KONY DIGITAL BANKING APPLICATION

Project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGYIN COMPUTER SCIENCE AND ENGINEERING

By

RISHABHCHAUHAN (151350)

UNDER THE GUIDANCE OF

Mr.RAMENDRA UMRAV

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

TABLE OF CONTENTS

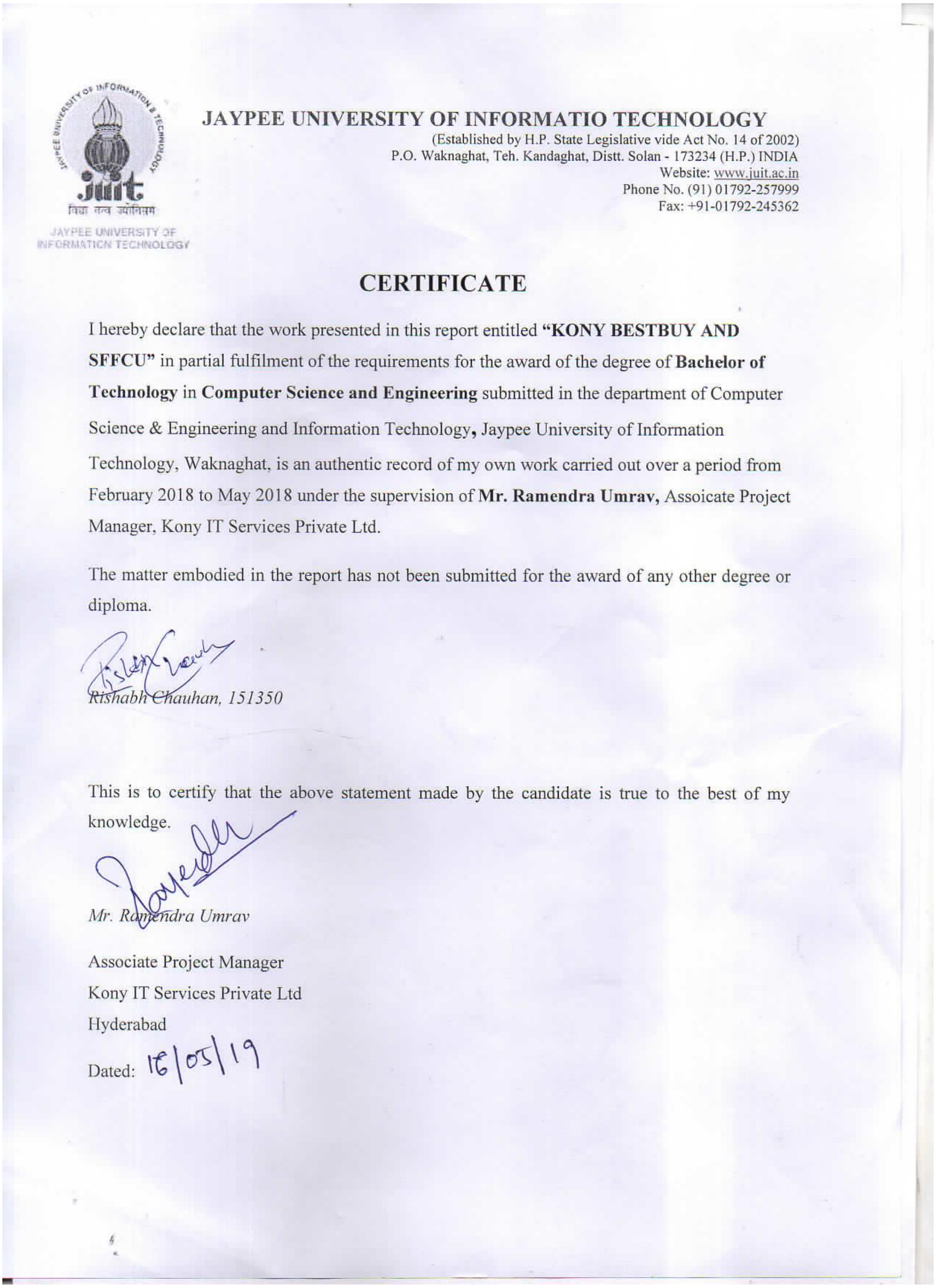
DECLARATION BY THE SCHOLAR

We hereby declare that the work reported in the B-Tech thesis entitled **“Kony Digital banking”** submitted at **Jaypee University of Information Technology, Waknaghat, India,** is an authentic record of my work carried out under the supervision of **MR. RAMENDRA UMRAV**. We have not submitted this work elsewhere for any other degree or diploma.

Rishabh Chauhan, 151350

Department of Computer Science and Engineering
Jaypee University of Information Technology
Waknaghat, India

Dated:16/5/19



JAYPEE UNIVERSITY OF INFORMATIO TECHNOLOGY

(Established by H.P. State Legislative vide Act No. 14 of 2002)
P.O. Waknaghat, Teh. Kandaghat, Distt. Solan - 173234 (H.P.) INDIA
Website: www.juit.ac.in
Phone No. (91) 01792-257999
Fax: +91-01792-245362

CERTIFICATE

I hereby declare that the work presented in this report entitled **"KONY BESTBUY AND SFFCU"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat, is an authentic record of my own work carried out over a period from February 2018 to May 2018 under the supervision of **Mr. Ramendra Umrav,** Assoicate Project Manager, Kony IT Services Private Ltd.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Rishabh Chauhan, 151350

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Ramendra Umrav

Associate Project Manager
Kony IT Services Private Ltd
Hyderabad

Dated: 16/05/19

ACKNOWLEDGEMENT

It is a pleasure to express my deep appreciation and gratitude to my mentor and to guide Mr. RamendraUmrav, Associate Senior Manager, Kony Services Private Ltd., Hyderabad, Telangana. His dedication and keen interest, and primarily, his overwhelming attitude to help his team had been solely and mainly responsible for the completion of my work. Timely advice, meticulous review, scholarly advice and a scientific approach have helped me a lot.

I owe a deep sense of gratitude to all my team, respectful and supportive, who helped me to participate in the very early stages of project development and I expect them to continue to guide me constantly. Their prompt inspirations, timely suggestions with kindness, enthusiasm and dynamism allowed us to complete my thesis.

RISHABH CHAUHAN(151350)

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Program Interface
ASA	Authentication Service Agency
AUA	Authentication User Agency
CIDR	Central Identities Data Repository
CSV	Comma Separated Value
DLL	Digital Logic Library
FMR	Finger Minutiae Record
FP	Fingerprint
GDSPL	Gemalto Digital Security Private Limited
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
ID	Identity
IIR	Iris Image Record
ISO	International Standards Organization
JSON	JavaScript Object Notation
MNREGA	Mahatma Gandhi National Rural Employment Guarantee Act
OTP	One Time Password/Pin
PC	Personal Computer
PD	Public Devices
PDF	Portable Document Format
PID	Personal Identity Data
PIN	Personal Identification Number
PoS	Point of Sale

RD	Registered Devices
SDK	Software Development Kit
SMS	Short Message Service
UI	User Interface
UIDAI	Unique Identification Authority of India
URL	Uniform Resource Locator
UX	User Experience
XML	Extensible Markup Language

LIST OF FIGURES

ABSTRACT

This report covers two immensely crucial mobile applications which I got a chance to develop during my internship (tentatively ending in August) in Kony, Hyderabad. The first application is Best Buy Serviceswhich was a certification project given to us. It behaves like a e-commerce application where we give a list of categories each of which has a list products. Each products has its own details and reviews. The second application is the Signal Financial Federal Credit Union app. A digital banking application that helps the customer enroll and register into the app. Manage their accounts and make transfers. The app uses services from the backend to authorize login and adding and deleting accounts as well as transferring money from one account to another.

CHAPTER – 1

INTRODUCTION

1.1About the Company

Kony is one the fastest growing mobile application solutions provider in the industry with over 1,600 live multi-channel apps in the B2C and B2B arenas. Kony provides an integrated mobile application development platform (MADP) to define, design, develop, test, deploy and manage multi-channel applications from a single code base. With Kony, you can deliver stunning user-first experiences, get to market faster and lower your application TCO. Kony also offers a suite of productized B2E and B2C apps that enable customers to quickly extend their business.

1.2Understanding Kony Platform

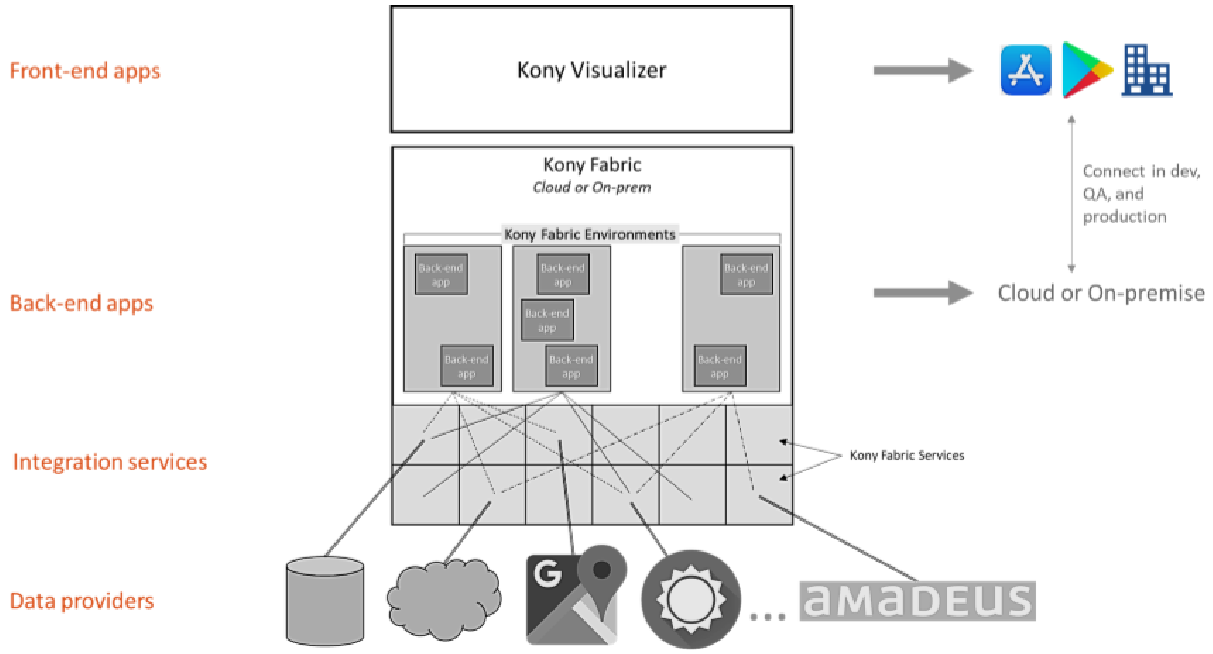


Fig 1.1 Platform Overview

The kony application works through 3 layers-

1. Kony Visualizer
2. Kony Fabric
3. Server Side.

1.3 Kony Visualizer

Visualizer is the platform developed by Kony that helps us build forms and screens. The client sided code and UI is designed at the Visualizer end. The following are its features-

1. Visualizer allows to manage multiple projects which are kept in a “workspace”.
2. You can only work at a single project at a time.
3. Whatever design we create on the visualizer is exactly how it will look like on the device when the app is running on it.
4. You are able to switch devices to tweak some changes you want.
5. Visualizer supports Apple, Android and Windows devices.

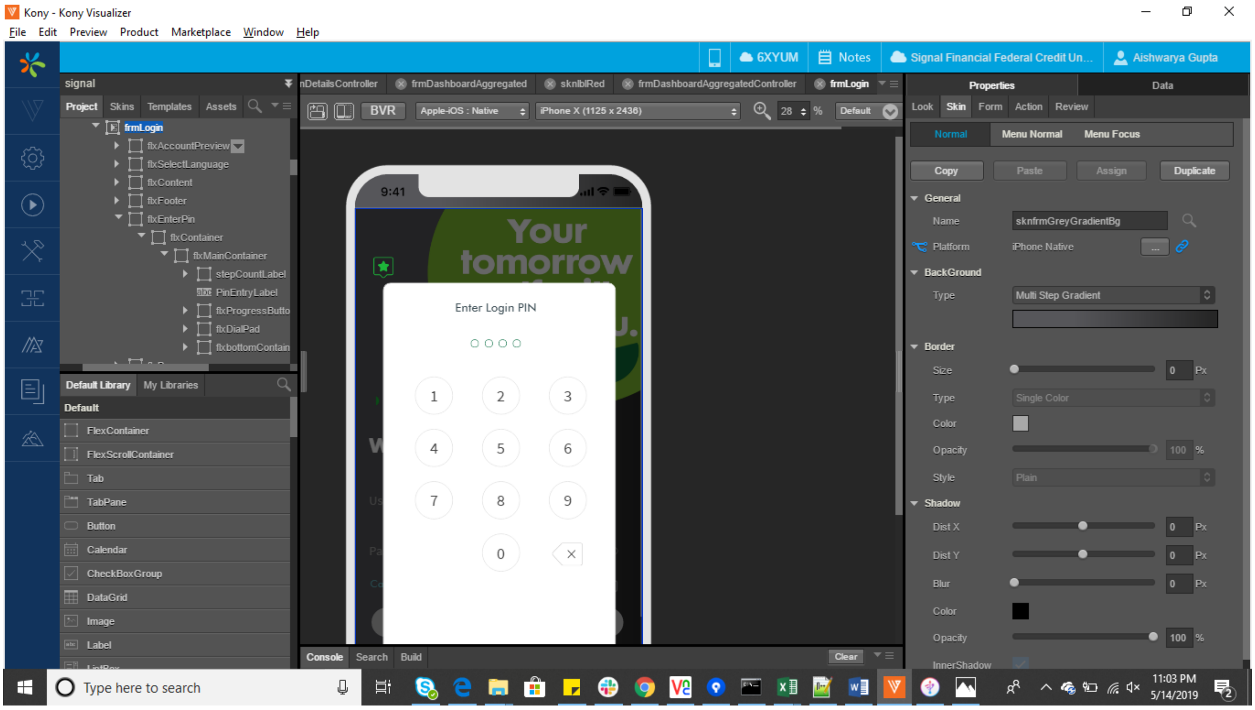


Fig 1.2 Visualizer

The above screenshot gives an overview of what the visualizer looks like.

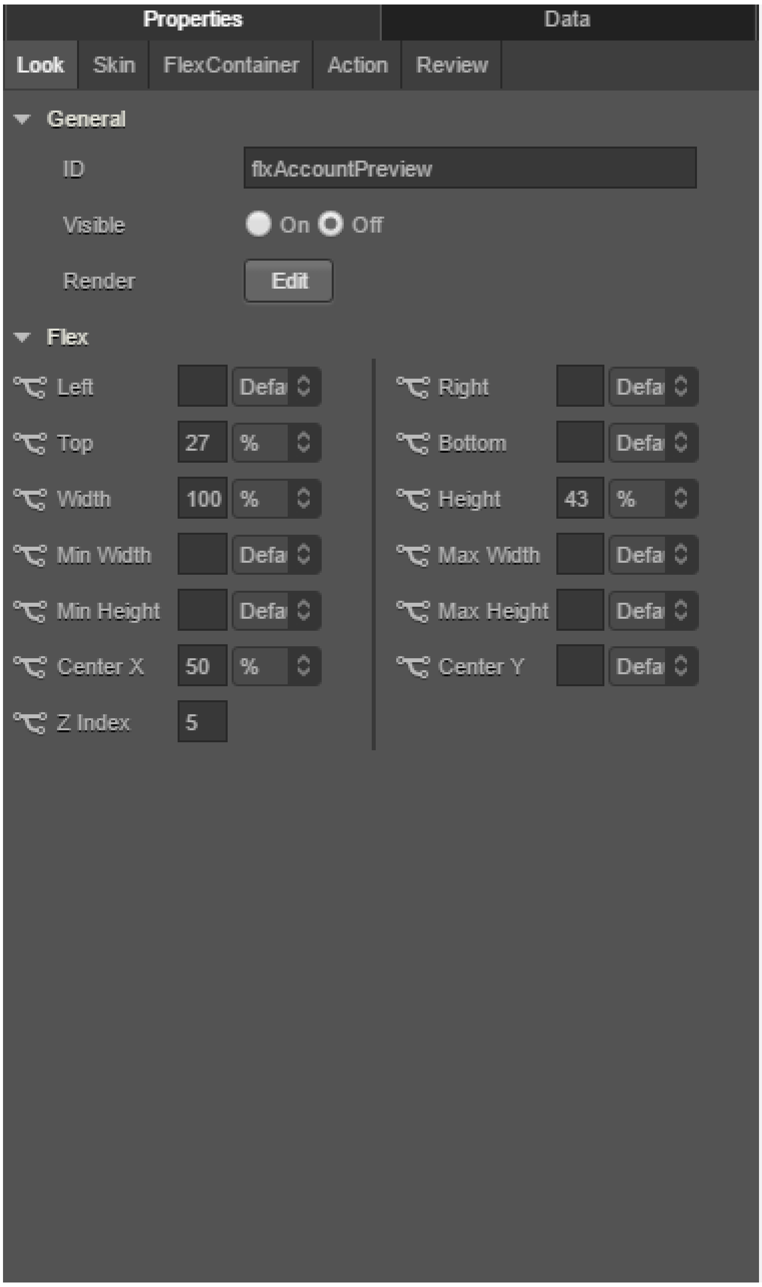
We have the project tab on the left-top side that contains all the forms, skins,assets,templates and components that are use in the project. Any new form that is created is shown in this tab.

It helps in managing resources of our project.

The right panel is the properties tab. The properties tab is where we can adjust the UI.

We have different tabs inside the properties window.

1.3.1 Look Tab – The look tab lets you adjust the size and position of the widget. The parameters for height, width and other things is taken in 3 different units – dp,px and %. The name of the widget can also be changed in this tab.



ID: flxAccountPreview

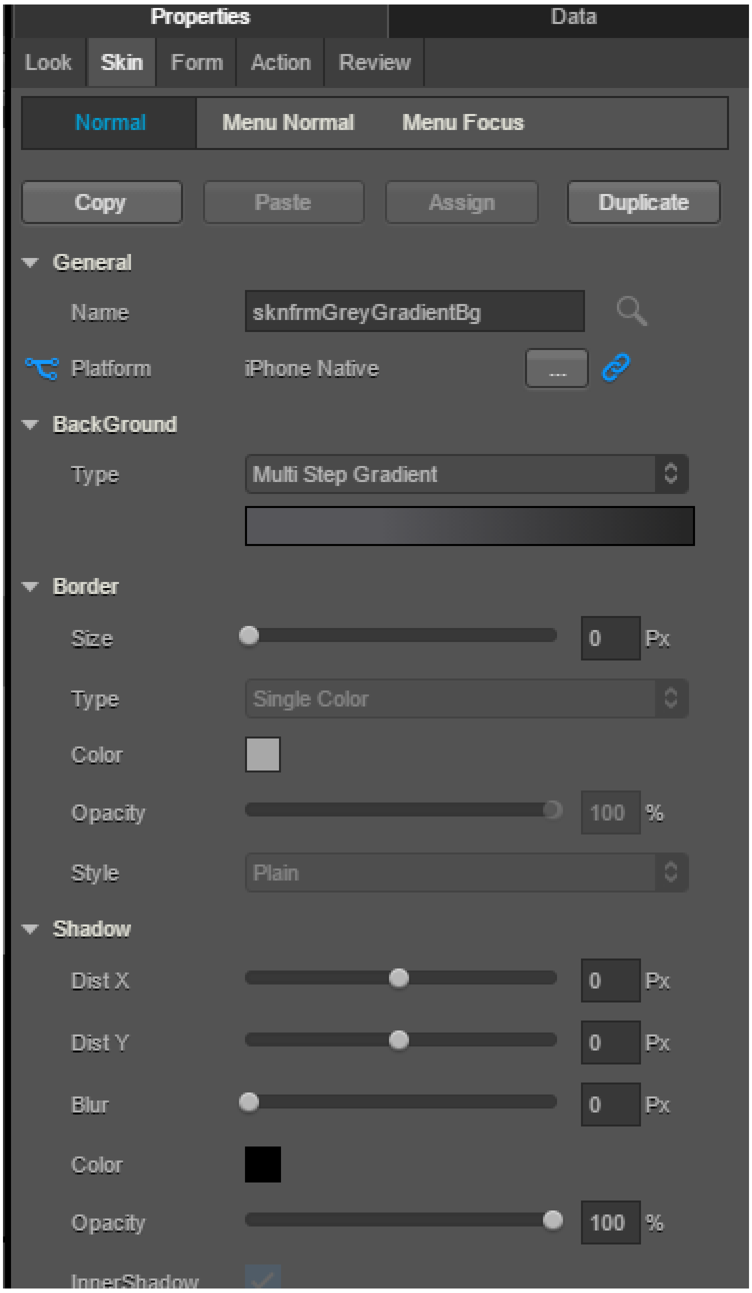
Visible: ○ On ● Off

Render: Edit

Left: Defa
Right: Defa
Top: 27 %
Bottom: Defa
Width: 100 %
Height: 43 %
Min Width: Defa
Max Width: Defa
Min Height: Defa
Max Height: Defa
Center X: 50 %
Center Y: Defa
Z Index: 5

Fig 1.3 Look

1.3.2 Skin Tab – The skin tab lets you change the look of the widget. It lets you play with the color,font,background and border for the widget. Skins can be created and applied to multiple widgets at a time. Changing the properties of a skin applies the changes to all the widgets the skin was applied to.



General

Name sknfrmGreyGradientBg

Platform iPhone Native

BackGround

Type Multi Step Gradient

Border

Size 0 Px

Type Single Color

Color

Opacity 100 %

Style Plain

Shadow

Dist X 0 Px

Dist Y 0 Px

Blur 0 Px

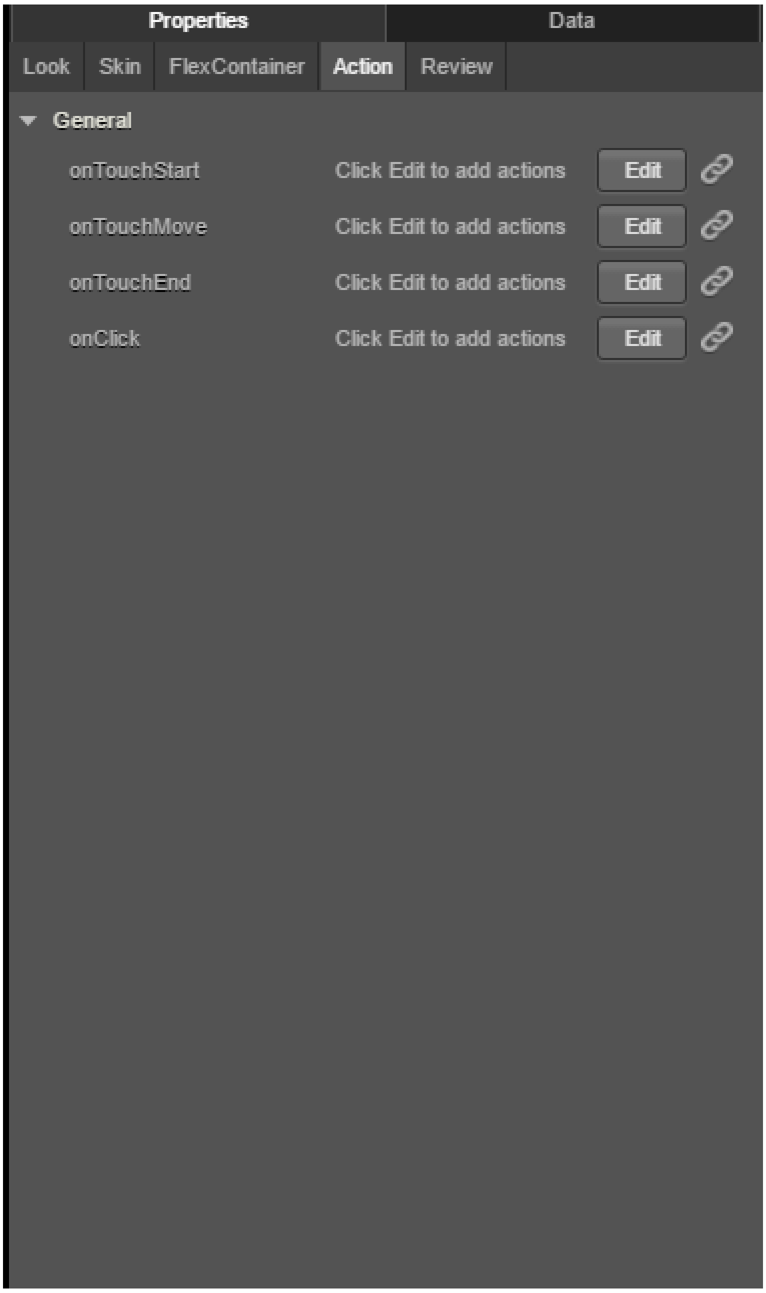
Color

Opacity 100 %

InnerShadow

Fig 1.4 Skin

1.3.3 Action Tab – The action tab lets to assign actions to the widgets. For example if we want to assign a on click action to a label, it is done through this tab. It allows you to add a code snippet or invoke a function or any other actions.



General

onTouchStart	Click Edit to add actions	Edit
onTouchMove	Click Edit to add actions	Edit
onTouchEnd	Click Edit to add actions	Edit
onClick	Click Edit to add actions	Edit

Fig 1.5 Action

1.3.4 The Widget Tab – It allows you to play with properties that are particular to widgets.

CHAPTER-2

2.1 Kony Products-

2.1.1 Kony Digital Banking – The Kony Digital Banking Solution delivers an amazing customer experience through modern, well integrated technology. The DBX also incorporates a kind of support for the back office side. This is called Customer 360.

The DBX app consists of three layers –

1. Digital Banking Applications.
2. Digital Banking Platform.
3. Secure Data storage.

2.1.2 Digital Banking Platform Architecture –

Scope of DBP –

1. Scope.
2. Application.
3. Middleware.
4. Data.

2.1.3 Client – There are external third party application for integration on parallel level. The client apps are developed using the MVC architecture.

2.1.4 Middleware – It comprises of Object Services,Management and Administration.The API's expose interfaces for third party consumptionand help connect the client with middleware.

2.1.5 Data – This layer enables access to the local database.

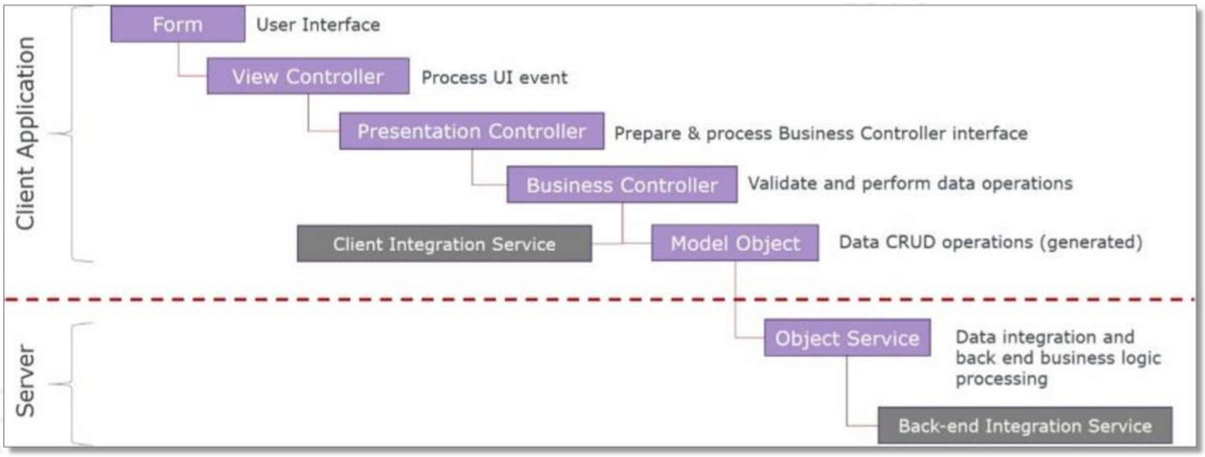

Client Application
Form
User Interface
View Controller
Process UI event
Presentation Controller
Prepare & process Business Controller interface
Business Controller
Validate and perform data operations
Client Integration Service
Model Object
Data CRUD operations (generated)
Server
Object Service
Data integration and back end business logic processing
Back-end Integration Service

Fig 2.1 Overview

2.2 Digital Banking Platform – Middleware

It provides the following services-

1. Administration Services.
2. Aggregation Services for Personal Finance Management Data.
3. Shared Business Object Services.
4. Open APIs and API-Management.
5. Complete Open-Banking Data Model.

The Digital Banking Platform provides tools that support the life cycle development , testing , deployment and maintainence.

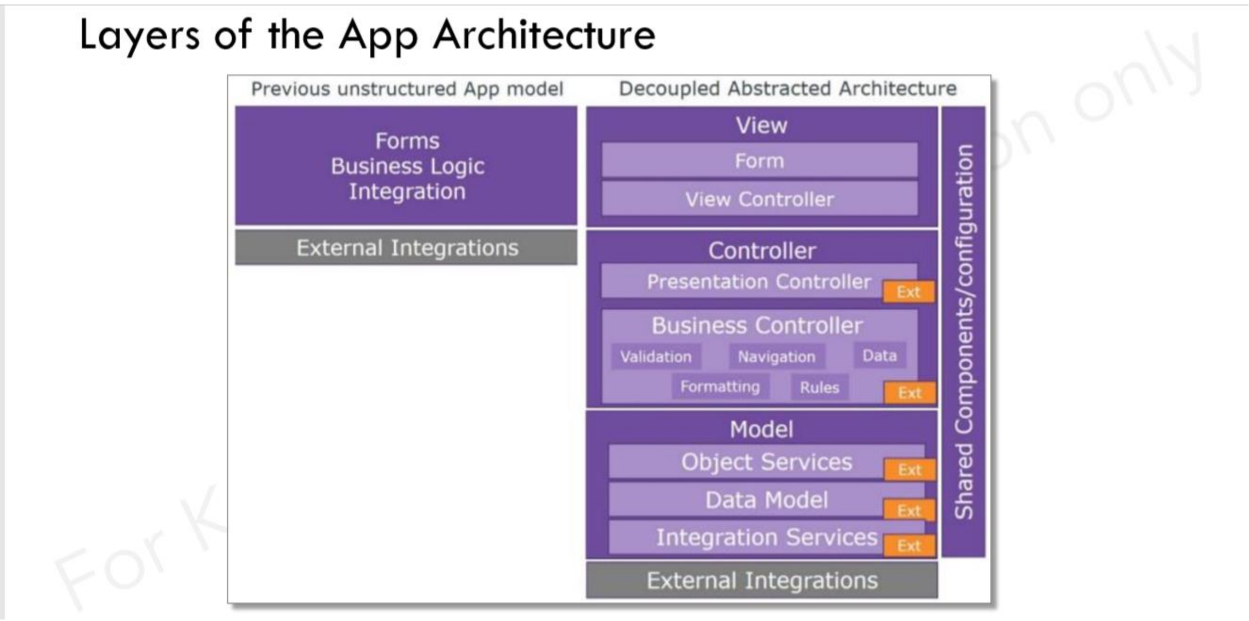


Fig 2.2 Layers of Architecture

2.3 DBP – Administration Services

Administration product is needed by Banks and Credit Unions in order to manage their banking products. The customer 360 can be used by –

1. Any lines of business.
2. Customer Services.
3. Digital Banking Team.
4. IT/Security Team.

The purpose of Customer 360 is to provide the administration side of the office tools to work and organize their end. It lets admins enter new users, change permissions , manage messages or give out alerts. The following are its features –

1. Employee management
2. Customer Management
3. Application Content management.
4. Reports.
5. Customer Services
6. Daashboard.

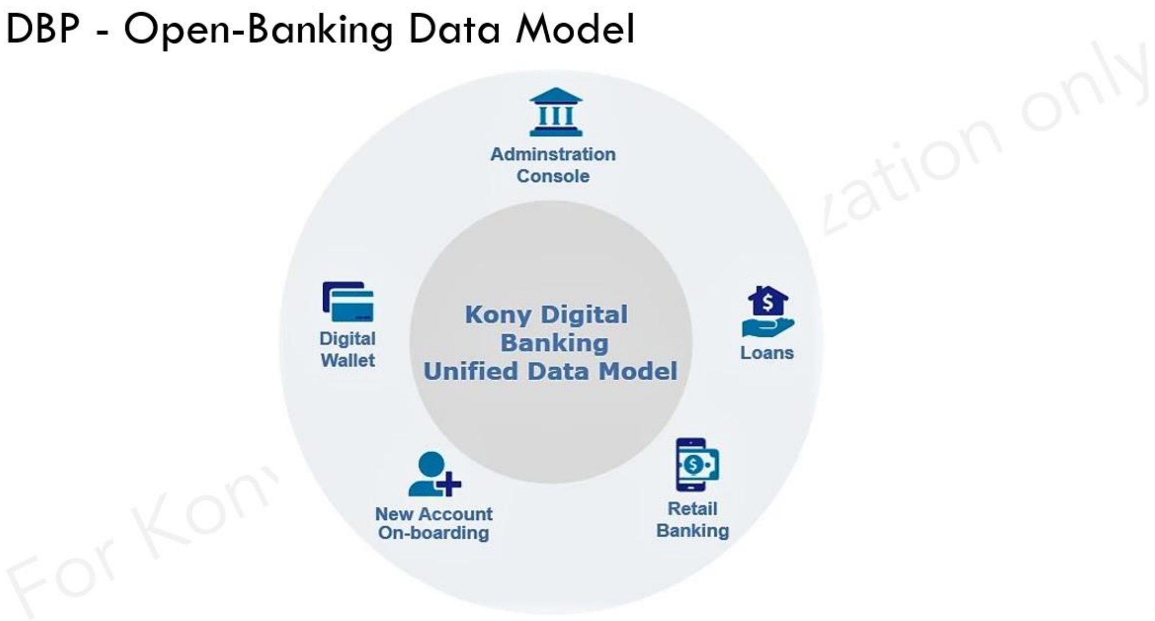


Fig 2.3 Data Model

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 BEST BUY

Best buy app was given to us as a certification project during my training at kony. The app was kind of e-commerce app for selling products like flipkart but on a shorter scale.
The app is divided into front end UI , middle part- The fabric and backend services.

TECHNOLOGIES USED LANGUAGE: Javascript

VERSION: ECMASCRIPT 7

IDE: Eclipse integrated with Kony Visualizer

Middleware: Kony Fabric

API Used:

Product- https://bestbuyapis.github.io/api-documentation/#products-api

Category- https://bestbuyapis.github.io/api-documentation/#categories-api

3.1.1 Front End –

The application is divided into three forms –

Category Form – This forms contains all the categories that have products. The categories have sub categories also.

Product Form – The product form contains the list of all the products with some details about them.

Product Details Form – This form contains all the details from rating to reviews. All the details are mentioned on this form.

3.2 Category Form –

The category forms consists of a header with the best buy icon, along with a search icon. The label below it shows the path we are on in the app category wise. It also has a segment widget. A segment is a widget that is used a lot in android applications. The segment is populated with data that comes from the backend.

This data comes from the best buy category api.

The data is fetched using the api at https://bestbuyapis.github.io/api-documentation/#categories-api. We request data from this api and what we get is categories for our app using the Mobile Fabric. The data comes in the form of JSON object which we can then use.

We associate our app with the services made on the Mobile Fabric.

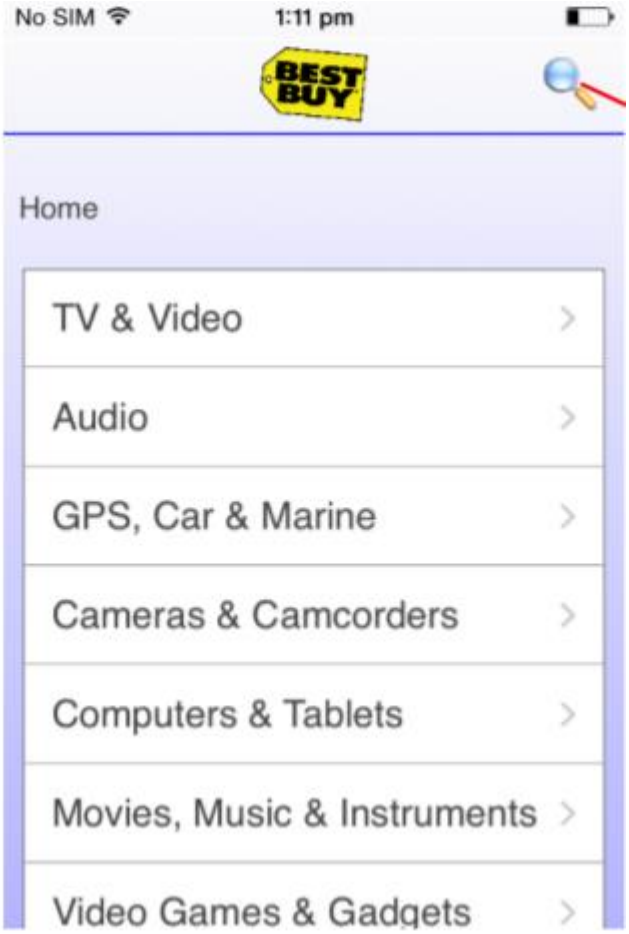


Fig 3.1 Best Buy Category

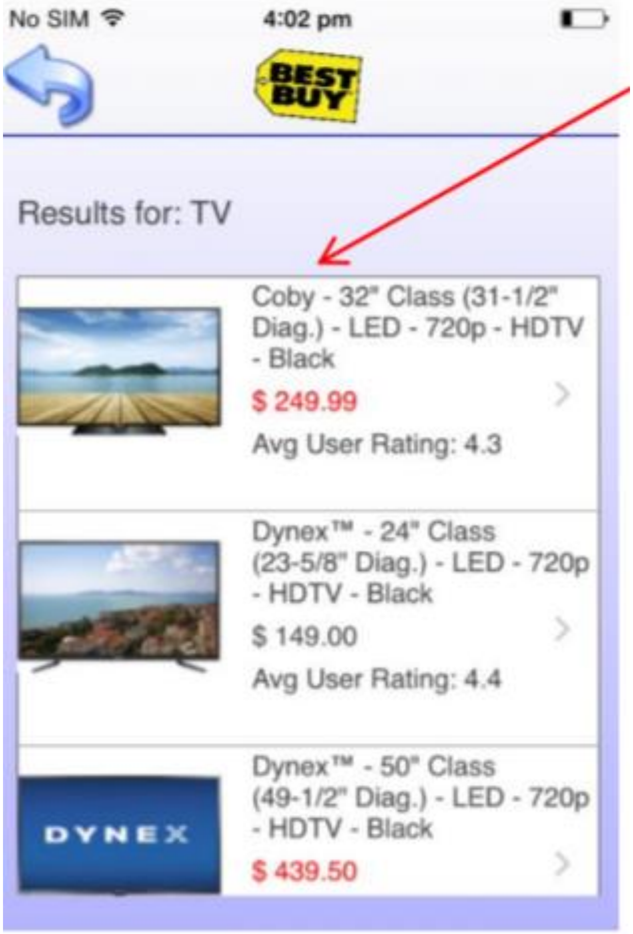
3.3 Product Form –

The product form is where the details of the forms are shown. The form consists of a header that has the best buy label. Also a back button that navigates us back to the category page. The label below shows the result for label that has the category name we selected or searched for. The next widget used is the segment again. The segment is populated with the data again which comes from the service on the Mobile Fabric, which in turn collects the data from the response of the api call we make.

The widget data map is used to bind the data we get from the service to the segment.

How we bind the data is decide by us by making temlates.

Templates are used to decide how the segment looks like. It gives labels and images that can be mapped to the data from the services.



Results for: TV

Coby - 32" Class (31-1/2" Diag.) - LED - 720p - HDTV - Black
$ 249.99
Avg User Rating: 4.3

Dynex™ - 24" Class (23-5/8" Diag.) - LED - 720p - HDTV - Black
$ 149.00
Avg User Rating: 4.4

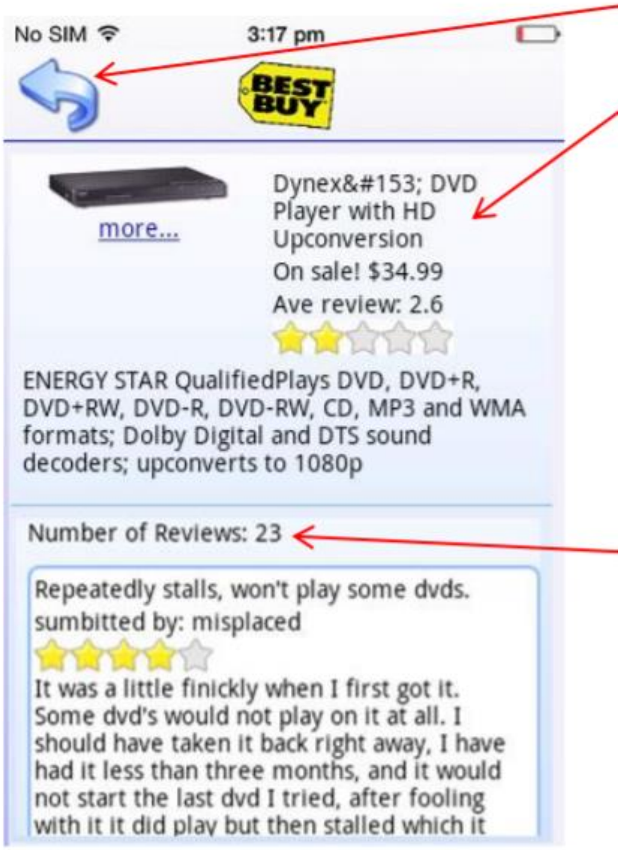
Dynex™ - 50" Class (49-1/2" Diag.) - LED - 720p - HDTV - Black
$ 439.50

Fig 3.2 Product Form

3.4 Product Details Form –

The products details form shows the complete details for a product. The form consists a header that has the best buy icon and a back icon that navigates us back to the product form. Next the flex that contains the image for the product, the rating and description.

Then we have a segment that contains the reviews. The reviews are collected from the response of the api call.



Dynex&#153; DVD Player with HD Upconversion

On sale! $34.99

Ave review: 2.6

ENERGY STAR QualifiedPlays DVD, DVD+R, DVD+RW, DVD-R, DVD-RW, CD, MP3 and WMA formats; Dolby Digital and DTS sound decoders; upconverts to 1080p

Number of Reviews: 23

Repeatedly stalls, won't play some dvds.
sumbitted by: misplaced

It was a little finickly when I first got it. Some dvd's would not play on it at all. I should have taken it back right away, I have had it less than three months, and it would not start the last dvd I tried, after fooling with it it did play but then stalled which it

Fig 3.3 Product Details

3.4.1 API used -

Get categories:

https://api.bestbuy.com/v1/categories(id=cat00000)?apiKey=YourAPIKey

Get products for a category:

https://api.bestbuy.com/v1/products(categoryPath.id=abcat0902001)?apiKey=YourAPIKey&page=1

Get products by search text:

http://api.bestbuy.com/v1/products(search=%22hdmi+cable%22)?apiKey=YourAPIKey&page=1

Get product details:

http://api.bestbuy.com/v1/products(productId=1218088854917)?apiKey=YourAPIKey

Get user reviews for a product:

http://api.bestbuy.com/v1/reviews(sku=1000671)?apiKey=YourAPIKey

You need to register (quick and free) to get an API key that needs to be passed to all API calls: https://developer.bestbuy.com/

3.5 The Middleware –

The middleware is the part where services are made and published. This is also called Mobile Fabric. The MF contains the following services–

1. Identity – This service is used to authorize logins to the app. The identity service contains logics and api's that help the user to login.
2. Integration – The integration services are ones where most of the work happens. The service contains logics and responses that help the app work.
3. Orchestration services – It is used in case we have to use two or more integration services at once. It helps to group together services and use them as one service intead of two.
4. Synchronization – it is used to map offline data.
5. Engagement – These services are used to send push messages through apps.
6. Reporting – It is used to measure app info and usage.

3.5.1 Identity Service –

Kony Fabric character administrations help you secure your application by including an authentication layer.

You can set up a personality administration dependent on the sort of the clients who are permitted to get to your application. To limit access to your organization's inner gathering of people, utilize Microsoft Active Directory confirmation. To enable access of your application to a bigger crowd, you can utilize endeavor personality suppliers (Microsoft Active Directory, Kony SAP Gateway, Open LDAP, OAuth 2.0, Salesforce, Custom Identity Service, SAML, Siteminder) and social character suppliers (Google, LinkedIn, Instagram, Amazon, Microsoft, Yahoo, BOX, Facebook.).

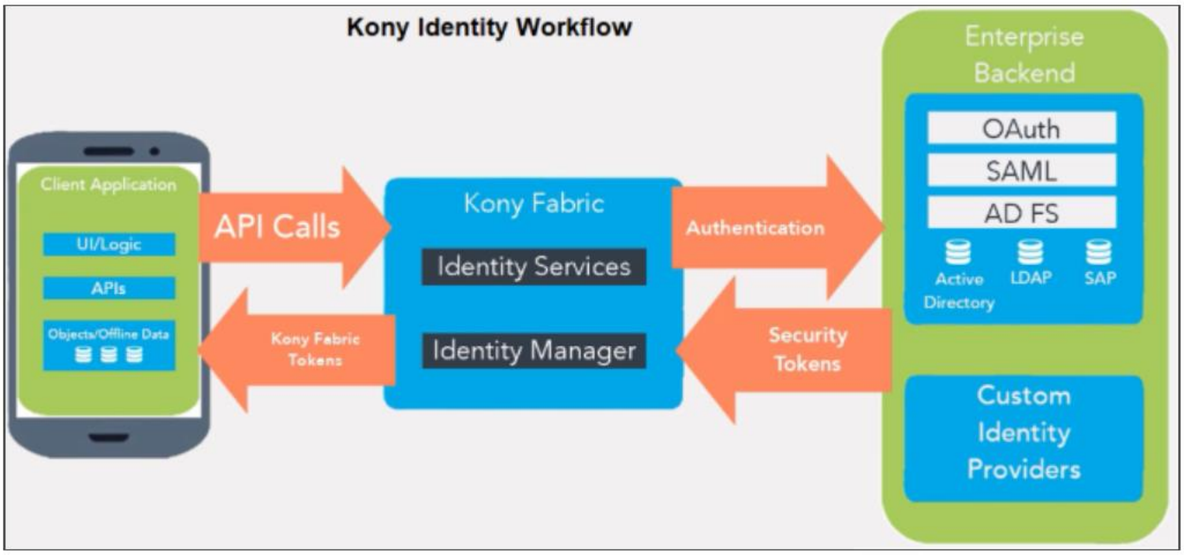
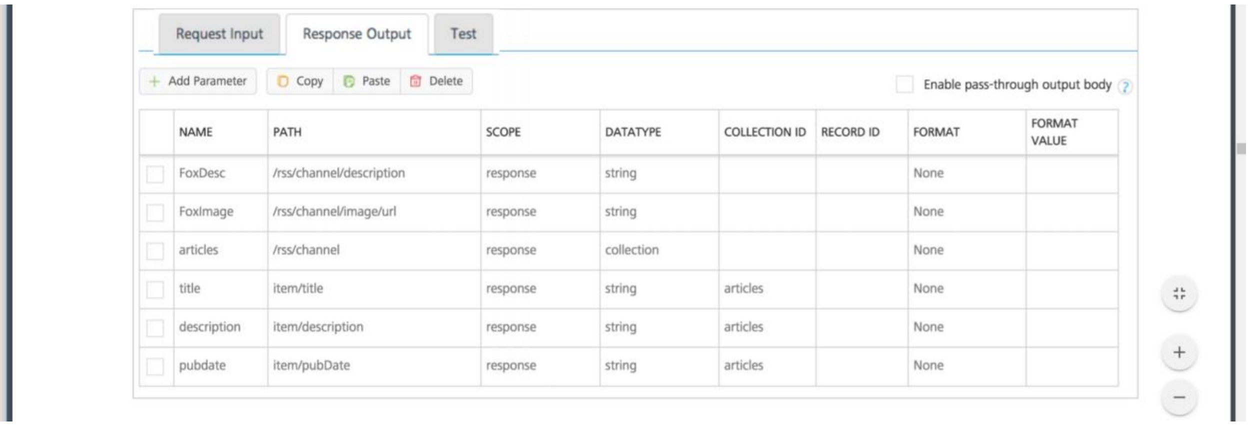


Fig 3.4 Workflow

3.5.2 Integration Services –

Moving the integration services to the middleware allows the user the following advantages –

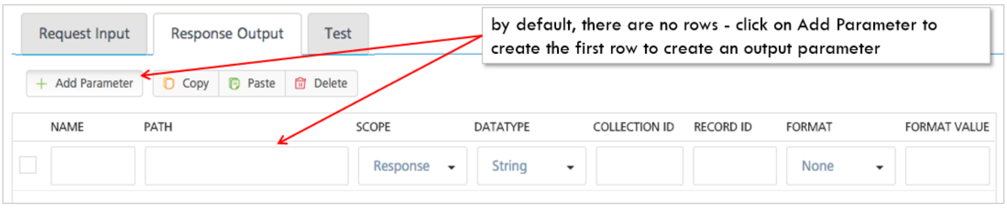
1. When the data is returned to Kony Fabric, its filtered such that only the fields you need on the mobile device are sent back to the mobile app.
2. It lets you do pre- and post- processing over the data on the server side rather than on the client side.
3. It lets you store data in fabric's cache.
4. It lets you process data before sending it to client side.
5. It lets you create a robust library of integration services.
6. Fixes can be made in fabric without affecting the mobile development.



	NAME	PATH	SCOPE	DATATYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE
□	FoxDesc	/rss/channel/description	response	string			None	
□	FoxImage	/rss/channel/image/url	response	string			None	
□	articles	/rss/channel	response	collection			None	
□	title	item/title	response	string	articles		None	
□	description	item/description	response	string	articles		None	
□	pubdate	item/pubDate	response	string	articles		None	

Fig 3.5 Fabric Response

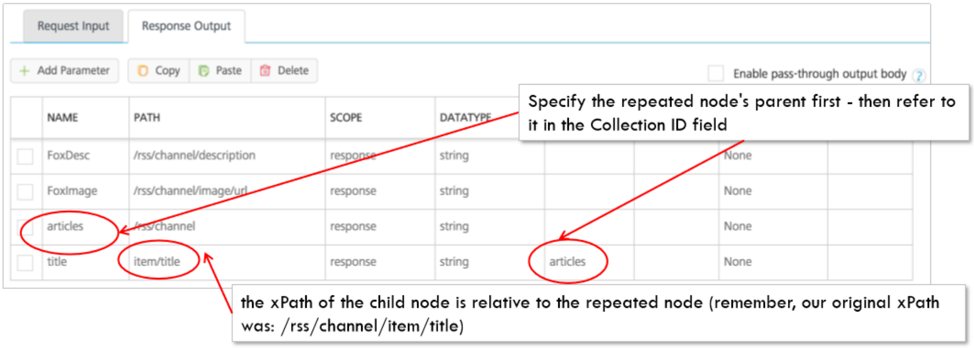
In the best buy app, we have the XML response output. After we create the services, the response we receive goes to the client side in the form of a JSON object.



by default, there are no rows - click on Add Parameter to create the first row to create an output parameter

	NAME	PATH	SCOPE	DATATYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE
□			Response	String			None	

Fig 3.6 Fabric Basics

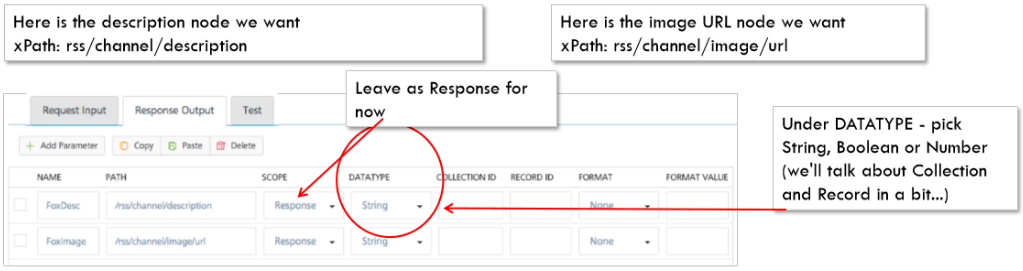


Specify the repeated node's parent first - then refer to it in the Collection ID field

	NAME	PATH	SCOPE	DATATYPE				
□	FoxDesc	/rss/channel/description	response	string			None	
□	FoxImage	/rss/channel/image/url	response	string			None	
□	articles	/rss/channel	response	string			None	
□	title	item/title	response	string	articles		None	

the xPath of the child node is relative to the repeated node (remember, our original xPath was: /rss/channel/item/title)

Fig 3.7 Fields


Here is the description node we want
xPath: rss/channel/description

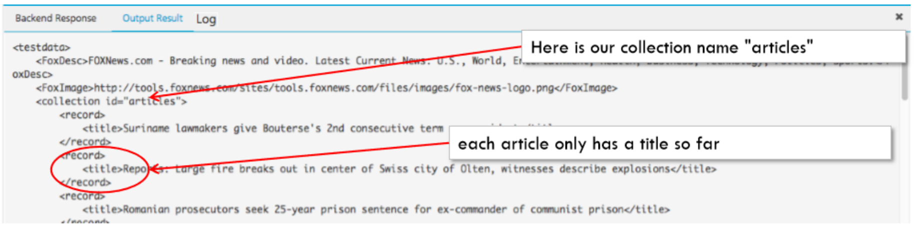
Here is the image URL node we want
xPath: rss/channel/image/url

Leave as Response for now

Under DATATYPE - pick String, Boolean or Number (we'll talk about Collection and Record in a bit...)

	NAME	PATH	SCOPE	DATATYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE
□	FoxDesc	/rss/channel/description	Response	String			None	
□	FoxImage	/rss/channel/image/url	Response	String			None	

Fig 3.8 Response Fields



Here is our collection name "articles"

each article only has a title so far

```
<testdata>
    <FoxDesc>FOXNews.com - Breaking news and video. Latest Current News: U.S., World, En
oxDesc>
    <FoxImage>http://tools.foxnews.com/sites/tools.foxnews.com/files/images/fox-news-logo.png</FoxImage>
    <collection id="articles">
        <record>
            <title>Suriname lawmakers give Bouterse's 2nd consecutive term
        </record>
        <record>
            <title>Repo  s: Large fire breaks out in center of Swiss city of Olten, witnesses describe explosions</title>
        </record>
        <record>
            <title>Romanian prosecutors seek 25-year prison sentence for ex-commander of communist prison</title>
```

Fig 3.9 XML Response

API’s to communicate with fabric

The first step deals with getting the integration service object. For this we use kony.sdk.getCurrentInstance().getIntegrationService(serviceName).

This procedure is followed when we already associated our fabric application with the visualizer application. Now, we use this object to call a particular operation.

integrationObj.invokeOperation(operationName, headers, data, this.operationSuccess, this.operationFailure) is the function which we use to call the operation corresponding to the integration service.

Parameters:

operationName- This specifies the name of the operation which we intend to call.

Data: This includes the additional data that we give or require to call the api.

operationSuccess: This function is called if the operation is executed successfully.

operationFailure: This function is called if the operation faces some failure in its execution.

Mapping API

This is used to map the output in the response object to the widget we want to. This may include mapping the text output to a particular widget and so on. We generally just mention the widget path and use the widgetDataMap function with this to map the output.

Example: this.view.segOther.widgetDataMap={lblData:”name”};

Navigation API

This API helps us in navigating from one form to another form. The general procedure is as follows:

1. Creating a navigation object. This is done using the kony.mvcapi. General format is- new kony.mvc.Navigation(formName)

2. Using the navigation object to actually navigate to the form we want.
   General format is- navigationObject.navigate(data). Data consists of the additional data we want to send to the other page.

   Now, the form to which we are navigating must contain the onNavigate function. This function handles and defines the actions that are to be performed when the form is loaded into the application.

OS API

This API deals with the system level events and is useful in getting system related information.

Kony.os.userAgent()- This gives us the information like phone name,IMEI number etc.

Kony.os.freeMemory()- This gives us information about the amount of free memory that is available.

Kony.os.deviceInfo()- This gives us information like the height , width of the screen and so on.

Segment API

This API is used to handle the data in the segment.

The general api's used are:

1. setData(object): This is used to set the data of the segment. The parameter given in the input contains the data that the segment is set to.
2. setDataAt(object): This is used to set the data of a particular row in the segment to a particular value.
3. removeAll(): This enables us to remove all the existing rows and sections in a segment.

The api also gives some important events that can be used to customize the application and meet the client's need. These events generally turn out to be very important and include:

1. selectedRowIndex: This returns the index of the row that is clicked. We generally use this to change the data or to trigger a particular event that generates important results.
2. onRowClick: This is an event triggered every time we click a particular row in the segment.
3. selectedRowItems: This returns the contents of the row that has been clicked by the user.

CHAPTER-4

4.1 SFFCU Banking App –

This is the project I am currently working on. It is a digital banking application being developed for the Signal Financial Federal Credit Union. The Project has the following components –

- Pre Login
- Accounts
- Money Movement
- Messages
- Menu

4.1.1 Pre Login –

Pre Login deals with the the following features –

1. Splash Screen - An introductory welcome screen when I launch the app on my device.
2. Device Registration – Gives the user ability to register their device to the SFFCU app.
3. Support -Has the FAQ's, terms and condition and privacy policy. Also has the contact details for the bank.
4. Locate Us – Provides the user the ability to find nearby ATM and branches.
5. Enrollment – Provides the user the ability to enroll in the online banking.
6. Login – Provides the user the ability for app users to authenticate and login to the app.

The modules I worked in this is the Locate Us Module.

4.1.2Working –

This is how the Locate Us module works –

1. From the login screen, on pressing the locations icon, I should navigate to the locations screen.

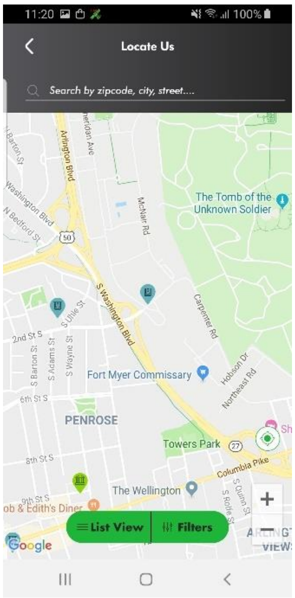
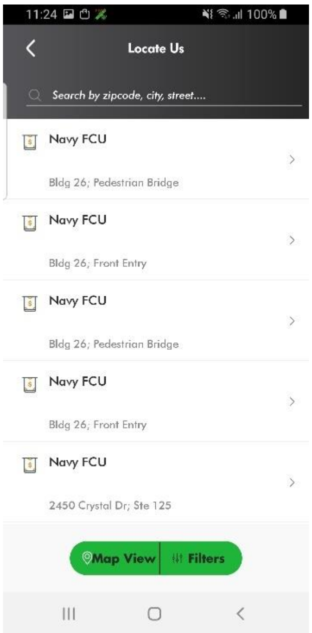


Fig 4.1

2. The locations screen looks like the one mentioned above. It contains a map, two buttons at the bottom, a search box, back button and current location button.
3. On pressing the List view button, the user should be able to see the list view of all the ATM and branches around his location.



Navy FCU
Bldg 26; Pedestrian Bridge

Navy FCU
Bldg 26; Front Entry

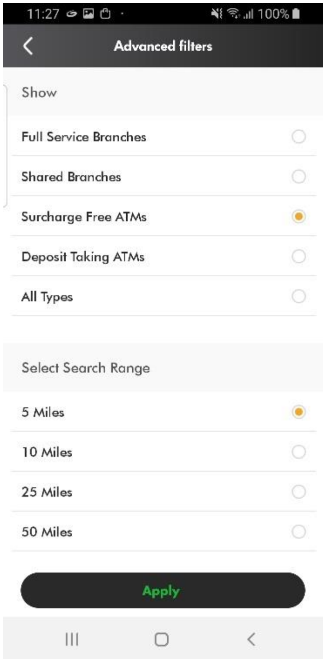
Navy FCU
Bldg 26; Pedestrian Bridge

Navy FCU
Bldg 26; Front Entry

Navy FCU
2450 Crystal Dr; Ste 125

Fig 4.2

4. This screen contains a segment which is populated using the response we get from the API call. The API call will be explained later in the section.
5. Pressing the map view takes the user back to the map view.
6. On pressing the filter icon, the Advanced filter screen is shown.



Show

- Full Service Branches ○
- Shared Branches ○
- Surcharge Free ATMs ●
- Deposit Taking ATMs ○
- All Types ○

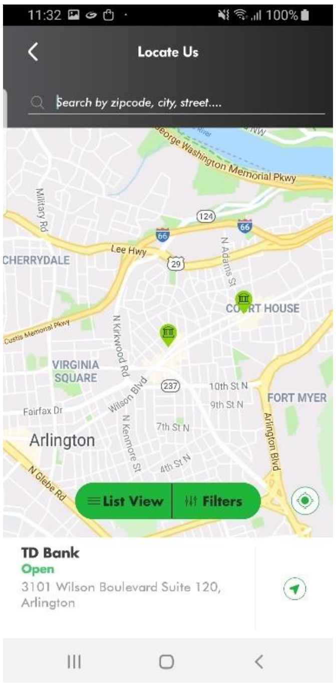
Select Search Range

- 5 Miles ●
- 10 Miles ○
- 25 Miles ○
- 50 Miles ○

Fig 4.3

This screen allows the user to to select the range and type of ATM and branches he wants to find. Based on his selection, the API call is made and only those values that fulfill the conditions are given back from the response.

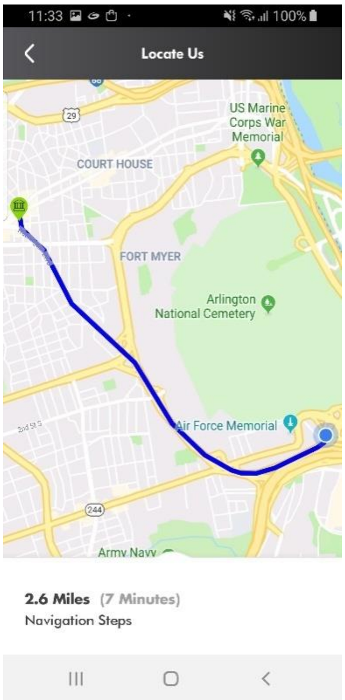
7. Pressing the icon for the ATM or Bank shows a small details tab for the location.



TD Bank
Open
3101 Wilson Boulevard Suite 120, Arlington

Fig 4.4

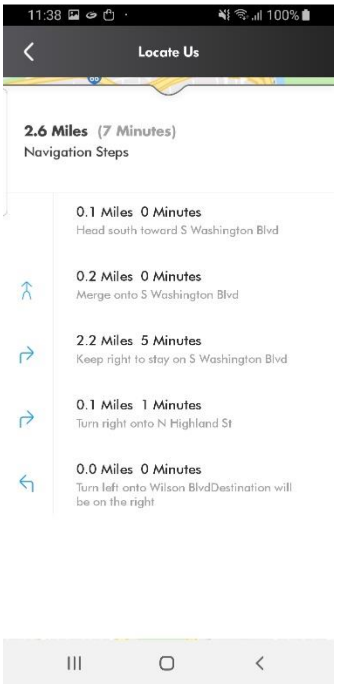
8. Pressing the navigation button to the right takes us to the Directions form. It shows the route we need to take to get to the location.



2.6 Miles (7 Minutes)
Navigation Steps

Fig 4.5

9. Dragging the navigation tab upwards will show us the specific directions we need to take.



11:38 100%

Locate Us

2.6 Miles (7 Minutes)
Navigation Steps

0.1 Miles 0 Minutes
Head south toward S Washington Blvd

0.2 Miles 0 Minutes
Merge onto S Washington Blvd

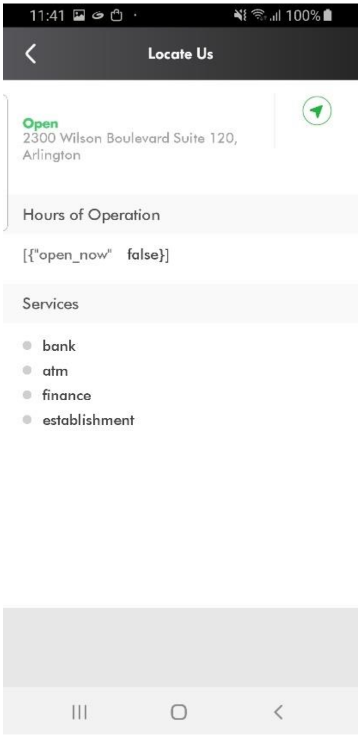
2.2 Miles 5 Minutes
Keep right to stay on S Washington Blvd

0.1 Miles 1 Minutes
Turn right onto N Highland St

0.0 Miles 0 Minutes
Turn left onto Wilson BlvdDestination will be on the right

Fig 4.6

10. On pressing on the name of the bank, we are navigated to the location details screen. The location details screen shows the address of the bank along with the timings and services provided by it. All these are also a part of the reponse from the API.



11:41 100%

Locate Us

Open
2300 Wilson Boulevard Suite 120, Arlington

Hours of Operation

[{"open_now" false}]

Services

- bank
- atm
- finance
- establishment

Fig 4.7

4.2 API Used

The API used here is not the google maps API but is called the LocatorSearch API.

4.3 LocatorSearchAPI

4.3.1 Overview –

Service Locator provides a highly intuitive and comprehensive service locator for financial institutions. There is virtually no limit to the types of products or services the Service Locator can be setup to find. Whether a customer is looking for an ATM, Branch, Shared Branch (credit unions) at the airport or any address location Service Locator is the answer. The Service Locator can be customized to meet specific branding needs so that it will integrate seamlessly into any application.

4.3.2 System Features –

-Customizable look and feel to meet your branding needs
-Service Locator can accept and map location data, including custom location features
-Multiple location search criteria options, making it easy for users who have an exact location in mind, as well as for users who may only have a general area like a postal code, city or airport in mind
-Search Engine Optimized for high search ranking with search engines like Yahoo or Google
-Supports multiple delivery mechanisms like websites, mobile-WAP, portlets, etc.

4.3.3 System Benefits –

-User friendly, flexible platform that is fast and cost-effective to implement
-Employs the latest Web 2.0 technology which provides a rich user experience
-Easily adapted to fit any website style or size configuration
-Enhancements and upgrades are automatically available to everyone using the platform
-Easily customizable to meet your specific needs

4.4 Services

It provides SOAP web services that are provided by the data warehouse. It takes latitude longitude to determine a location based on several conditions. This process is called geocoding. The locations around a person can be determined once we have the latitude and longitude.
The FindLocation services provides the ability nearest location based on latitude and longitude or a point based on radius.

Username/Password – All requests to the service require a username and password to get the response.

https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx/FindLocations?UserName=string&Password=string&AddressLine= &City= &State= &PostalCode= &Country= &Latitude= &Longitude= &Type= &Offset=

4.5 GetSearchByOptions

Returns search types that can then be used by the Find Location service.

Parameters – Username and Password.
Returns – List of search types.

Example Request –

https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx/GetSearchbyOptions?UserName=string&Password=string

Sample Response –

```
<?xml version="1.0" encoding="utf-8" ?>
<SearchOptions>
<option Name="LS Credit Union Branches" Value="FCS" />
<option Name="Surcharge Free ATMs" Value="ATMSF" />
<option Name="Deposit Taking ATMs" Value="ATMDP" />
<option Name="Shared Branches" Value="ESC" />
</SearchOptions>
```

4.6 FindLocations

Parameters –

-**AddressLine**(optional, default is no AddressLine): AddressLine of the street address, usually the street number and street. Such as "42 Elm Ave". **This field can also be used for the single line address.**
-**City** (optional): The city. – Los Angeles
-**State** (optional): The state. – CA
-**PostalCode**(optional): The Postal Code, or the ZIP Code in the USA.
-**Country** (optional): Name of Country or ISO codes.
-**Latitude** (optional): The latitude to use as the base point of the search. Usually, this latitude comes from a geocoded address or GPS device.
-**Longitude** (optional): The longitude to use as the base point of the search. Usually, this longitude comes from a geocoded address or GPS device.
-**Type** (optional, by default all location types will be returned): The Type is defined for each client in the **GetSearchByOptions**service. The standard location types are listed below. Multiple types can be passed separated by commas. The following location types are valid in addition to the types defined by the **GetSearchByOptions**web service.
1 - ATM
2 - Branch
-**Offset** (optional): Defines maximum search radius in miles (max 100).

Returns: An XML list of locations that match the search criteria.

Example(replace “string” with your provided credentials, and enter a postal code near you):

https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx/FindLocations?UserName=**string**&Password=**string**&AddressLine=&City=&State=&PostalCode=**12345**&Country=&Latitude=&Longitude=&Type=&Offset=

Example Result XML:

```
<locations locationCount="20" StartLatitude="33.14125" StartLongitude="-96.84839"
StartAddress="75034">
<location id="983819251" distanceUnit="miles" distance="3.41">
<attributes>
<attribute key="LocationID">983819251</attribute>
<attribute key="LocationType">BRANCH</attribute>
<attribute key="LocationTypeLabel">Branch</attribute>
<attribute key="LocationName">InTouch CU</attribute>
<attribute key="RetailOutlet"></attribute>
<attribute key="AddressLine">9151 Warren Pkwy</attribute>
<attribute key="CityName">Frisco</attribute>
<attribute key="StateCode">TX</attribute>
<attribute key="PostalCode">75035</attribute>
<attribute key="CountryCode">USA</attribute>
<attribute key="Latitude">33.10862</attribute>
<attribute key="Longitude">-96.80416</attribute>
<attribute key="Distance">3.41</attribute>
<attribute key="DistanceUnit">miles</attribute>
<attribute key="DistanceMeters">5495</attribute>
<attribute key="ImageURL"></attribute>
<attribute key="ImageText"></attribute>
<attribute key="MapIcon">6.png</attribute>
<attribute key="MapUrl">
http://maps.google.com/maps/api/staticmap?center=33.14125,-
96.84839&maptype=roadmap&markers=color:red%7Clabel:1%7C33.14125,-
96.84839&markers=color:red%7Ccolor:red%7Clabel:2%7C33.10862,-
96.80416&sensor=true
</attribute>
```

4.7 GetLocationList

Returns locations in the client’s own location list. The result is filtered by the type parameter.

Parameters –
Username/Password
Type(optional) –
FCS – Client’s Branches
FIATM – Client’s ATMs

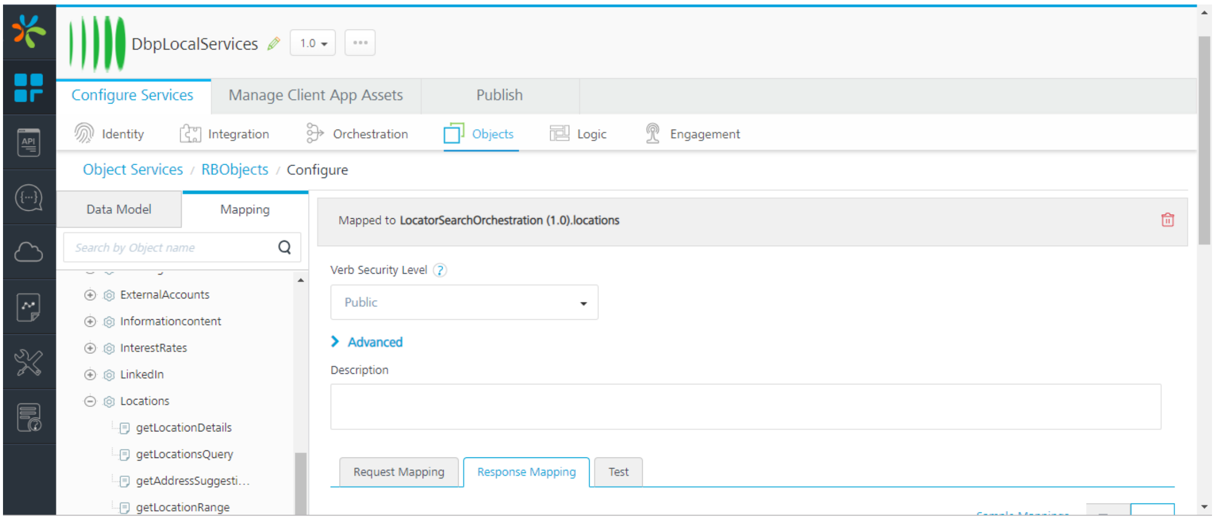
Returns – An XML list of search option types.

4.8 Error Responses –

Invalid Username/Password Or Access is restricted.
Invalid Address or Address not found.
Missing Address Entered or Address not found.
Missing Parameters.

4.9 Mobile Fabric Side –

4.9.1 Object Services–

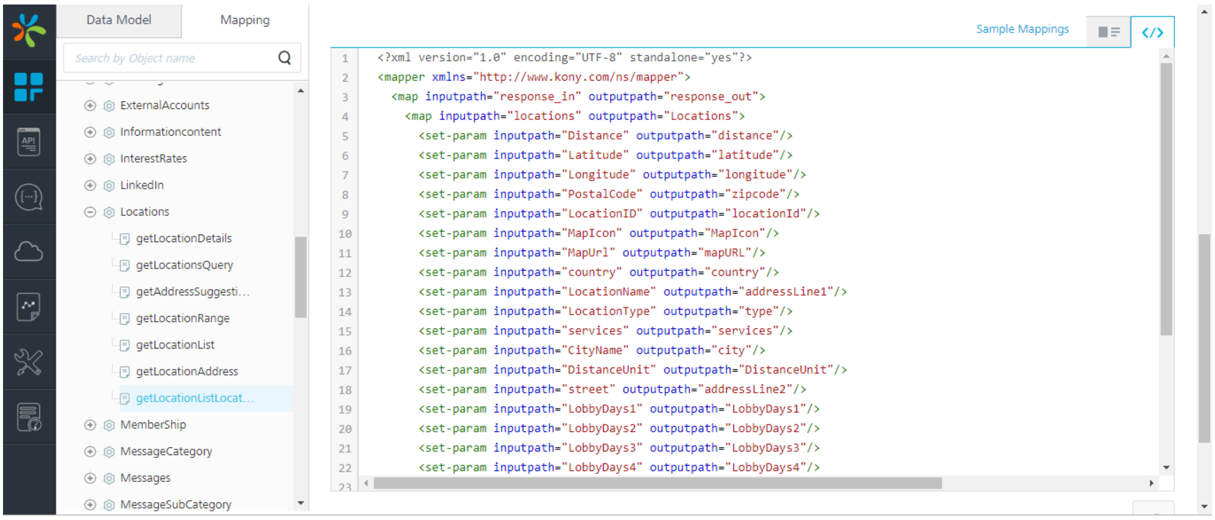


Mapped to LocatorSearchOrchestration (1.0).locations

Verb Security Level: Public

Description

Fig 4.9



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mapper xmlns="http://www.kony.com/ns/mapper">
  <map inputpath="response_in" outputpath="response_out">
    <map inputpath="locations" outputpath="Locations">
      <set-param inputpath="Distance" outputpath="distance"/>
      <set-param inputpath="Latitude" outputpath="latitude"/>
      <set-param inputpath="Longitude" outputpath="longitude"/>
      <set-param inputpath="PostalCode" outputpath="zipcode"/>
      <set-param inputpath="LocationID" outputpath="locationId"/>
      <set-param inputpath="MapIcon" outputpath="MapIcon"/>
      <set-param inputpath="MapUrl" outputpath="mapURL"/>
      <set-param inputpath="country" outputpath="country"/>
      <set-param inputpath="LocationName" outputpath="addressLine1"/>
      <set-param inputpath="LocationType" outputpath="type"/>
      <set-param inputpath="services" outputpath="services"/>
      <set-param inputpath="CityName" outputpath="city"/>
      <set-param inputpath="DistanceUnit" outputpath="DistanceUnit"/>
      <set-param inputpath="street" outputpath="addressLine2"/>
      <set-param inputpath="LobbyDays1" outputpath="LobbyDays1"/>
      <set-param inputpath="LobbyDays2" outputpath="LobbyDays2"/>
      <set-param inputpath="LobbyDays3" outputpath="LobbyDays3"/>
      <set-param inputpath="LobbyDays4" outputpath="LobbyDays4"/>
```

Fig 4.10

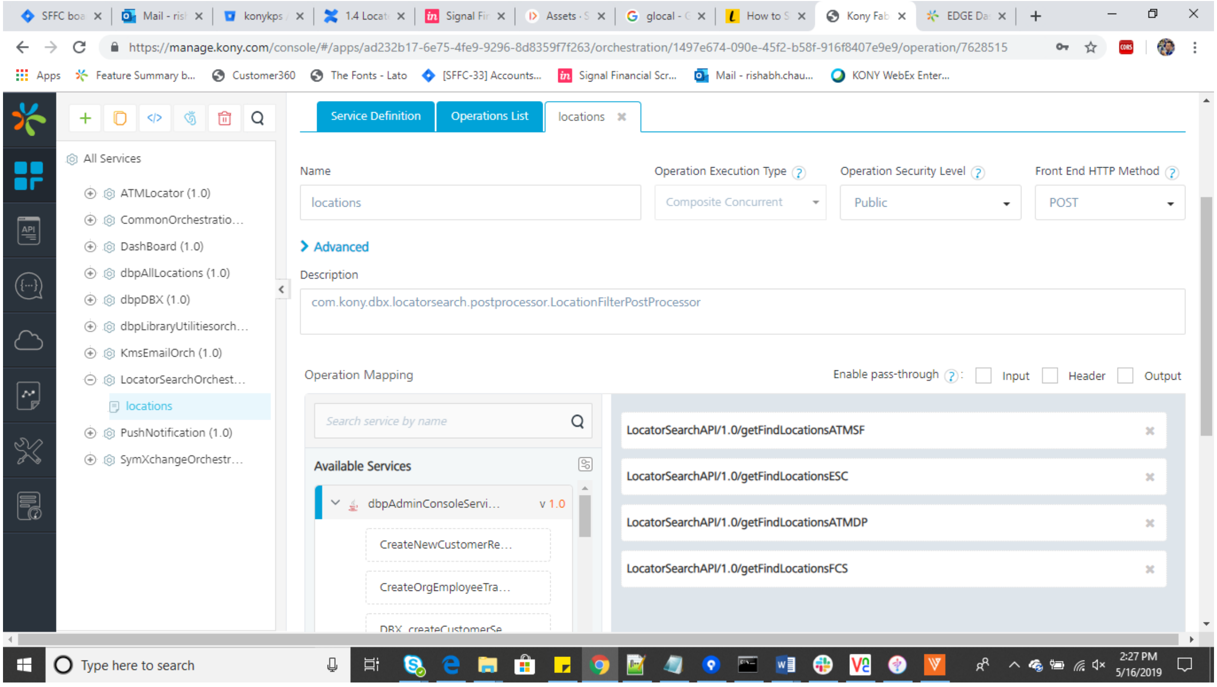
We create an object service that helps us to map data from the response to our variables. For Example – In the image above, The input path is ‘Distance’ while we map it to ‘distance’ which is what we get in the response. So when we map the data on the client side, we use the variable ‘distance’ instead of ‘Distance’.

4.9.2 Orchestration Services –

Orchestration is where we combine two or more services that we know should act as one. In this case, we are using four services in our orchestration service –

- getFindLocationsATMSF
- getFindLocationESC
- getFindLocationATMDP
- getFindLocationsFCS

We are also using a post processor to filter data we do not need from the service. The post processor can not be declared here due to the company policies.



https://manage.kony.com/console/#/apps/ad232b17-6e75-4fe9-9296-8d8359f7f263/orchestration/1497e674-090e-45f2-b58f-916f8407e9e9/operation/7628515

Name: locations

Operation Execution Type: Composite Concurrent

Operation Security Level: Public

Front End HTTP Method: POST

Description: com.kony.dbx.locatorsearch.postprocessor.LocationFilterPostProcessor

Operation Mapping

Enable pass-through: □ Input □ Header □ Output

LocatorSearchAPI/1.0/getFindLocationsATMSF

LocatorSearchAPI/1.0/getFindLocationsESC

LocatorSearchAPI/1.0/getFindLocationsATMDP

LocatorSearchAPI/1.0/getFindLocationsFCS

Fig 4.11

4.9.3 Integration Services –

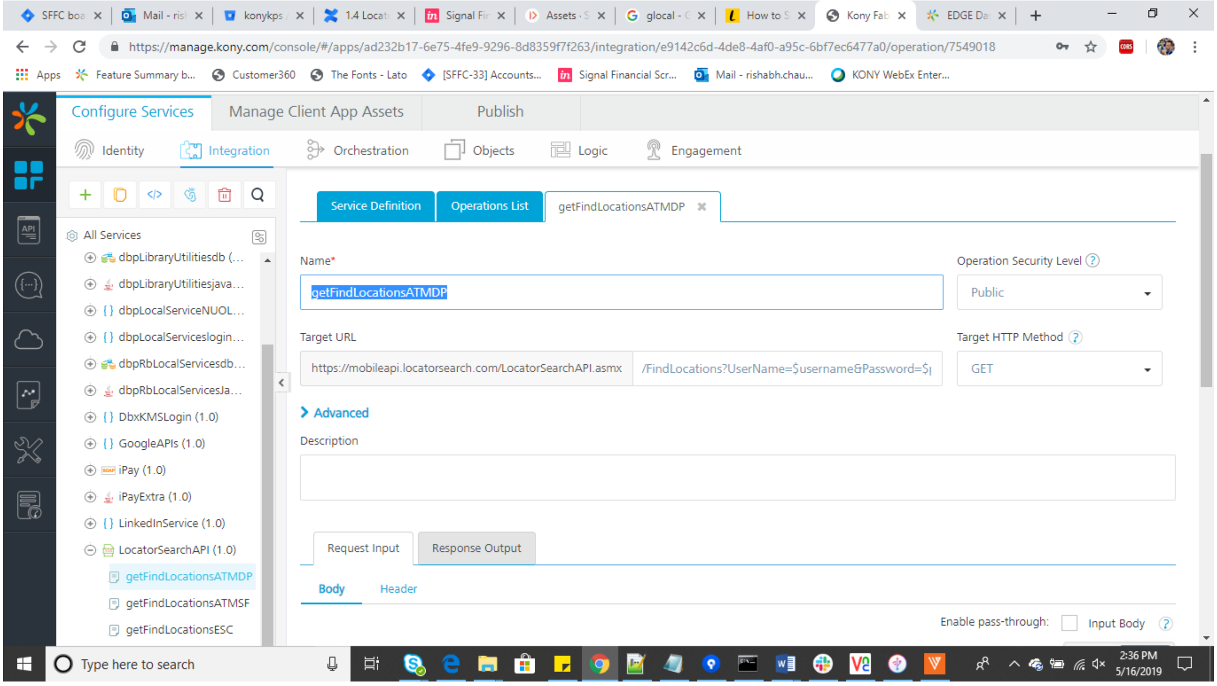
The four services mentioned above are the integration services we use in the project. These are –

getFindLocationsATMDP –

The service takes the following fields as input – username,address,city,state,password, country, latitude,longitude, type and offset.

The response output consists of – LocationType,Citycode,Statecode,PostalCode,CountryCode,Address,Email,LocationID,Distance, Distance UnitsCity,State, zip , Country, Lobby Hours ,Service.

What this service does is that it finds all the ATMs that are Deposit taking ATM and gives the details of these ATMs as part of our response.



Name*: getFindLocationsATMDP

Operation Security Level: Public

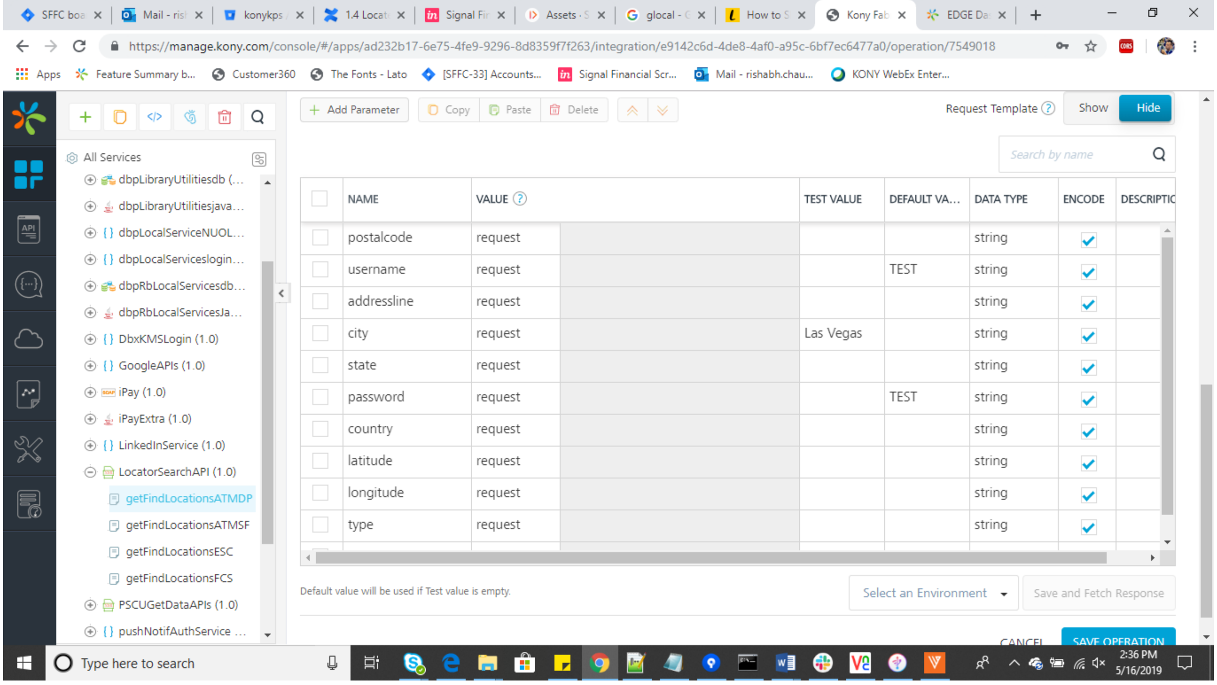
Target URL: https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx /FindLocations?UserName=$username&Password=$|

Target HTTP Method: GET

Description:

Enable pass-through: □ Input Body

Fig 4.12



□	NAME	VALUE		TEST VALUE	DEFAULT VA...	DATA TYPE	ENCODE	DESCRIPTIC
□	postalcode	request				string	✓	
□	username	request			TEST	string	✓	
□	addressline	request				string	✓	
□	city	request		Las Vegas		string	✓	
□	state	request				string	✓	
□	password	request			TEST	string	✓	
□	country	request				string	✓	
□	latitude	request				string	✓	
□	longitude	request				string	✓	
□	type	request				string	✓	

Default value will be used if Test value is empty.

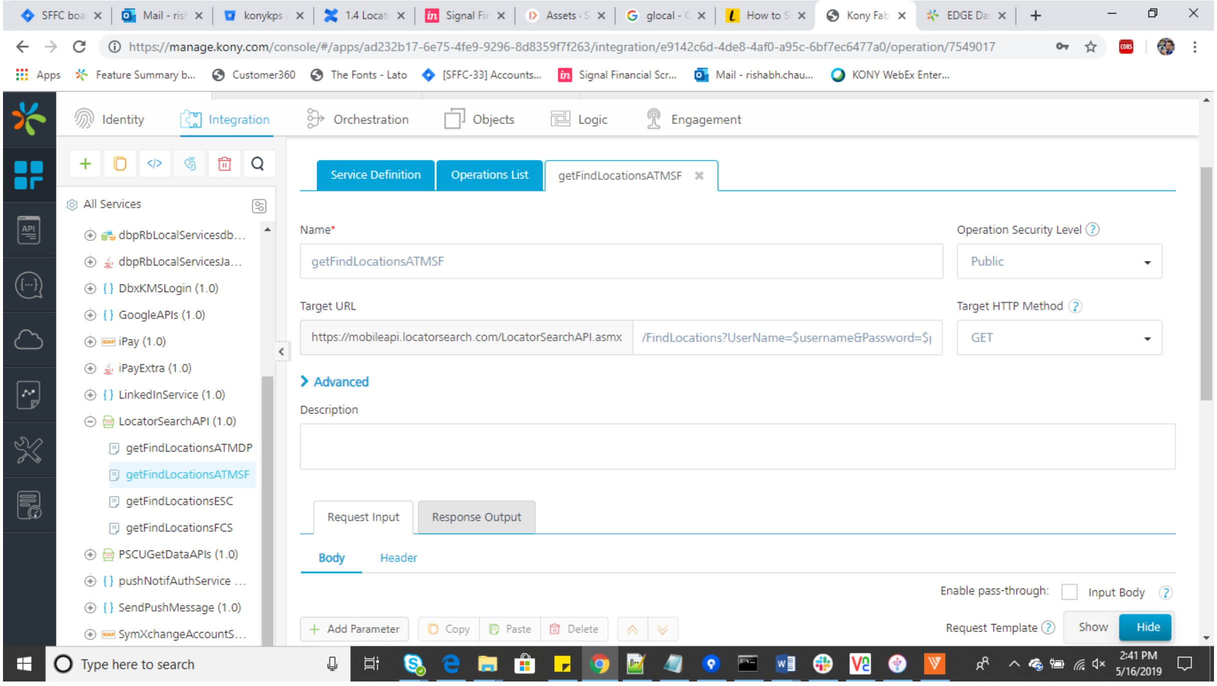
Fig 4.13

getFindLocationsATMSF –

The service takes the following fields as input – username,address,city,state,password, country, latitude,longitude, type and offset.

The response output consists of – LocationType,Citycode,Statecode,PostalCode,CountryCode,Address,Email,LocationID,Distance, Distance UnitsCity,State, zip , Country, Lobby Hours ,Service.

What this service does is that it finds all the ATMs that are Surcharge Free ATMs and gives the details of those ATMs as part of our response.



https://manage.kony.com/console/#/apps/ad232b17-6e75-4fe9-9296-8d8359f7f263/integration/e9142c6d-4de8-4af0-a95c-6bf7ec6477a0/operation/7549017

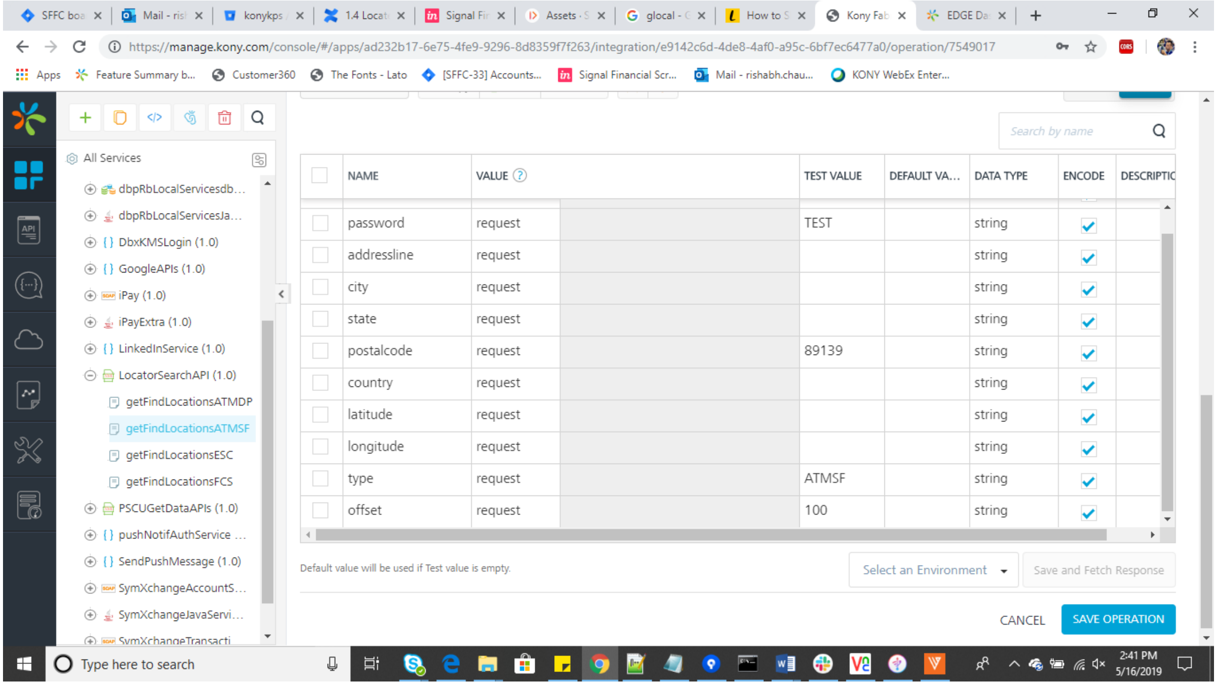
Name*: getFindLocationsATMSF

Operation Security Level: Public

Target URL: https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx /FindLocations?UserName=$username&Password=$|

Target HTTP Method: GET

Fig 4.14



	NAME	VALUE		TEST VALUE	DEFAULT VA...	DATA TYPE	ENCODE	DESCRIPTIC
□	password	request		TEST		string	✓	
□	addressline	request				string	✓	
□	city	request				string	✓	
□	state	request				string	✓	
□	postalcode	request		89139		string	✓	
□	country	request				string	✓	
□	latitude	request				string	✓	
□	longitude	request				string	✓	
□	type	request		ATMSF		string	✓	
□	offset	request		100		string	✓	

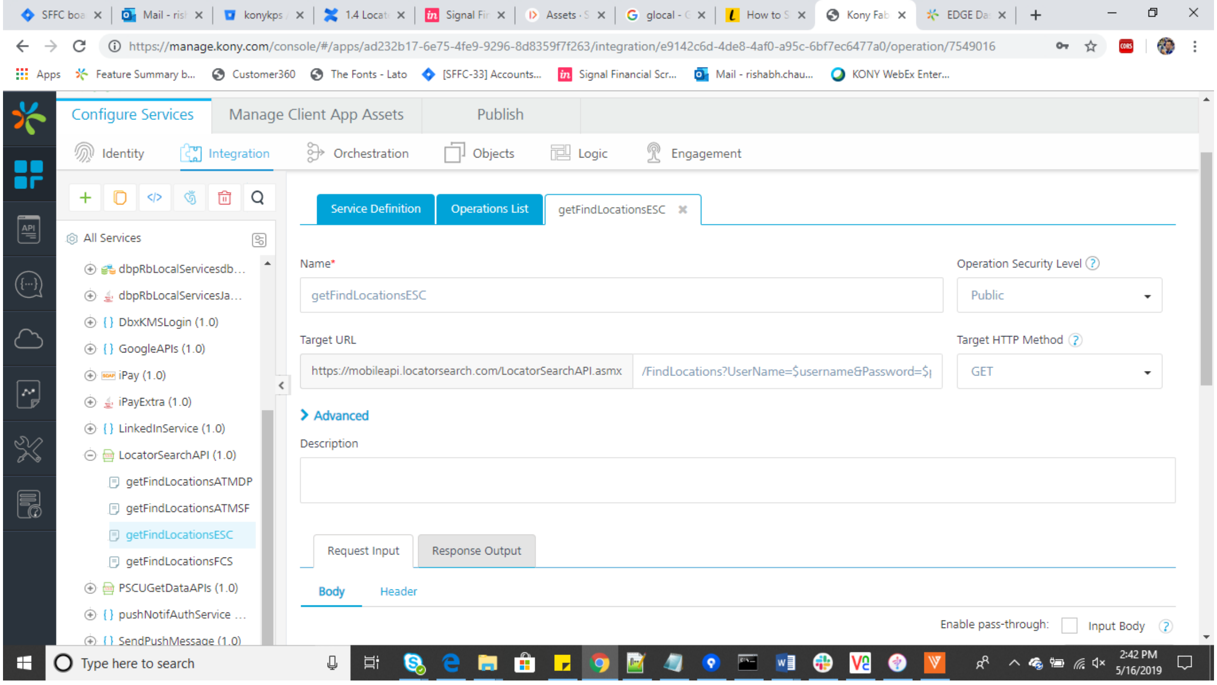
Default value will be used if Test value is empty.

Fig 4.15

getFindLocationESC –

The service takes the following fields as input – username,address,city,state,password, country, latitude,longitude, type and offset.
The response output consists of – LocationType,Citycode,State code, Postal Code,CountryCode,Address,Email,LocationID,Distance, Distance Units, City, State, zip , Country, Lobby Hours ,Service.

What this service does is it finds all the shared branches in the 10 mile. The details of the branches are given back to us as part of the response.



https://manage.kony.com/console/#/apps/ad232b17-6e75-4fe9-9296-8d8359f7f263/integration/e9142c6d-4de8-4af0-a95c-6bf7ec6477a0/operation/7549016

Name* getFindLocationsESC

Operation Security Level: Public

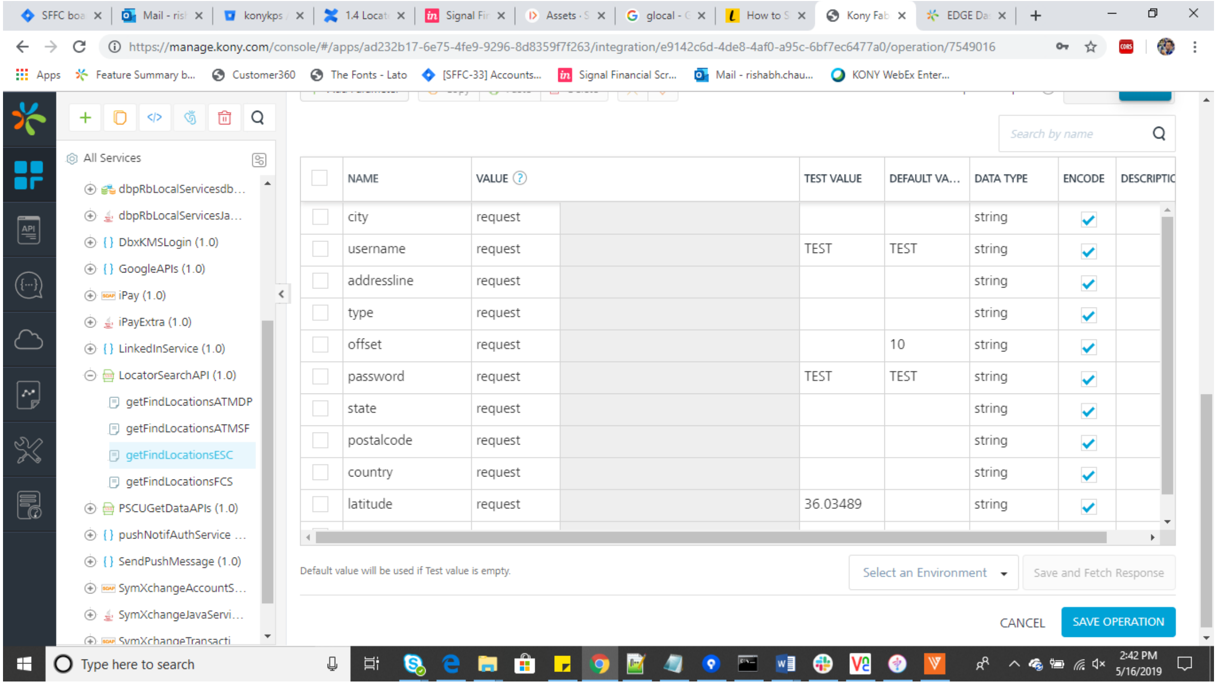
Target URL: https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx /FindLocations?UserName=$username&Password=$|

Target HTTP Method: GET

Description:

Enable pass-through: □ Input Body

Fig 4.16



https://manage.kony.com/console/#/apps/ad232b17-6e75-4fe9-9296-8d8359f7f263/integration/e9142c6d-4de8-4af0-a95c-6bf7ec6477a0/operation/7549016

□	NAME	VALUE		TEST VALUE	DEFAULT VA...	DATA TYPE	ENCODE	DESCRIPTIC
□	city	request				string	☑	
□	username	request		TEST	TEST	string	☑	
□	addressline	request				string	☑	
□	type	request				string	☑	
□	offset	request			10	string	☑	
□	password	request		TEST	TEST	string	☑	
□	state	request				string	☑	
□	postalcode	request				string	☑	
□	country	request				string	☑	
□	latitude	request		36.03489		string	☑	

Default value will be used if Test value is empty.

Select an Environment | Save and Fetch Response

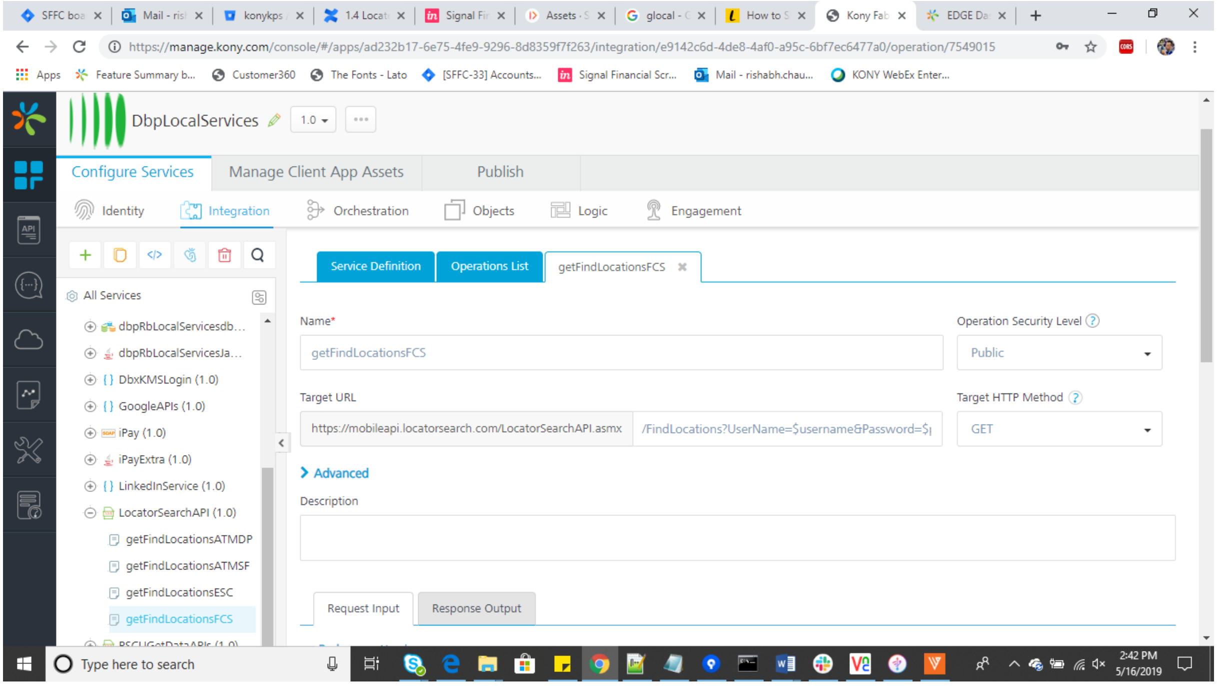
CANCEL | SAVE OPERATION

Fig 4.17

getFindLocationFCS –

The service takes the following fields as input – username,address,city,state,password, country, latitude,longitude, type and offset.
The response output consists of – LocationType,Citycode,State code, Postal Code,CountryCode,Address,Email,LocationID,Distance, Distance Units, City, State, zip , Country, Lobby Hours ,Service.

What this service does is it finds all the LS Credit Union Branches in the 10 mile radius around your location. The details of these banks are given back to us as the response.



Name*: getFindLocationsFCS

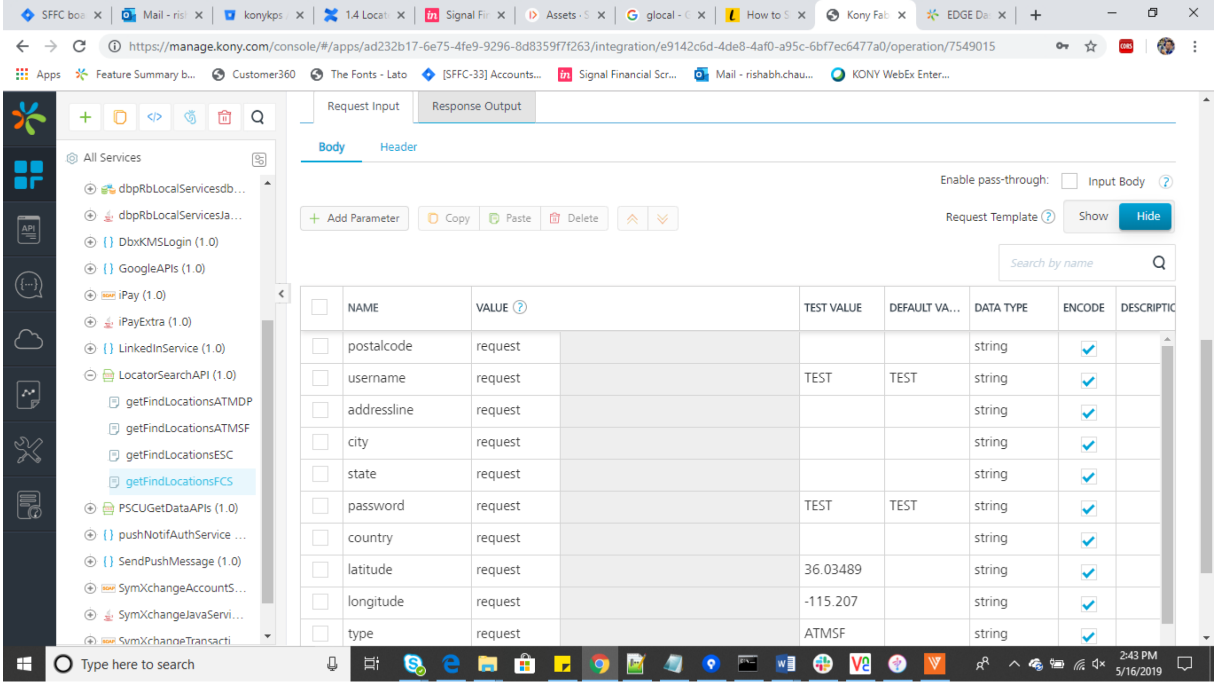
Operation Security Level: Public

Target URL: https://mobileapi.locatorsearch.com/LocatorSearchAPI.asmx /FindLocations?UserName=$username&Password=$|

Target HTTP Method: GET

Description:

Fig 4.18



Enable pass-through: □ Input Body

Request Template: Show Hide

□	NAME	VALUE		TEST VALUE	DEFAULT VA...	DATA TYPE	ENCODE	DESCRIPTIC
□	postalcode	request				string	✓	
□	username	request		TEST	TEST	string	✓	
□	addressline	request				string	✓	
□	city	request				string	✓	
□	state	request				string	✓	
□	password	request		TEST	TEST	string	✓	
□	country	request				string	✓	
□	latitude	request		36.03489		string	✓	
□	longitude	request		-115.207		string	✓	
□	type	request		ATMSF		string	✓	

Fig 4.19

CHAPTER-5

PERFORMANCE ANALYSIS AND TESTING

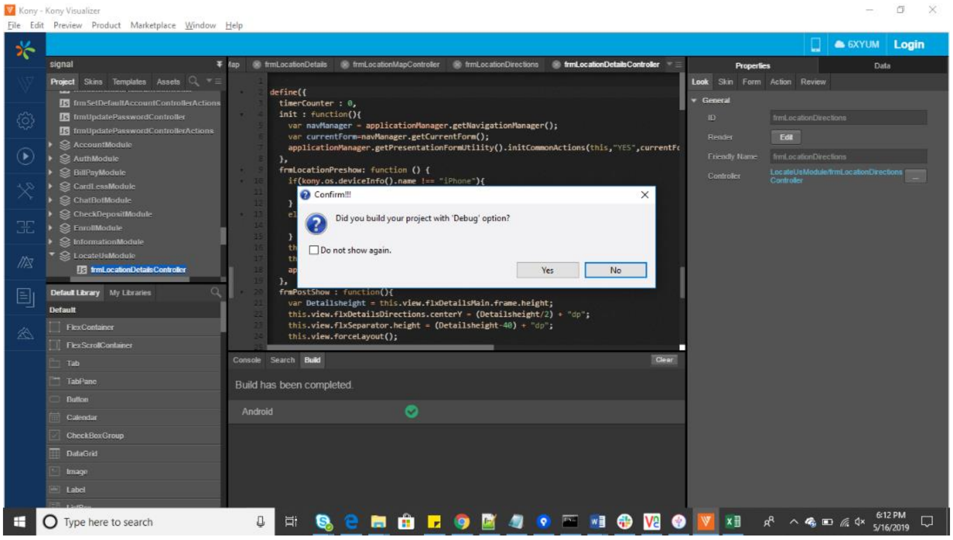
Kony provides tools that support debugging applications for troubleshooting purposes.
The debugger allows us to –

-Setting breakpoint.
-Suspending applications
-Going through your code step by step.
-Examining the variables and what they store in them.

To build an application in degub mode is required in order to debug the app. It provides debugging for both IOS and android. We can use the Google Chrome Debugger to debug. You can view more at https://developers.google.com/web/tools/chrome-devtools/javascript/ .

5.1 Debugging on Android-

-Build the app in debug mode.
-In the product menu in the visualizer, select build and select the mode as debug.
-Install the .apk file.
-Start the app on the device.
-Go to visualizer, on the product tab, select debug as -> android application.
-On the device, we get an prompt to start debugger.
-You'll get a URL to start debugging.
-Open the URL and start your app.
-You can now start your debugging.



```
define({
  timerCounter : 0,
  init : function(){
    var navManager = applicationManager.getNavigationManager();
    var currentForm=navManager.getCurrentForm();
    applicationManager.getPresentationFormUtility().initCommonActions(this,"YES",currentFo
  },
  frmLocationPreshow: function () {
    if(kony.os.deviceInfo().name !== "iPhone"){
```

```
  },
  frmPostShow : function(){
    var Detailsheight = this.view.flxDetailsMain.frame.height;
    this.view.flxDetailsDirections.centerY = (Detailsheight/2) + "dp";
    this.view.flxSeparator.height = (Detailsheight-40) + "dp";
    this.view.forceLayout();
```

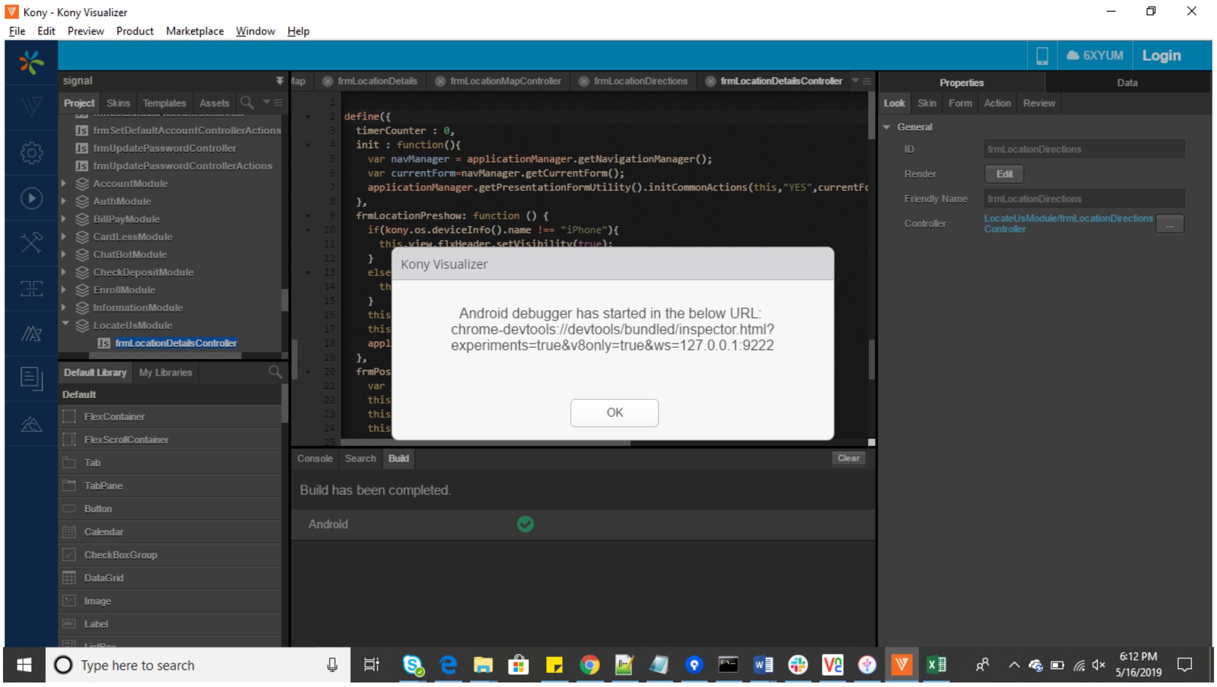
Confirm!!!

Did you build your project with 'Debug' option?

□ Do not show again.

Build has been completed.

Android ✓



```
define({
  timerCounter : 0,
  init : function(){
    var navManager = applicationManager.getNavigationManager();
    var currentForm=navManager.getCurrentForm();
    applicationManager.getPresentationFormUtility().initCommonActions(this,"YES",currentFo
  },
  frmLocationPreshow: function () {
    if(kony.os.deviceInfo().name !== "iPhone"){
      this.view.flxHeader.setVisibility(true);
```

Kony Visualizer

Android debugger has started in the below URL:
chrome-devtools://devtools/bundled/inspector.html?experiments=true&v8only=true&ws=127.0.0.1:9222

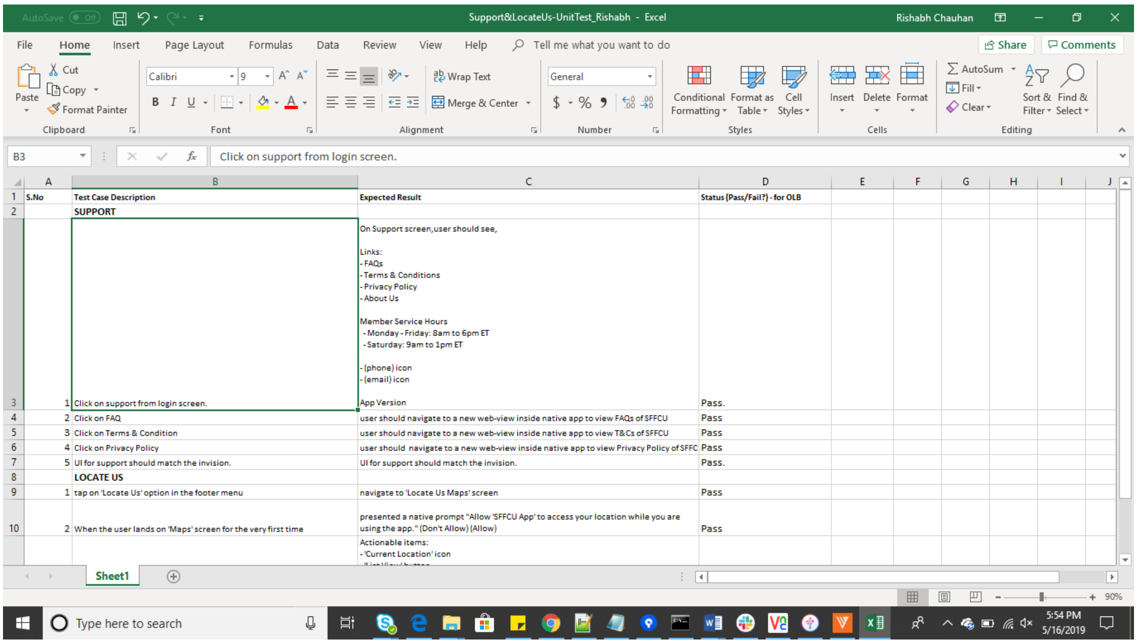
Build has been completed.

Android ✓

5.2 Testing –

We are encouraged to run unit test cases on our project before sending them ahead for more detailed testing by the testing team. We are supposed to write our own unit test cases and test them on the app. Any unit test casing failing should be added as a defect on the team page and should be assigned to the person responsible for the development of that module or screen.

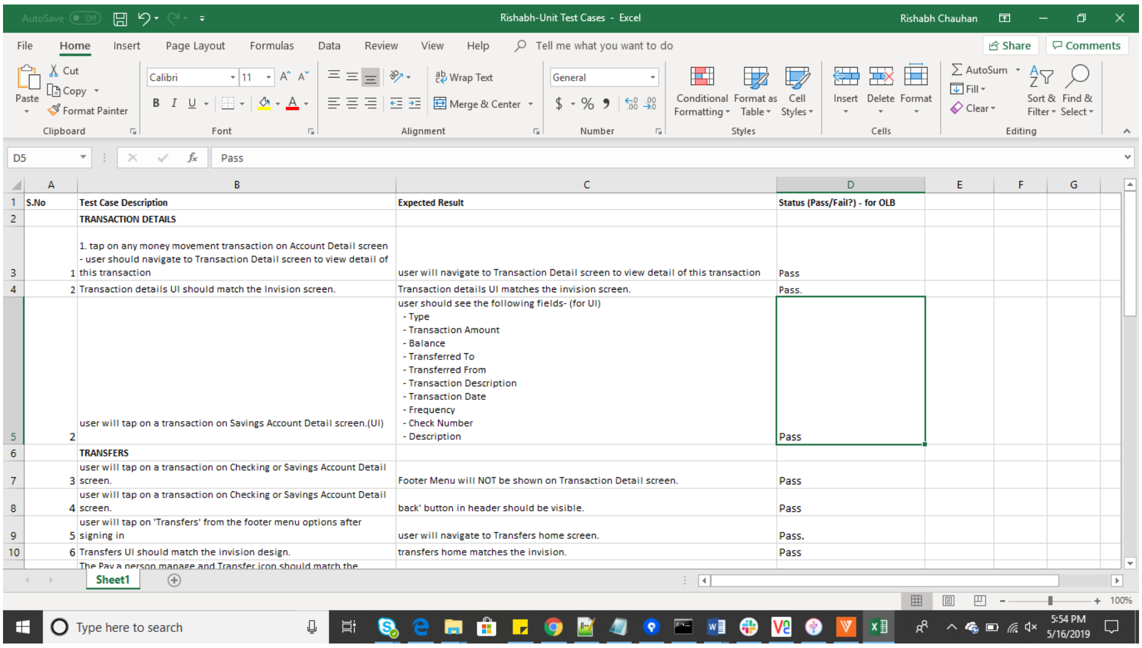
The following images show the test cases written and run by me.



Support&LocateUs-UnitTest_Rishabh - Excel

B3: Click on support from login screen.

S.No	Test Case Description	Expected Result	Status (Pass/Fail?) - for OLB
	SUPPORT		
1	Click on support from login screen.	On Support screen,user should see, Links: - FAQs - Terms & Conditions - Privacy Policy - About Us Member Service Hours - Monday - Friday: 8am to 6pm ET - Saturday: 9am to 1pm ET - (phone) icon - (email) icon App Version	Pass.
2	Click on FAQ	user should navigate to a new web-view inside native app to view FAQs of SFFCU	Pass
3	Click on Terms & Condition	user should navigate to a new web-view inside native app to view T&Cs of SFFCU	Pass
4	Click on Privacy Policy	user should navigate to a new web-view inside native app to view Privacy Policy of SFFC	Pass
5	UI for support should match the invision.	UI for support should match the invision.	Pass.
	LOCATE US		
1	tap on 'Locate Us' option in the footer menu	navigate to 'Locate Us Maps' screen	Pass
2	When the user lands on 'Maps' screen for the very first time	presented a native prompt "Allow 'SFFCU App' to access your location while you are using the app." (Don't Allow) (Allow)	Pass
		Actionable items: - 'Current Location' icon	



Rishabh-Unit Test Cases - Excel

D5: Pass

S.No	Test Case Description	Expected Result	Status (Pass/Fail?) - for OLB
	TRANSACTION DETAILS		
1	1. tap on any money movement transaction on Account Detail screen - user should navigate to Transaction Detail screen to view detail of this transaction	user will navigate to Transaction Detail screen to view detail of this transaction	Pass
2	Transaction details UI should match the Invision screen.	Transaction details UI matches the invision screen.	Pass.
2	user will tap on a transaction on Savings Account Detail screen.(UI)	user should see the following fields- (for UI) - Type - Transaction Amount - Balance - Transferred To - Transferred From - Transaction Description - Transaction Date - Frequency - Check Number - Description	Pass
	TRANSFERS		
3	user will tap on a transaction on Checking or Savings Account Detail screen.	Footer Menu will NOT be shown on Transaction Detail screen.	Pass
4	user will tap on a transaction on Checking or Savings Account Detail screen.	back' button in header should be visible.	Pass
5	user will tap on 'Transfers' from the footer menu options after signing in	user will navigate to Transfers home screen.	Pass.
6	Transfers UI should match the invision design.	transfers home matches the invision.	Pass
	The Pay a person manage and Transfer icon should match the		

These are a few unit test cases reported by me.

After these tests are done, the app binary is sent to the testing team for evaluation. Detailed testing is done on the app and the defects found are assigned to the developers for correction. There is a allowed percentage of errors that can be let through. But anything above that is to be fixed before the next sprint. Once fixed, the particular bug is then reopened for testing. If anything else is found, it can again be assigned to the developer, else it is closed and the module is considered to be ready.

The following is a screenshot of the bugs raised against me by the tester.

CHAPTER – 6

LITERATURE SURVEY

For the development of these applications , the basic understanding of the various application programming interface and the methodologies required to develop an application and other minute details have been gained from the following documents.

6.1Kony Documentation

Documentation is a lot of records given on paper, or on the web, or on computerized or simple media, for example, sound tape or CDs. Models are client guides, white papers, on-line help, brisk reference guides. It is winding up less basic to see paper (printed version) documentation. Documentation is appropriated by means of sites, programming items, and other on-line applications.

This is a set of documents which is regularly updated by the product team with every new release of the Kony visualizer version, Kony fabric version or Kony DBX version. The documentation for Kony Visualizer documentation is hosted at https://basecamp.kony.com/s/quantum-documentation?index=0. This document contains wide variety of information which includes information from installation guide of the visualizer to the api guide which is the necessary tool in the process of development of any application on the platform.

Similarly, for the other products like Kony DBX documentation is hosted on the https://konydocs.atlassian.net/wiki/spaces/ARBD41/pages/935691289/Digital+Banking+Solution+Overview website. This documentation contains information on Introduction, DBX App Suite, Digital Banking Platform, Data Storage, Open Banking and PSD2, Open Integration, UI / UX and Data Model Design, Extensibility and Customization, Enterprise Security and GDPR Readiness. In addition, there is also information about the Digital Banking Platform. The Kony Digital Banking Platform provides comprehensive assets to help the whole life. Kony Visualizer enables low code enhancement and the ability to share and use regular parts effortlessly, incorporating those provided by the Digital Banking Suite through both open and private marketplaces.Visualizer is used to expand the standard suite as well as to create custo m apps. Kony Fabric provides the board with the versatile runtime of middleware, reconciliation administrations, outsider programming connectors, identity and API.

6.2 Kony Boot Camp

This is the go-to website for all the doubts and the problems that a developer face. This is an inquiry and answer site for expert and aficionado software engineers. The domain is https://basecamp.kony.com/s/topic/0TO6A0000001I9ZWAU/developer-bootcamp?.

CHAPTER – 7

CONCLUSIONS

Working at Kony Private Ltd. has been as fun and productive as it has been working on these projects. Best buy and SFFCU Application are equally significant in their own places. The former provides an insight into the e-commerce side of things while the later provides an insight into the digital banking experience.

SFFCU has been the most extensive application of the two as it processes complex and heavy server requests. The application needs more modifications for increasing efficiency. Also, there are certain bugs which need to be dealt with to make it deployable to the Signal Finance Federal Credit Union. After the aforementioned changes, the application will go through an extensive modular and system testing to make it work at it's best efficiency.

Overall, development of these applications has been a really learning experience in terms of writing legible, comprehensible and maintainable code, debugging to an extent that each and every module knows how to handle exceptional situations, creating a UI that is not just good to look but is equally good to use and work with.

There are indeed places which need extemporization and diligence to reach the mark of perfection and excellence. I've been working hard to reach there and will confidently accomplish it some day. I will keep these precious learnings with me forever and hope to flourish more with Kony.

REFERENCES

1. https://basecamp.kony.com/s/
2. https://basecamp.kony.com/s/dbx
3. https://konydocs.atlassian.net/wiki/spaces/ARBD/pages/950906415/Digital+Banking+Platform
4. https://konydocs.atlassian.net/wiki/spaces/ARBD/pages/950906758/Retail+Banking