UNUSED LOGIN FLOW DESCRIPTION

A login Flow version of the Modal was created by Ashley Sarah Bee but not included and a modal within Experience Builder Page was chosen

However this document illustrates use of Login Flow for the purpose of displaying Weekly Information received from Lightning Message Channel. This was not tested as a messageChannel for login flow was not created

WHY LOGIN FLOW?

-Login Flow is not within the DOM for Hero Hub so an ideal candidate to use LMS
-The Portal can have multiple purposes so putting the screens in the Login Flow reminds the student to look up stats beyond shown.

ASSUMPTION

1. There are two custom checkbox fields created on Contact Standard Oblect – Logged_In__c and Weekly_News__c which are not on the Page Layout with default values false
2. There is a Scheduled Flow which at the end of the week sets Weekly_News__c to False

OVERALL STRATEGY

1. Create a Flow to display a Welcome Screen with some base stats to the user after successful login the first Time he/she logs in
2. Every week new Stats are posted by LMS and pushed to the Flow by a LWC component on the screen. Whether the screen itself is a LWC component or Display Text can be chosen. My example used Display Text and LWC did not have html
3. Every Week when the user logs into the portal the stats for the week are displayed after login for the first login in the week or the next.
4. LWC component subscribes to LMS, parses the message, populates the Flow parameters and pushes them to the flow through FlowAttributeChangeEvents
5. The meta-xml file establishes the parameters to send to the flow

LWC Files

receivedModalMessage.js

```
import { LightningElement,api,track } from 'lwc';
import SAMPLEMC from"@salesforce/messageChannel/WeeklyMessage__c";
import { createMessageContext, subscribe } from"lightning/messageService";
import {FlowAttributeChangeEvent} from 'lightning/flowSupport';

export default class ReceiveModalMessage extends LightningElement {
@api topHero1;
@api topHero2;
@api topTeam1;
@api topTeam2;
@api yourSquadName;
@api topHero1Arete;
```

```
@api topHero2Arete;
@api topTeam1Arete;
@api topTeam2Arete;
@api topWeeksTeam1;
@api topWeeksTeam2;
@api topTeamWeeks1;
@api topTeamWeeks2;
context = createMessageContext();
channel;
messageThread;
parsedMessage;
flowEvent1(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('yourSquadName',this.yourSquadName);
this.dispatchEvent(attributeChangeEvent);

}

flowEvent2(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topHero1',this.topHero1);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent3(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topHero1Arete',this.topHero1Arete);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent4(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topHero2',this.topHero2);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent5(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topHero2Arete',this.topHero2Arete);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent6(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeam1',this.topTeam1);
this.dispatchEvent(attributeChangeEvent);
}
flowEvent7(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeam1Arete',this.topTeam1Arete);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent8(){
```

```
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeam2',this.topTeam2);
this.dispatchEvent(attributeChangeEvent);
}
flowEvent9(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeam2Arete',this.topTeam2Arete);
this.dispatchEvent(attributeChangeEvent);

}
flowEvent10(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topWeeksTeam1',this.topWeeksTeam1);
this.dispatchEvent(attributeChangeEvent);
}
flowEvent11(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeamWeeks1',this.topTeamWeeks1);
this.dispatchEvent(attributeChangeEvent);

}flowEvent12(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topWeeksTeam2',this.topWeeksTeam2);
this.dispatchEvent(attributeChangeEvent);
}
flowEvent13(){
const attributeChangeEvent= new FlowAttributeChangeEvent ('topTeamWeeks2',this.topTeamWeeks2);
this.dispatchEvent(attributeChangeEvent);

}


connectedCallback(){
this.channel = subscribe(this.context, SAMPLEMC, (payload) => {this.messageThread =
payload.message;});
this.parsedMessage=this.messageThread.split(",");
this.yourSquadName=this.parsedMessage[0];
this.topHero1=this.parsedMessage[1];
this.topHero1Arete=Integer.valueof(this.parsedMessage[2]);
this.topHero2=this.parsedMessage[3];
this.topHero2Arete=Integer.valueof(this.parsedMessage[4]);
this.topTeam1=this.parsedMessage[5];
this.topTeam1Arete=Integer.valueof(this.parsedMessage[6]);
this.topTeam2=this.parsedMessage[7];
this.topTeam2Arete=Integer.valueof(this.parsedMessage[8]);
this.topWeeksTeam1=this.parsedMessage[9];
this.topTeamWeeks1=Integer.valueof(this.parsedMessage[10]);
this.topWeeksTeam2=this.parsedMessage[11];
this.topTeamWeeks2=Integer.valueof(this.parsedMessage[12]);

flowEvent1();
flowEvent2();
```

```
flowEvent3();
flowEvent4();
flowEvent5();
flowEvent6();
flowEvent7();
flowEvent8();
flowEvent9();
flowEvent10();
flowEvent11();
flowEvent12();
flowEvent13();






}


}
```

receivedModalMessage.js.meta-xml

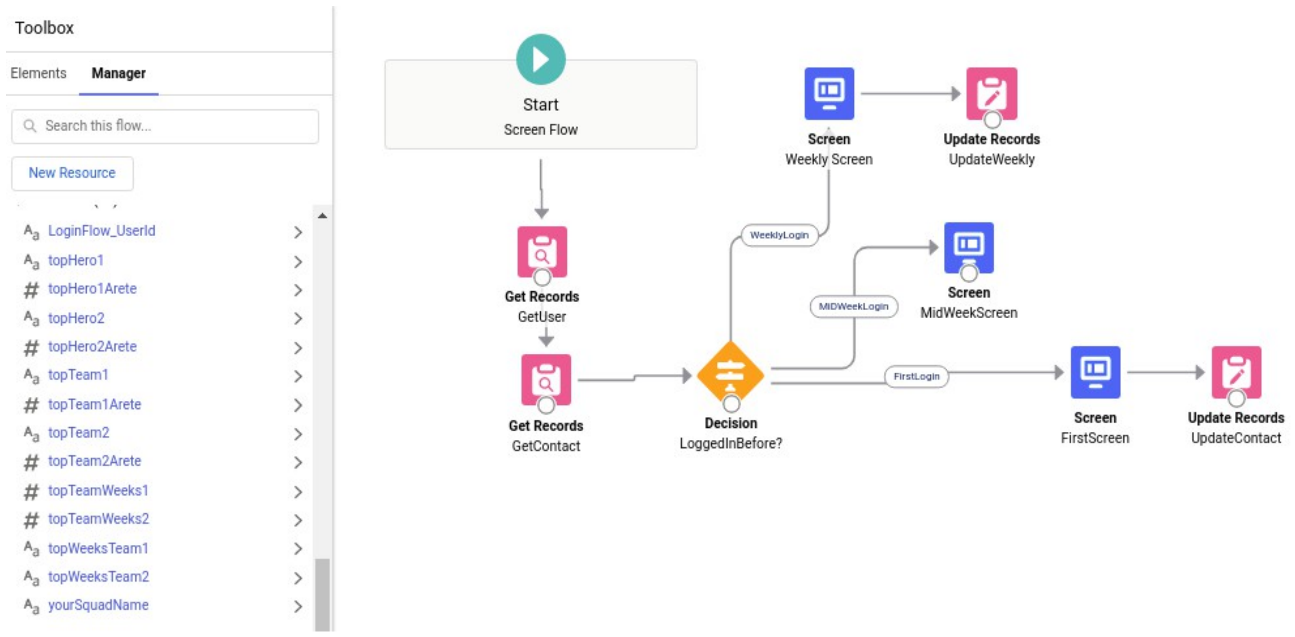
```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>52.0</apiVersion>
<isExposed>true</isExposed>
<targets>
<target>lightning__FlowScreen</target>
</targets>

<targetConfigs>
<targetConfig targets="lightning__FlowScreen">
<property name="yourSquadName" type="String" label="yourSquadName" role="outputOnly"/>
<property name="topHero1" type="String" label="topHero1" role="outputOnly"/>
<property name="topHero2" type="String" label="topHero2" role="outputOnly"/>
<property name="topHero1Arete" type="Integer" label="topHero1Arete" role="outputOnly"/>
<property name="topHero2Arete" type="Integer" label="topHero2Arete" role="outputOnly"/>
<property name="topTeam1" type="String" label="topTeam1" role="outputOnly"/>
<property name="topTeam2" type="String" label="topTeam2" role="outputOnly"/>
<property name="topTeam1Arete" type="Integer" label="topTeam1Arete" role="outputOnly"/>
<property name="topTeam2Arete" type="Integer" label="topTeam2Arete" role="outputOnly"/>
<property name="topWeeksTeam1" type="String" label="topWeeksTeam1" role="outputOnly"/>
<property name="topWeeksTeam2" type="String" label="topWeeksTeam2" role="outputOnly"/>
<property name="topTeamWeeks1" type="Integer" label="topTeamWeeks1" role="outputOnly"/>
```

```
<property name="topTeamWeeks2" type="Integer" label="topTeamWeeks2" role="outputOnly"/>
</targetConfig>
</targetConfigs>
</LightningComponentBundle>
```

Message Channel Folder needs the following meta.xml file
weeklyMessage.messageChannel-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningMessageChannel xmlns="http://soap.sforce.com/2006/04/metadata">
<masterLabel>WeeklyMessage</masterLabel>
<isExposed>true</isExposed>
<description>A Weekly Lightning Message comprising Leaderboard Data</description>
</LightningMessageChannel>
```

Flow Details



Get User

Gets the User Record for Id = LoginFlow_UserId where LoginFlow_UserId is the built in listener that inherits the property when you declare in Setup this flow as Login Flow

GetContact

Gets the Contact Record where the Id is equal to the ContactId of the logged in Community User

Decision Element LoggedInBefore? Has 3 outcomes.

1. FirstLogin which is the very first time the user logs in at all when Logged_In__c checkbox on corresponding Contact as well as Weekly_News__c is false

2. WeeklyLogin which is the first time in a week that the user logs in when Logged_In__c checkbox is checked and Weekly_News checkbox is unchecked on the corresponding Contact

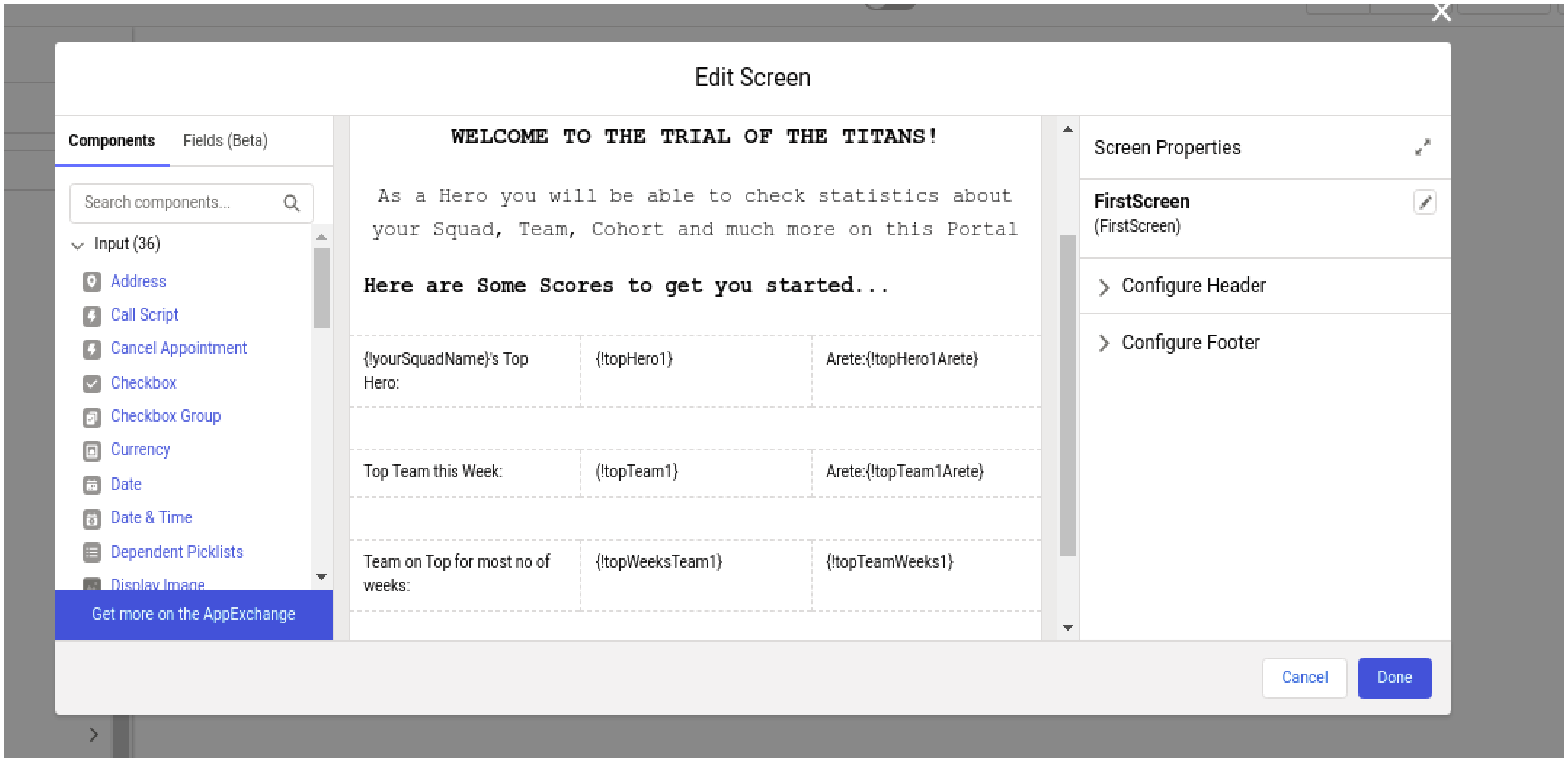
3. MidWeekLogin which is a login after the first one in a week where both checkboxes are checked

Screen Element

There are 3 screen Elements

One for Outcome 1 above and similar for outcomes 2 and 3

Screen 1 Example



WELCOME TO THE TRIAL OF THE TITANS!

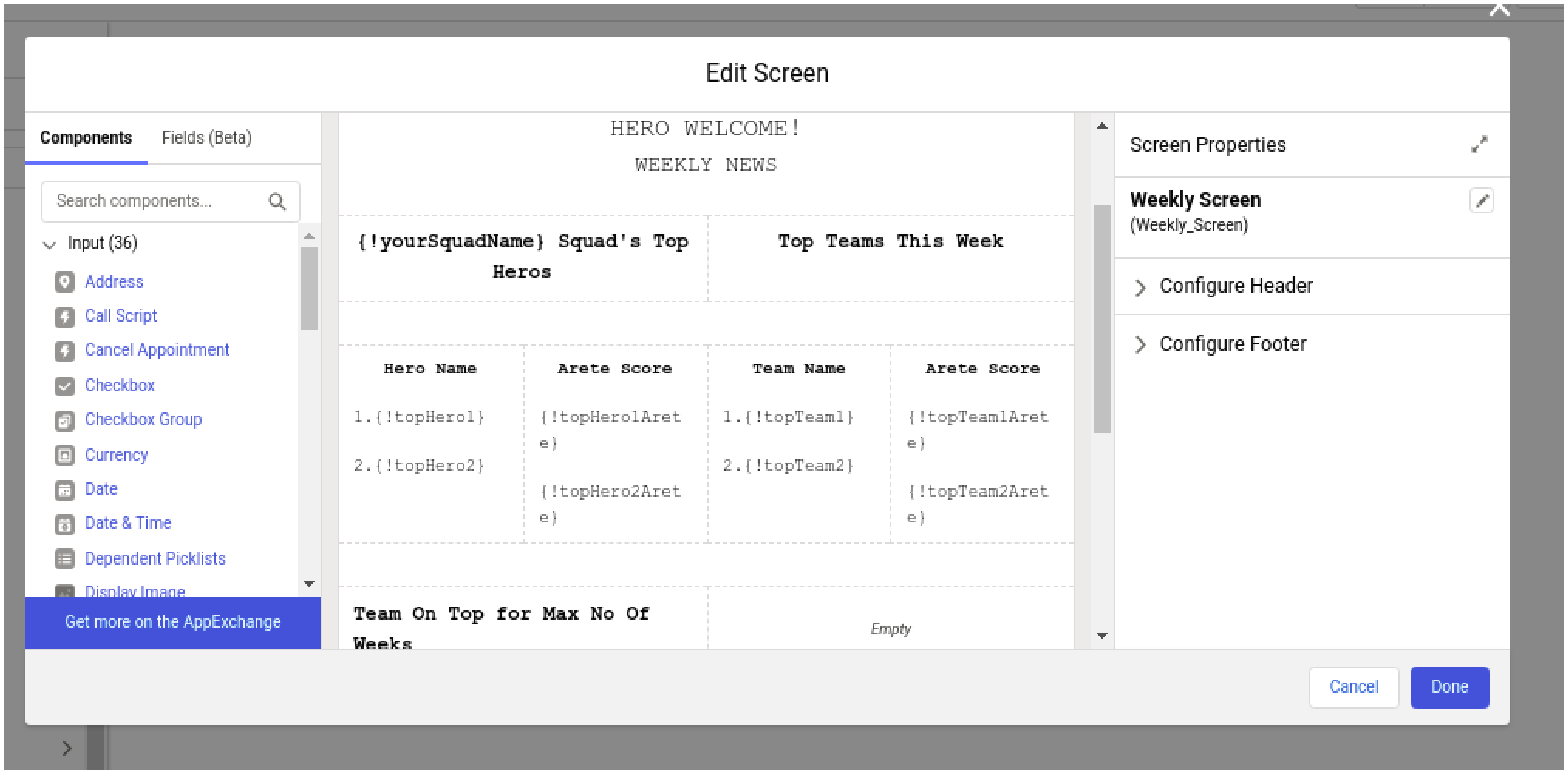
As a Hero you will be able to check statistics about your Squad, Team, Cohort and much more on this Portal

Here are Some Scores to get you started...

{!yourSquadName}'s Top Hero:	{!topHero1}	Arete:{!topHero1Arete}
Top Team this Week:	{!topTeam1}	Arete:{!topTeam1Arete}
Team on Top for most no of weeks:	{!topWeeksTeam1}	{!topTeamWeeks1}

Above this display the displayless component needs to be pulled and therein all the parameter variables need to be set and equated to Flow resources created on the MANAGER tab in the left hand panel The flow picture in the beginning shows the Flow resource variables on the left hand panel Then these parameters will show up when placed in Display Text

Screens 2 and 3 Example



Edit Screen

Components Fields (Beta)

Search components...

Input (36)

- Address
- Call Script
- Cancel Appointment
- Checkbox
- Checkbox Group
- Currency
- Date
- Date & Time
- Dependent Picklists
- Display Image

Get more on the AppExchange

```
HERO WELCOME!
WEEKLY NEWS
```

{!yourSquadName} Squad's Top Heros	Top Teams This Week

Hero Name	Arete Score	Team Name	Arete Score
1.{!topHero1} 2.{!topHero2}	{!topHero1Arete} {!topHero2Arete}	1.{!topTeam1} 2.{!topTeam2}	{!topTeam1Arete} {!topTeam2Arete}

Team On Top for Max No Of Weeks	Empty

Screen Properties

Weekly Screen
(Weekly_Screen)

Configure Header

Configure Footer

Cancel Done

Again for this as well the lwc component needs to pulled on to the screen and parameters mapped to resource variables'

UpdateContact1 and UpdateContact2

For Decision Outcome 1 an update Element needs to be connected to Screen1 . This element will update the Contact with Id same as loggedin user's ContactId field. Logged_In__c field of that Contact is checked to true

For Decision Outcome 2 an update Element needs to be connected to Screen 2. This element will update the Contact with Id same as loggedin user's ContactId field. Logged_In__c field is already checked so it will check the Weekly_News__c checkbox.

For Decision Outcome 3. nothing needs to be done and the values loaded at the week's first login can be redisplayed until the scheduled flow resets Weekly_News__c

Activate Flow

Save Flow. In Flow settings, Advanced Section set property to With Sharing. Activate Flow.

Set Up Login Flow

Search for Login Flow in Set Up and create a new one by selecting the flow created as the Login Flow