* Find buses by lacations

1. **Mobile** **application**

* View cards’ information
* View outcome report

## System Testing Test Case

### Communication Diagram

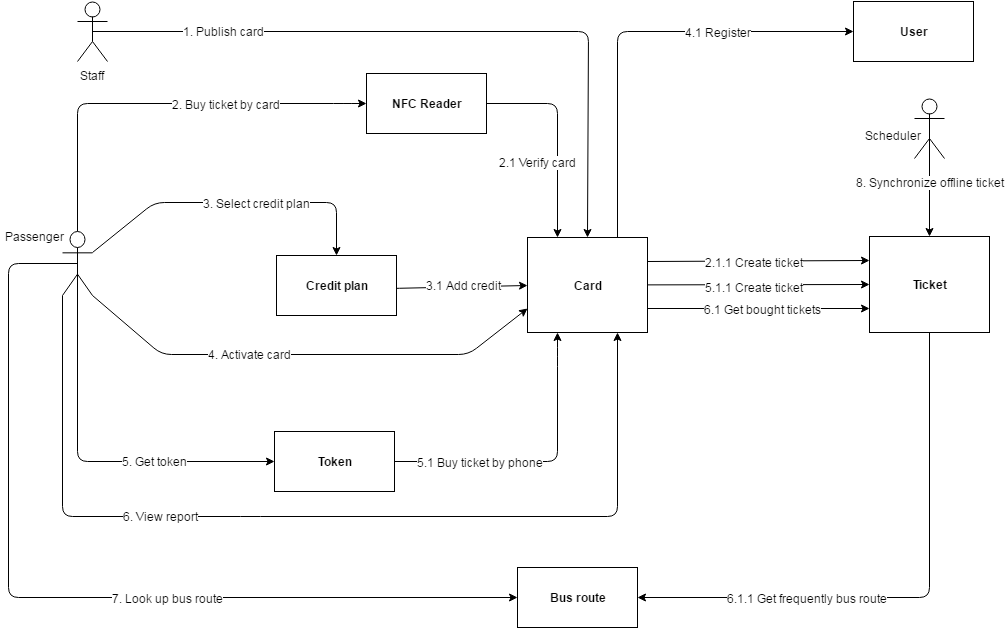


Figure 63: <Testcase> System Test



### System Test Cases

#### Web Application Test Cases

* + - 1. <Unauthorized User> Login

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| LG01 | Login with valid phone number and password | 1. Go to login page 2. Input valid phone and password 3. Press “Đăng nhập” button | Show target main page. |  | Pass | From 20/9/16  To 12/12/16 |
| LG02 | Login with invalid phone number or password | 1. Go to login page 2. Input invalid phone and password 3. Press “Đăng nhập” button | Show message “Sai số điện thoại hoặc mật khẩu” |  | Pass | From 20/9/16  To 12/12/16 |
| LG03 | Login with blocked account | 1. Go to login page 2. Input blocked account phone number and password 3. Press “Đăng nhập” button | Show message “Tài khoản đã bị khóa” |  | Pass | From 20/9/16  To 12/12/16 |

Table 75: <Unauthorized> Login

* + - 1. <Manager> Add card

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| AC01 | Add card successful | 1. Go to card page 2. Press “Thêm thẻ mới” button 3. Tap new card on the emulator 4. Input initial card balance 5. Press “Lưu” button | Updated card page with new promotion added |  | Pass | From 1/10/16  To 12/12/16 |

Table 76: <Manager> Add Card

* + - 1. <Staff> Activate/Deactivate card

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| AD01 | Activate card successful | 1. Go to card page 2. Press “Mở khóa thẻ” button | “Mở khóa thẻ” button change to “Khóa thẻ” button |  | Pass | From 1/10/16  To 12/12/16 |
| AC02 | Deactivate card successful | 1. Go to card page 2. Press “Mở khóa thẻ” button | “Khóa thẻ” button change to “Mở khóa thẻ” button |  | Pass | From 1/10/16  To 12/12/16 |

Table 77: <Staff> Add Card

* + - 1. <Passenger> Add credit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test case procedure | Expected output | Inter-test case dependence | Result | Test Date |
| CR01 | Add credit successfully | 1. Go to card page 2. Press “Nạp tiền” button 3. Select a credit plan 4. Input valid PayPal account 5. Press “Continue” button | Updated “Thẻ của bạn” panel with updated balance |  | Pass | From 25/9/16  To 12/12/16 |
| CR02 | Add credit using invalid PayPal account | 1. Go to card page 2. Press “Nạp tiền” button 3. Select a credit plan 4. Input invalid PayPal account 5. Press “Continue” button | Show error message “Check your email address and password and try again.” |  | Pass | From 25/9/16  To 12/12/16 |

Table 78: <Passenger> Add Credit

#### Mobile Application Test Cases

* + - 1. <Unauthorized User> Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **UUSI01** | Sign in with valid email and password | 1. Go to Log in screen.  2. Input fields:   * **Username: “**havh, anhnt, thiendn…” * **Password: “**123123, 123123, 123123…”   3. Touch “ĐĂNG NHẬP” button. | Transfer to borrower’s home page. | Pass | 1/4/2017 |
| **UUSI02** | Sign in with invalid email or password | 1. Go to Log in screen.  2. Input fields:   * **Username**: “scp,adfs, 23#@, …”. * **Password: “**123, abc, 1, 3,…**”**   3. Touch “ĐĂNG NHẬP” button. | Mobile application shows error message “Invalid user Id or password” | Pass | 1/4/2017 |
| **UUSI03** | Sign in with blank email or password | 1. Go to Log in page  2. Input fields:   * **Email**: “” * **Password: “**123456**”**   3. Touch “ĐĂNG NHẬP” button. | Mobile application shows error message “Invalid user Id or password”. | Pass | 1/4/2017 |

* + - 1. <Borrower> Check in with NFC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **BCIWNFC01** | Normal borrower scan NFC under emulator | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Transfer to In Library screen. | Pass | 1/4/2017 |
| **BCIWNFC02** | Deactivate borrower scan NFC under emulator | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Mobile show: “Your account is activated, please contact to Librarian”. | Pass | 1/4/2017 |
| **BCIWNFC03** | Borrower scan phone with outdated NFC code. | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Mobile show: “Invalid key”. | Pass | 1/4/2017 |

Table 80: <Passenger> Add Credit

* + - 1. <Borrower> Check in with QR Code

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **BCIWQRC01** | Normal borrower scan QR code under emulator | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Transfer to In Library screen. | Pass | 1/4/2017 |
| **BCIWQRC02** | Deactivate borrower scan QR code under emulator | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Mobile show: “Your account is activated, please contact to Librarian”. | Pass | 1/4/2017 |
| **BCIWQRC03** | Borrower scan phone with outdate QR code. | 1. Successfully logged in as Borrower role.  2. Go to Barcode screen.  3. Scan phone under emulator | Mobile show: “Invalid key”. | Pass | 1/4/2017 |

* + - 1. <Borrower> Scan book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **BSB01** | Borrower’s balance is smaller than the needed money to borrow book. | 1. Check in successfully.  2. Scan book. | Ring fail alarm. | Pass | 1/4/2017 |
| **BSB02** | Borrower scan book has been borrowed. | 1. Check in successfully.  2. Scan book. | Ring double ‘pip’. | Pass | 1/4/2017 |
| **BSB03** | Borrower scan available book and his balance is possible to borrow. | 1. Check in successfully.  2. Scan book. | Ring ‘pip’. | Pass | 1/4/2017 |
| **BSB04** | Borrower scan book does not belong to library. | 1. Check in successfully.  2. Scan book. | Do nothing. | Pass | 1/4/2017 |

Table 81: <Passenger> Buy ticket by phone

* + - 1. <Borrower> Check out

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **BCU1** | Go check out without carrying any books. | 1. Borrower does scan any book.  2. Go out to check out. | Phone show message: “Check out successfully”. | Pass | 1/4/2017 |
| **BCU2** | Go check out with available book. | 1. Borrower scan book successfully.  2. Go out to check out. | Phone notify message contain books have been borrowed. | Pass | 1/4/2017 |
| **BCU3** | Go check out with available book and open app. | 1. Borrower scan book successfully.  2. Go out to check out. | Phone notify message contain books have been borrowed. | Pass | 1/4/2017 |
| **BCU4** | Go check out with available book and app run in background. | 1. Borrower scan book successfully.  2. Go out to check out. | Phone notify message contain books have been borrowed. | Pass | 1/4/2017 |
| **BCU5** | Go check out with available book and turn off phone. | 1. Borrower scan book successfully.  2. Go out to check out. | Phone notify message contain books have been borrowed. | Pass | 1/4/2017 |