

WIAPP MODIFICATIONS
Design Document

Version 1.0
30/03/2022

Document Owner: Brahmini Boina, Software Developer Trainee

Creation Date: 30/03/2022

Change Record:

Date	Author	Change Summary
30/03/2022	Brahmini Boina	Enhanced version

Project	WiApp
Redmine Ticket	Feature #159
Task	WiApp Enhancements – Check the billing flow and make enhancements.
Assignee	Brahmini Boina
Assigned By	Nagaraju
Assigned Date	08/03/2022
Estimated Hours for Completion	16
Actual Hours taken	-
Reviewed By	Nagaraju
Review Date	--/--/2022
Deploy Date	--/--/2022

Table of Contents:

Overview:	05
System Components.....	06
Component 1 – Java /JSP Changes:.....	07
Component 2 – DB Changes:.....	14
Component 3 – Environment/ Configuration:.....	14
Component 4 – Test Cases and Results:.....	15

Overview:

Check the billing flow individually for both MSISDN and Google sign in. Make required enhancements in the code corresponding to billing. Here billing flow must work correctly based on the cookies, msisdn and Google sign in.

When the user login with mail ID, try to access the content, the page will be redirected to billing page. After buying the content, it again redirects to home page. Then the user can access the content. Same flow goes for the user when he enables msisdn. Here made some changes in the code to make additional enhancements.

General flow:

<https://dev.wicore.in/wiapp> → user clicks on content → displays alert message [to enable msisdn or Google sign in] → after sign in or enabling msisdn [data] → clicks on content → redirects to billing page [shows subscription types] → clicks on payment mode → purchase content → redirects to home page → clicks on content → access the content.

System Components

- Component 1 – Java/ JSP changes
- Component 2 – DB Changes
- Component 3 – Environment/Script changes
- Component 4 – Test cases and results along with screen shots

Component 1 – Java/JSP Changes:

Java Changes in WiAPP:

1. In `google_signin.js` file
[`/data/wiapp/WiappElements/login/LoginDefaultStyle/js`], written a new method `submitPostWithGoogleToken(idToken)` which passes the google tokenID to the `Index.java` class using `form post method`. Here this method is called in `onFirstSuccessSignIn()` method, when the user logs in.
2. In `Index.java` we removed all the unnecessary code and called a method `manageCookies()` by passing the parameters request, response, customerBean, tokenID, tokenIDresponse.
3. Here we validate the token using `validateToken()` method by passing this tokenID with Google api. After validating we pass this token response to this method. In `manageCookies()` method which is in `CookieUtility.java` class, we get the details like gmail, name of the user, locale using this token response.
4. After that based on the msisdn and gmailID conditions, we create cookies when the user signs in or enable msisdn.
5. When the user doesn't sign in or disables the msisdn, and when he clicks on content, we display alert message to either sign in or enable mobile data. So, inorder to do that, made some changes in the `contentUtility.java` class, created a `userObject` which stores the msisdn and gmail id of the user and passes this to `RenderHtml.java` class.
6. In this java class, we will call a method `addFormPost(headEle, doc)` and provided the implementation of this method in `HTMLUtility.java` class. Here by using form tag and post method, we will send the mail and gmail using `redirect method` by passing these parameters to the `custom.js` file [`/data/wiapp/WiappElements/content/CardVerticalContent/js/`]. Here in `redirect method`, written a code of displaying **alert message** when the user doesn't sign in or enables mobile data[msisdn].

Code Snippet in HTMLUtility.java class:

[wi frame image.setAttribute\("onclick", "redirect\('"+item link+"', '"+msisdn+"', '"+gmail+"'\)"\);](#)

Code Snippet in custom.js file:

```
function redirect(url, msisdn, gmail) {  
    if (msisdn == "" && gmail == ""){  
        alert ("please login with Google or enable mobile data.");  
        return false;  
    }  
}
```

Code written to check the various user sign in and msisdn conditions:

For MSISDN:

```
/*    if msisdn exists and cookies exists  
    if msisdn exists and cookies not exists. */  
  
if(!msisdn.equals("")) {    //MSISDN exists  
    cookieID = getCookieidFromBrowser(request, log_info);  
  
    if (!cookieID.equals("") && !cookieID.equals("NA")) { //cookie  
exists.  
        userID = CookieUtility.getUseridFromCookieId(cookieID, "");  
    } else { //cookie doesn't exists.  
        userID = CookieUtility.getUserID(con, msisdn);  
        cookieID = CookieUtility.getCookieExistsWithUserID(con,  
userID, log_info);  
  
        if (!cookieID.equals("") && !cookieID.equals("NA")) {  
//Cookie exists with user id.  
            CookieUtility.createBrowserCookieWithExistingCookie(re  
sponse, cookieID, log_info);  
        } else { /////Cookie exists not exists with user id we need  
to create a new cookie (Fresh request)  
            boolean isCreateCookie = true;  
            cookieID = CookieUtility.createBroswerCookie(request,  
response, log_info, isCreateCookie);  
  
            if (!cookieID.equals("")) {  
                CookieUtility.storeCookieINFO(customerBean,  
cookieID, log_info);  
                userID = customerBean.getCookieUserID();
```



```

    }

    TokenInfoDAO.msisdnUserInfo(userID,
    customerBean.getMobileBean().getMobile_no(),
    customerBean.getCust_id(),
    customerBean.getCustomerServiceBean().getService_id(),
    customerBean.getMobileBean().getOp_id() );
    }
    }
    Log.info (log_info+" MSISDN userID: " +userID+" | MSISDN
cookieID: " +cookieID);

} else { //MSISDN doesn't exists
    cookieID = getCookieidFromBrowser(request, log_info);

    if (!cookieID.equals("") && !cookieID.equals("NA")) { //cookie
exists.
        userID = CookieUtility.getUserIdFromCookieId(cookieID, "");
    } else { //cookie doesn't exists.
        //here we will do nothing.
    }
    Log.info (log_info+" MSISDN userID: " +userID+" | MSISDN
cookieID: " +cookieID);
}

```

For Gmail:

```

/*    if gmail exists and cookies exists
    if gmail exists and cookies not exists. */

if(!gmail.equals("")) { // if gmail exists
    cookieID = getCookieidFromBrowser(request, log_info);

    if (!cookieID.equals("") && !cookieID.equals("NA")) { //cookie
exists.
        userID = CookieUtility.getUserIdFromCookieId(cookieID, "");
        Log.info("if no cookieID: " +cookieID);
    } else { //cookie doesn't exists.
        userID = CookieUtility.getGmailUserID(gmail);
        cookieID = CookieUtility.getCookieExistsWithUserID(con,
userID, log_info);
    }
}

```

```

        if (!cookieID.equals("") && !cookieID.equals("NA")) {
//Cookie exists with user id.
            CookieUtility.createBrowserCookieWithExistingCookie (
response, cookieID, log_info);
        } else { ////Cookie exists not exists with user id we need
to create a new cookie
            boolean isCreateCookie = true;
            cookieID = CookieUtility.createBroswerCookie(request,
response, log_info, isCreateCookie);
            /** Here storing to cookie info and user id info is not
required as we are doing this in Validate Auth API**/

            if (!cookieID.equals("")) {
                CookieUtility.storeCookieINFO (customerBean,
cookieID, log_info);
            }

            if(userID.equals("")) {
                userID = customerBean.getCookieUserID();
            }
            TokenInfoDAO.googleUserInfo(con, userID, gmail,
given_name, family_name, locale, tokenID);
            Log.info("TokenInfoDAO " +cookieID);
        }
    }

    Log.info(log_info+" GMAIL userID: " +userID+" | GMAIL cookieID: "
+cookieID);

} else { //gmail doesn't exists
    cookieID = getCookieidFromBrowser(request, log_info);

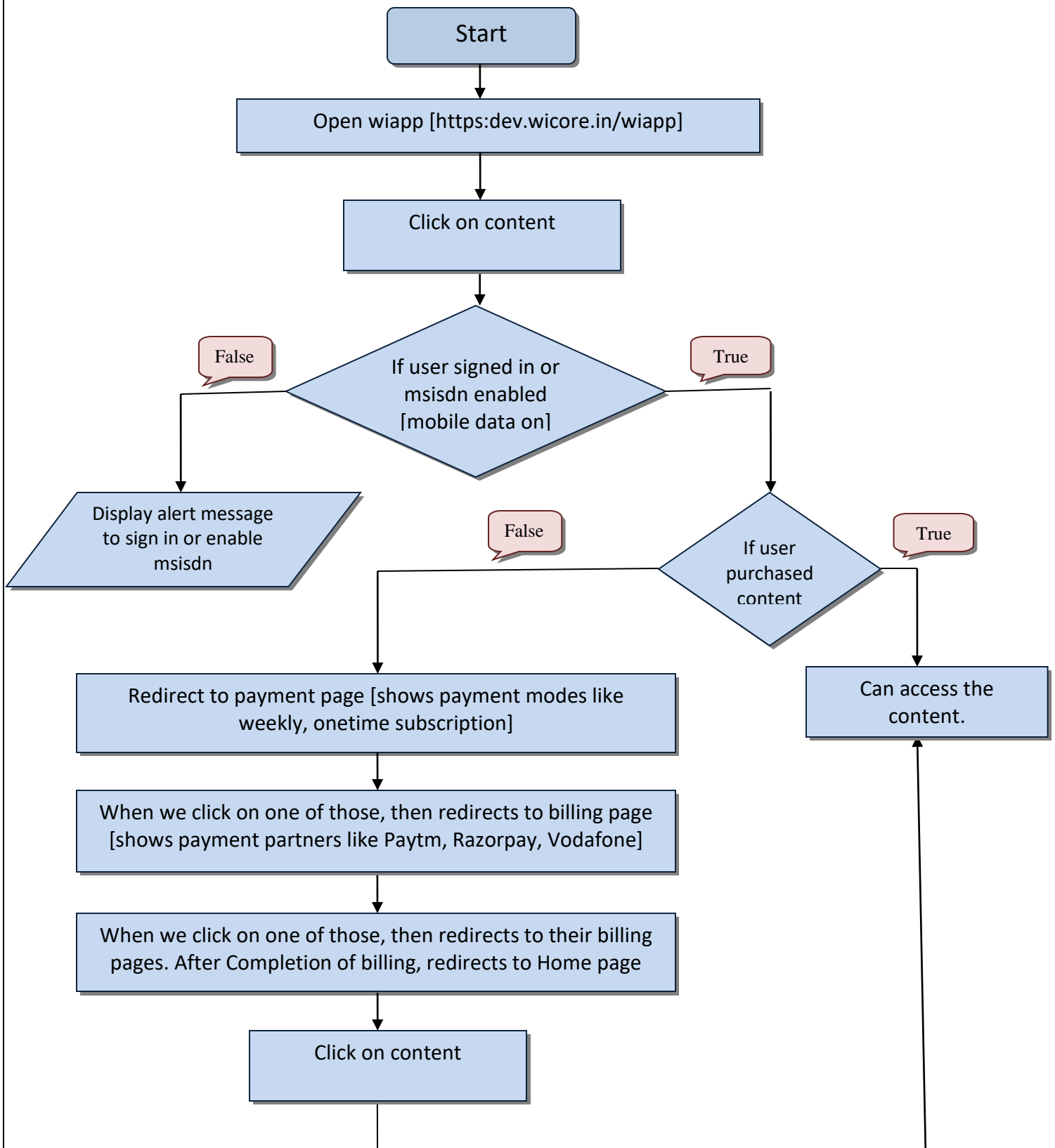
    if (!cookieID.equals("") && !cookieID.equals("NA")) { //cookie
exists.
        userID = CookieUtility.getUserIdFromCookieId(cookieID, "");
    } else { //cookie doesn't exists.
        //here we will do nothing.
    }

    Log.info(log_info+" gmail userID: " +userID+" | GMAIL cookieID: "
+cookieID);
}

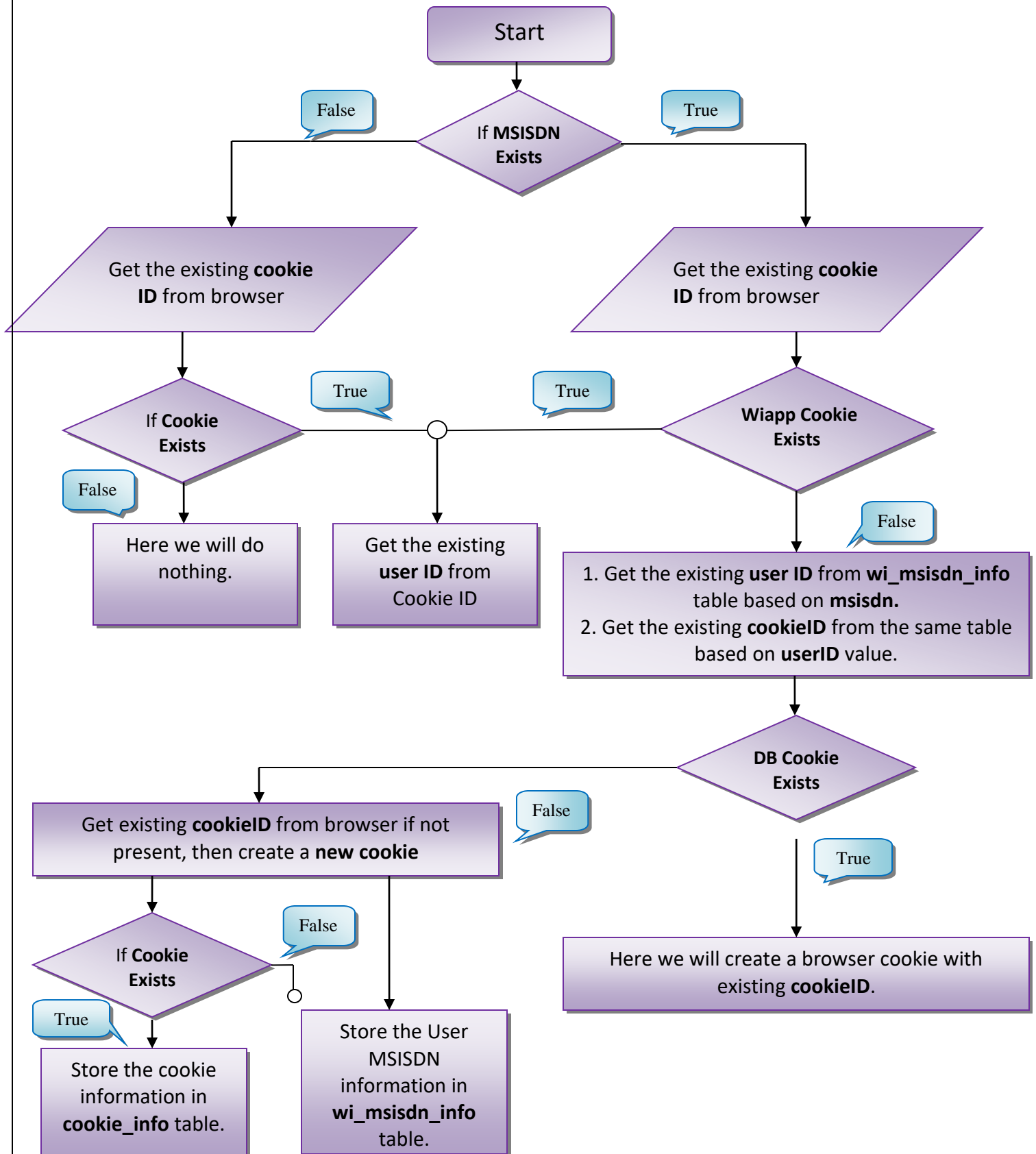
    Log.info(log_info+" Overall userID: " +userID+" | Overall
cookieID: " +cookieID);

```

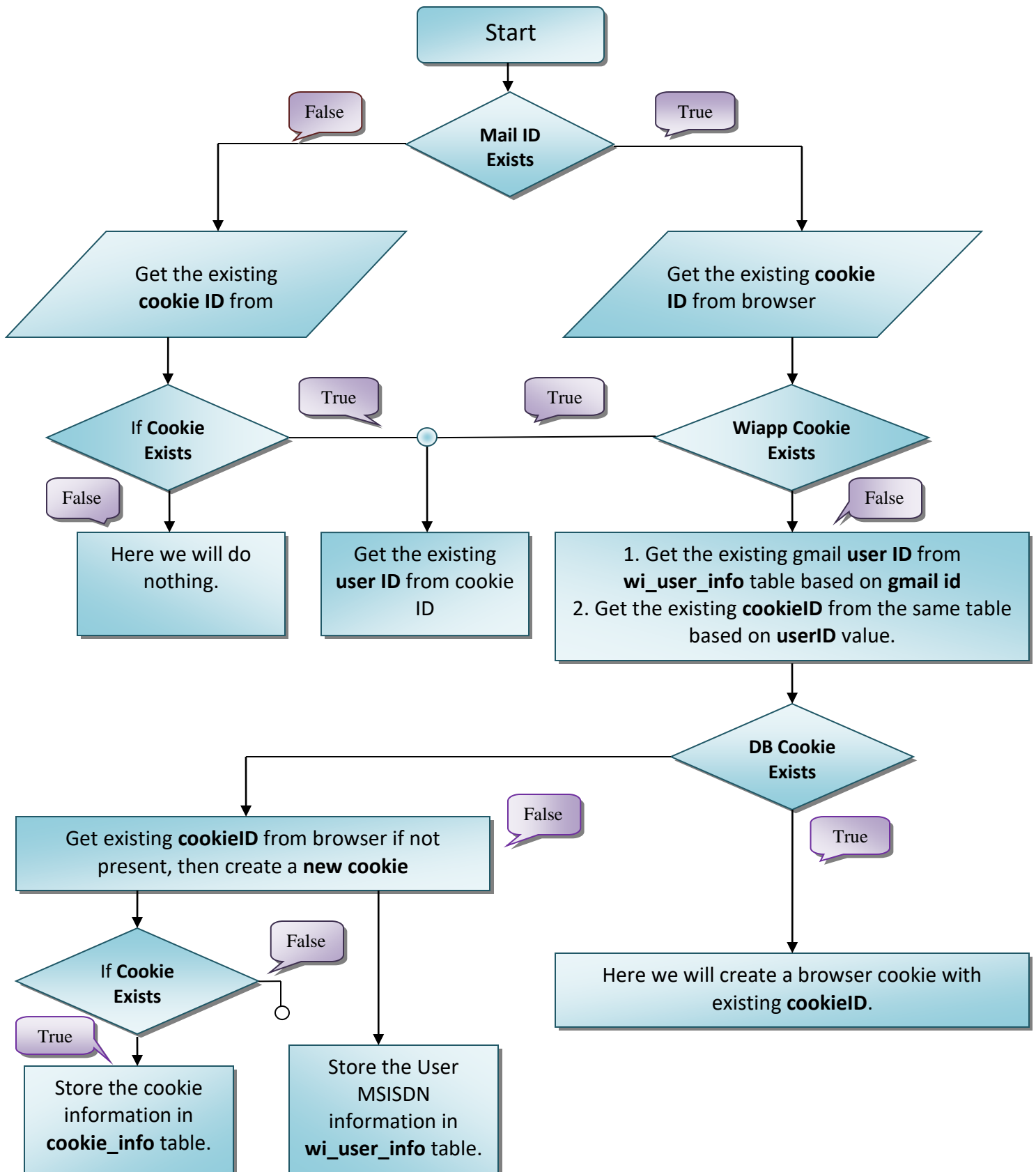
Billing flow – Flow Chart:



Code flow [MSISDN]- Flow Chart:



Code flow [Gmail]- Flow Chart:



Final List of files changed in WiApp :

- | | |
|-----------------------|---|
| 1. Index.java | [com.wicore.wiapp.api] |
| 2. CookieUtility.java | [com.wicore.wiapp.utility] |
| 3. CookieUtility.java | [com.wicore. wiapp.utility] |
| 4. RenderHTML.java | [com.wicore.wiframeTest] |
| 5. HTMLUtility.java | [com.wicore.wiframeTest] |
| 6. Custom.js | [/data/wiapp/WiappElements/content/CardVerticalContent/js/] |
| 7. Google_signin.js | [/data/wiapp/WiappElements/login/LoginDefaultStyle/js/] |

DB Changes [wiappdb]:

DDL Changes: NO

DML Changes: NO

Infrastructure / Environmental Configuration Elements: No

Test cases:

MSISDN Test Cases: [No Sign in]

1. Disable MSISDN and GMail → Click on content → Doesn't Redirect to Billing page or not able to access content → Display alert box. **Result: OK**
2. Clear all cookies → Enable MSISDN → Go to billing → Do billing → Click on content → Access content → Close window → Access content **Result: OK**
3. Clear all cookies → Enable MSISDN → Access content → content must be accessed **Result: OK**
4. Disable MSISDN → Again Enable it → Access content → Content must be accessed **Result: OK**
5. Clear all cookies → Disable MSISDN → Again Enable it → Access content → content must be accessed **Result: OK**

GMail Test Cases: [No MSISDN]

6. Disable MSISDN → Clear all cookies → Sign in → Click on content → Go to billing → Do billing → Click on Content → Access content → Again close window → Access content → User can access the content. **Result: OK**
7. Sign Out → Click on content → Display alert box. → User cannot access Content **Result: OK**
8. Sign Out → Sign in → Click on content → Content must be accessed. **Result: OK**
9. Sign Out → Clear all cookies → Sign in → Content must be accessed. **Result: OK**

MSISDN & Gmail Test Cases:

10. Enable both MSISDN & Sign in → Click on content → User can access the content **Result: OK**

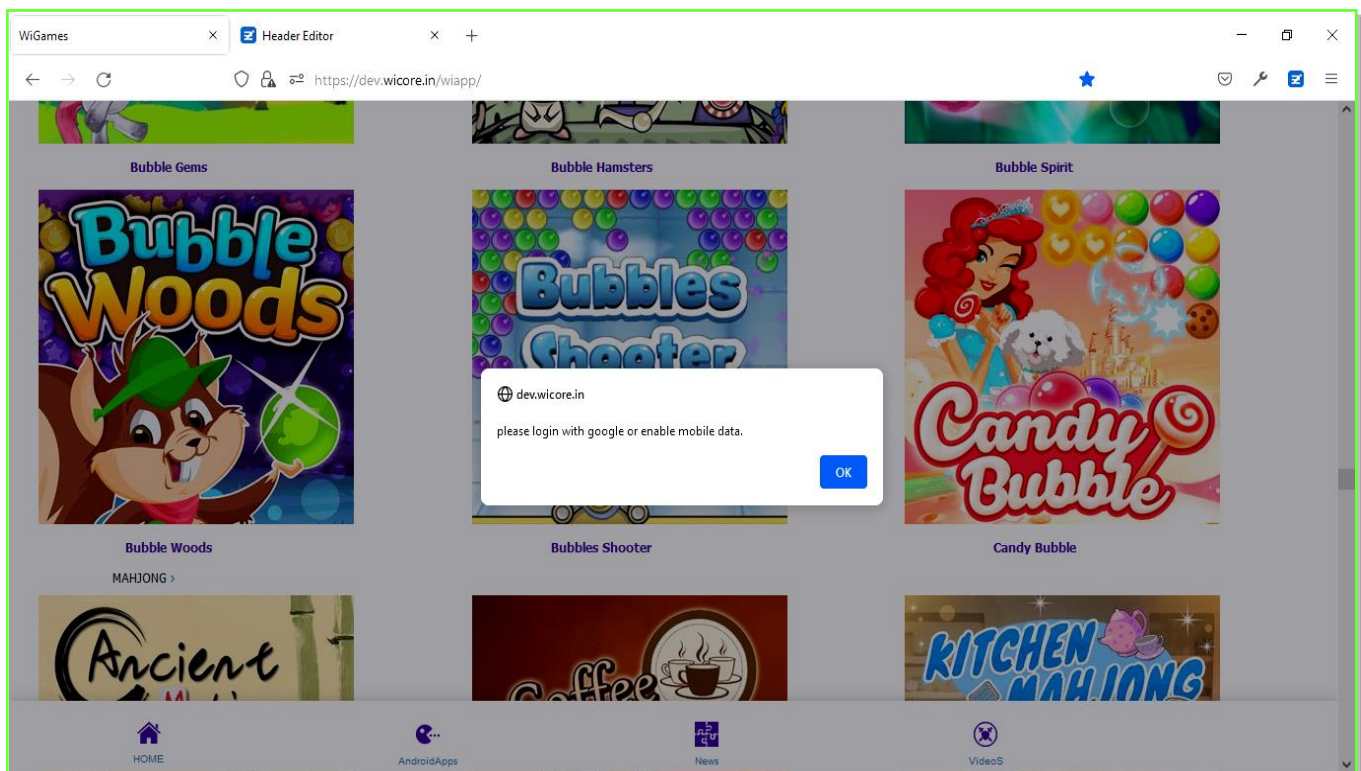
[These conditions need to be implemented]

11. Clear all cookies → Enable both MSISDN & Sign in → Click on content → User can access the content **Result: Not OK**

12. So Disable MSISDN & Enable Sign in -> Click on content -> User can access the content. **Result: Not OK**

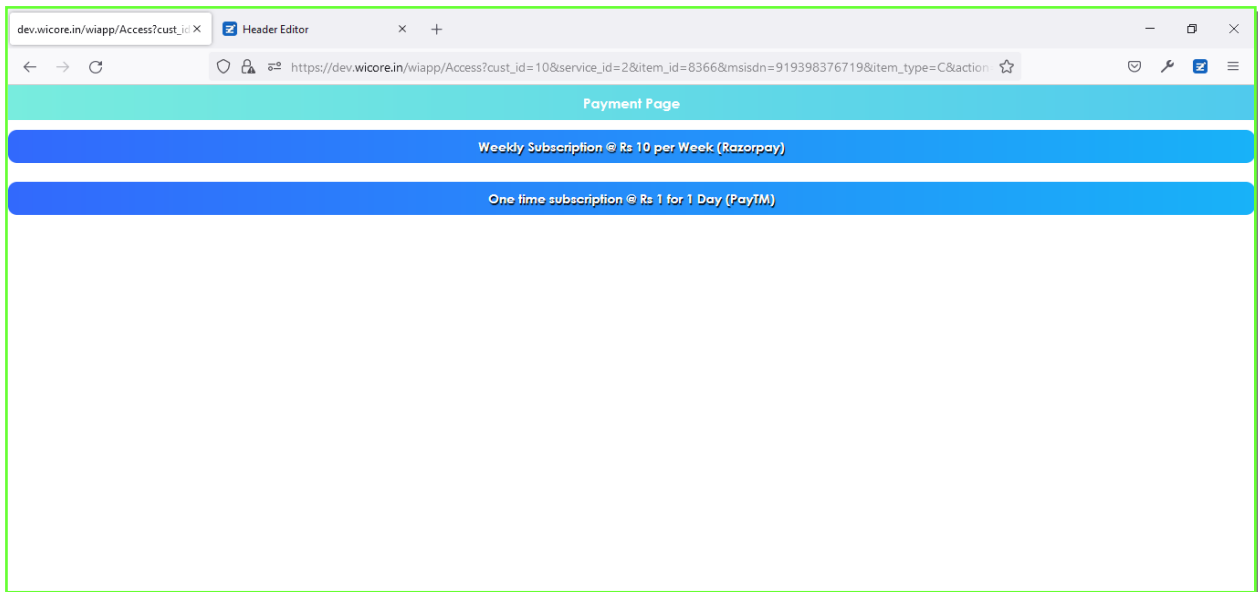
Failure Cases:

When the user clicks on content without signing in or enabling MSISDN (enabling mobile data). Then the user can not access the content. It displays alert box on the screen to sign in or enable msisdn.

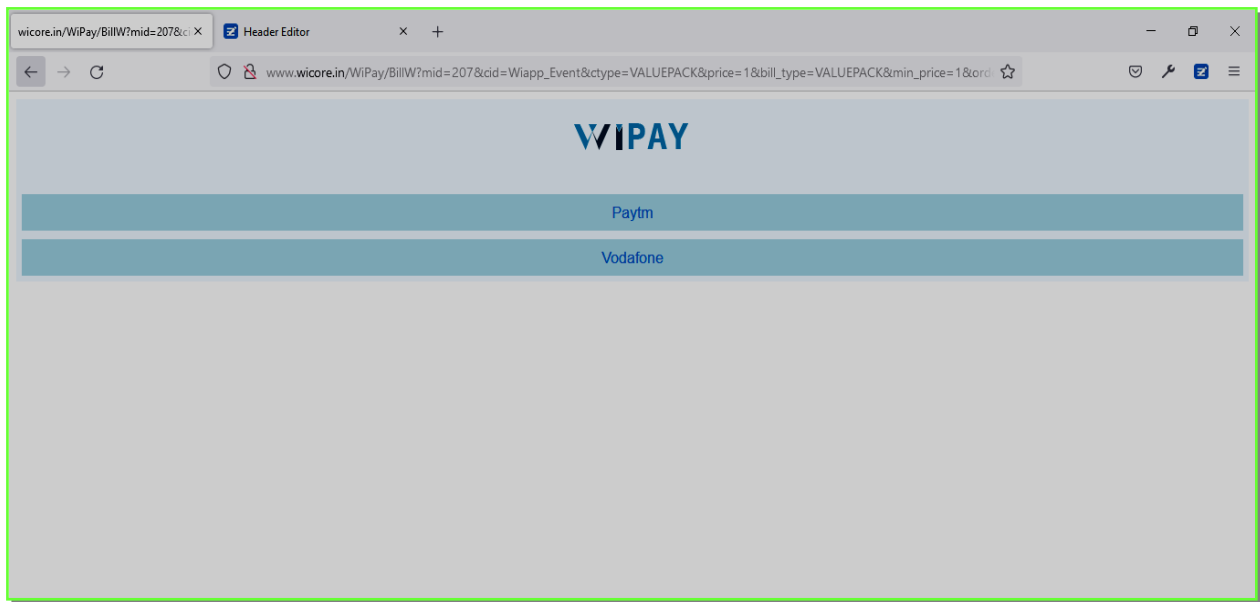


Success Cases:

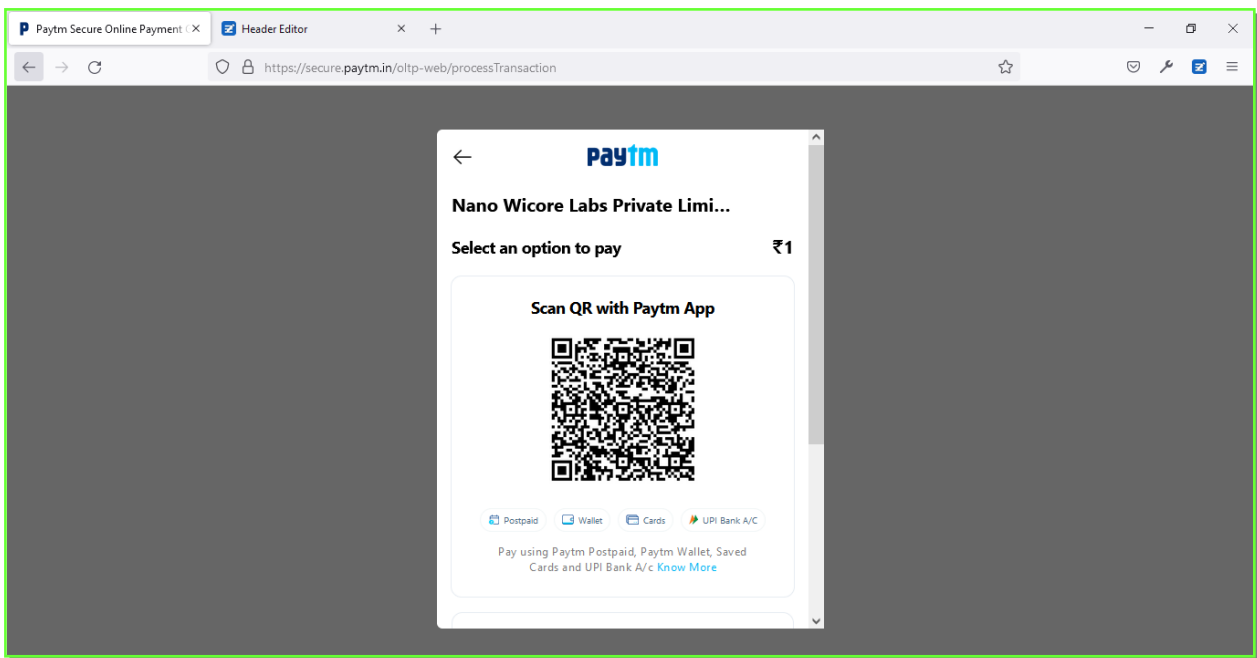
1. When the user signs in or enable the msisdn or done both. When he/she clicks on content. The page will be redirected to billing page.



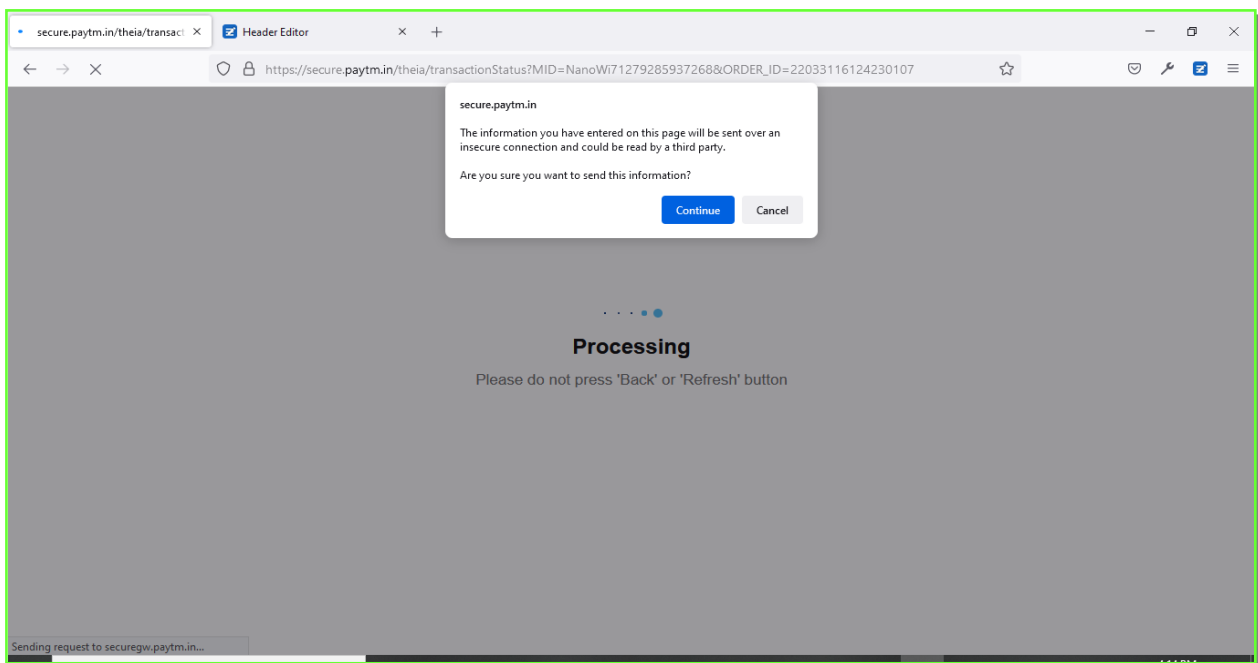
2. After clicking on any one of the subscriptions. It leads to payment page.

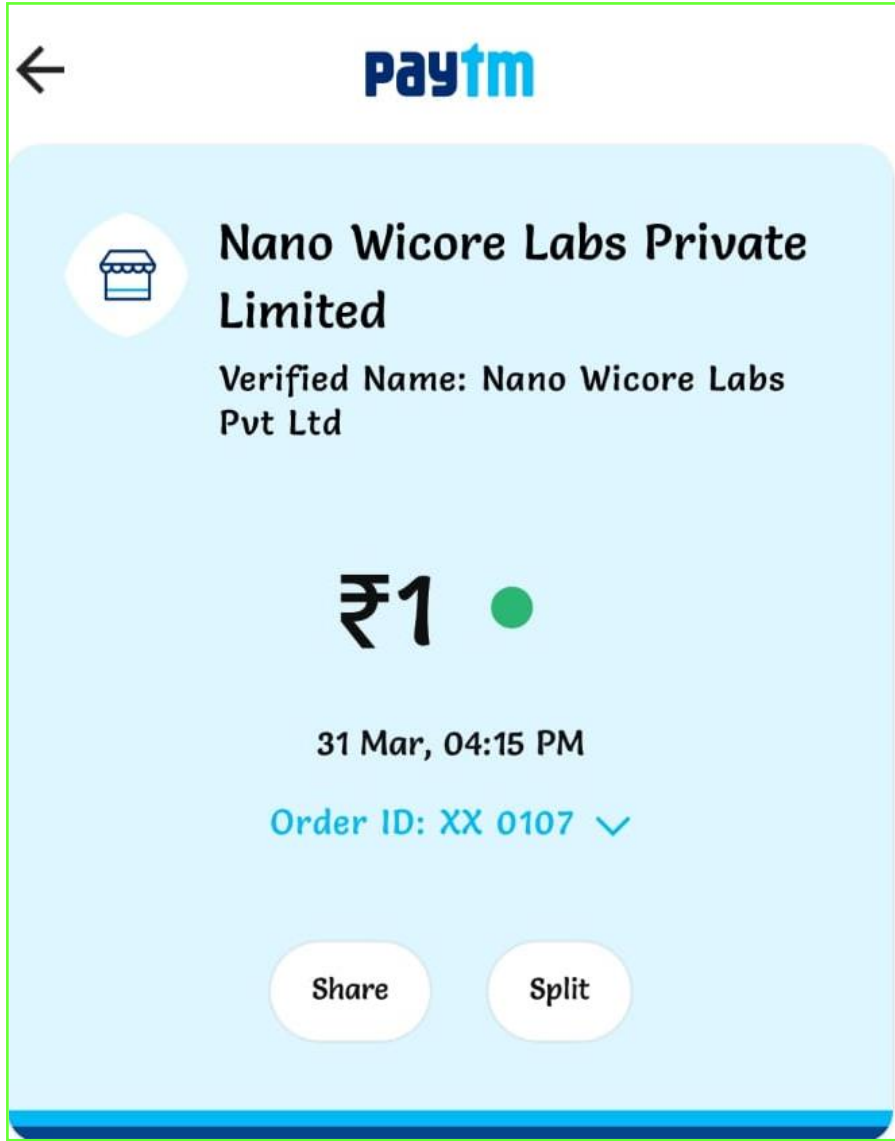


- When the user clicks on Paytm, the page will be redirected to paytm payment page.



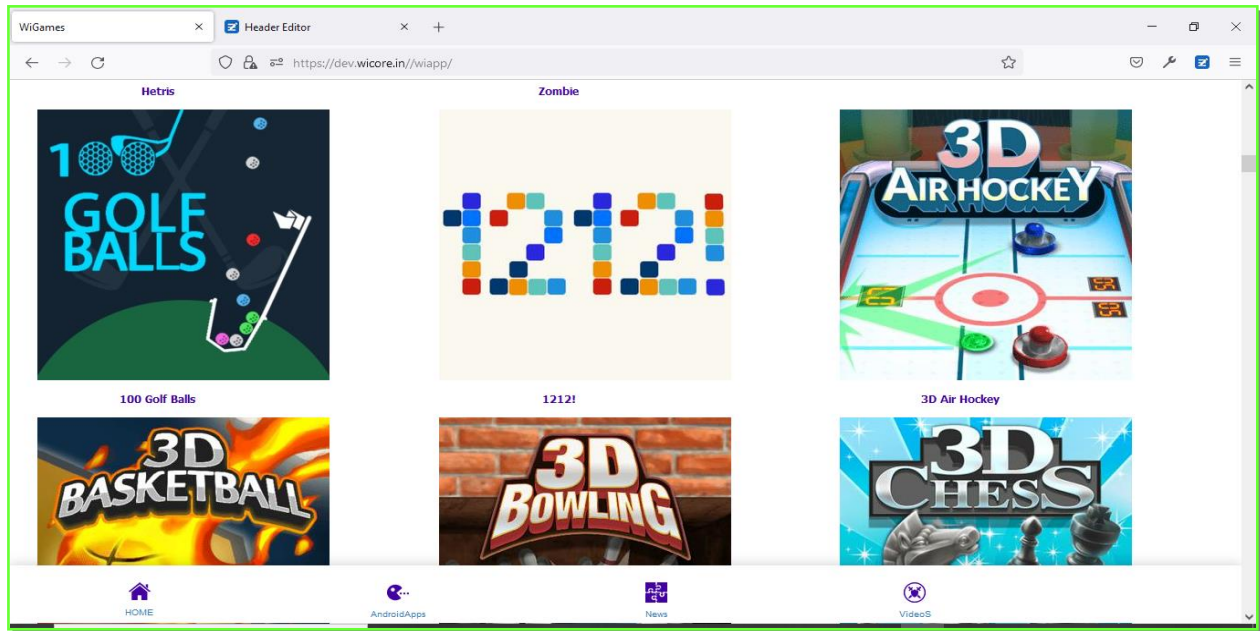
- When the user purchases the content using any UPI application, then the page will be processed to confirm the payment.



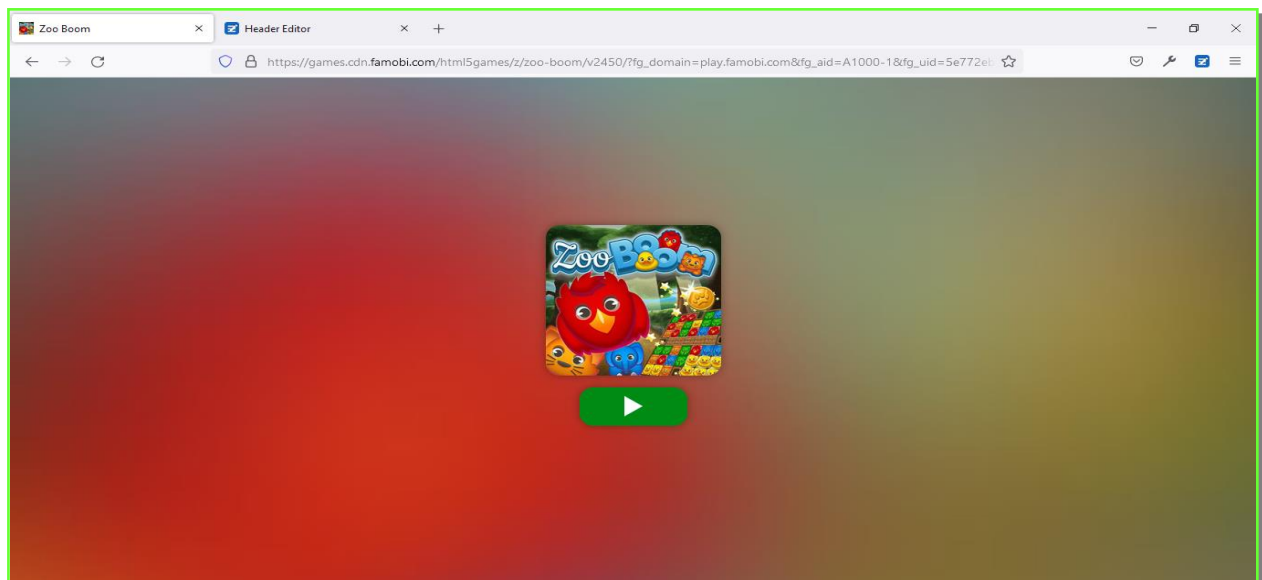


User Payment Page [Paytm Application].

5. After completion of payment, then the user will be redirected to Home Page.



6. So, finally when the user clicks on content [either enable sign in or msisdn], then the user can access the content.



-----THE END-----