

# Mobile Recharge Testing....

U.Nanda kumar reddy  
192111339  
CSA3732  
Software testing  
for android applications

# CONTENTS

- TITLE
- OBJECTIVE
- ABSTRACT
- PROPOSED SYSTEM
- FLOW CHART
- CONCEPT MAP
- NO. OF TEST CASES
- APP TOOLS AND INSTALLATION
- IMPLEMENTATION TESTING SCREENSHOTS
- TEST CASE OUTCOMES
- TEST SCENARIOS
- CONCLUSION

# OBJECTIVE

- Test the mobile recharge by implementing practical , efficiency and cost
- Functionality checking using Appium software

# ABSTRACT

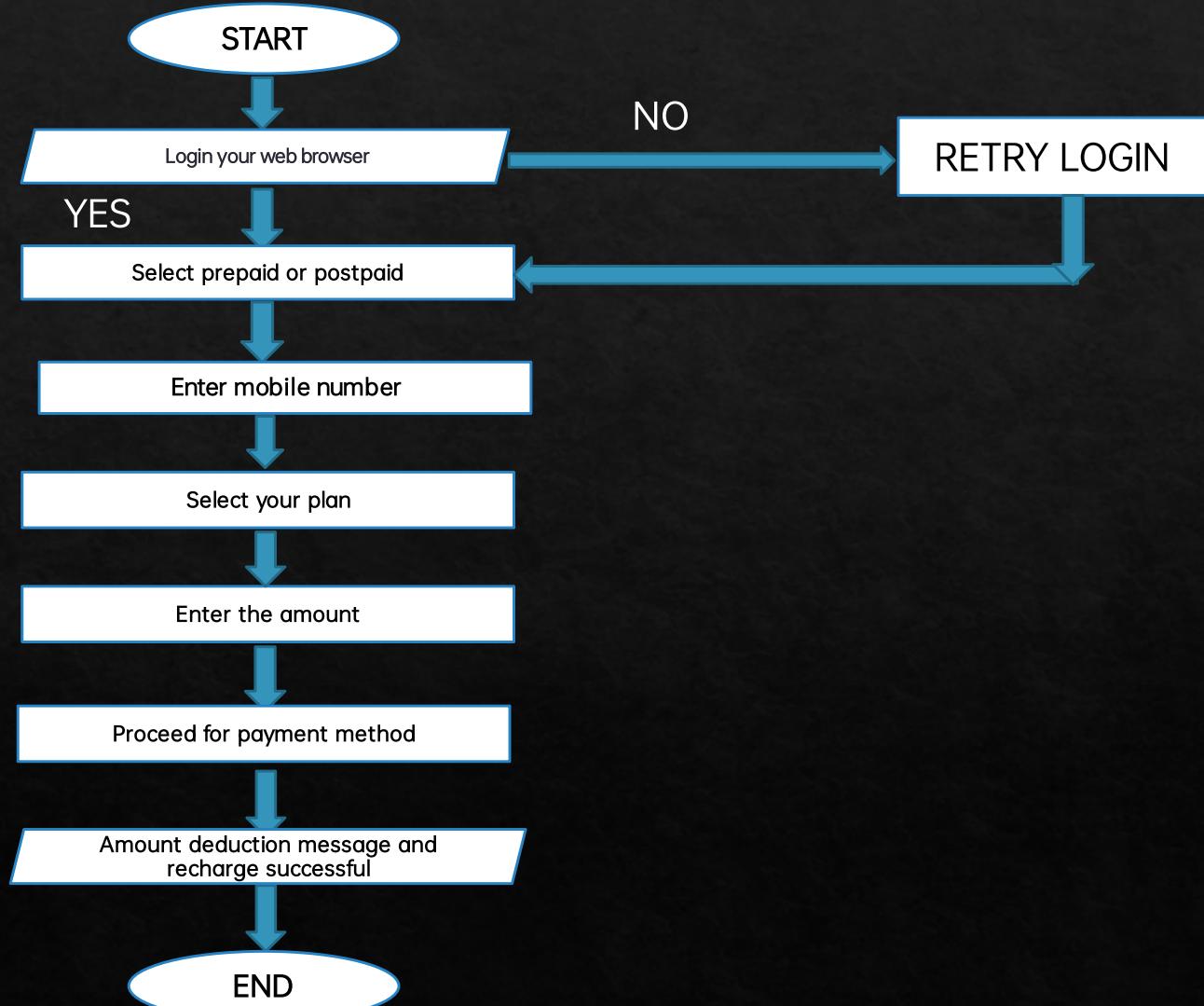
- ❖ Online Mobile Recharge is a web-based application developed in ASP.NET to recharge mobile phones. The project focuses at providing an easy and reliable platform to recharge mobile of any telecocoman company through online without buying recharge card.
- ❖ The registered users of the system can recharge their prepaid mobile phones from anywhere at any time. The proposed project for recharging mobiles developed to automate the mobile recharging process. It roots out the manual card system of recharge and introduces a new and genuine online recharge process.
- ❖ The online mobile recharge system is beneficial to both the admins and users. Using the online application, the admin can add new operators, tariff plans, offers and update or modify the existing tariff plans. It helps the users in creating their account, and then recharging the mobiles phones at any time.

# PROPOSED WORK

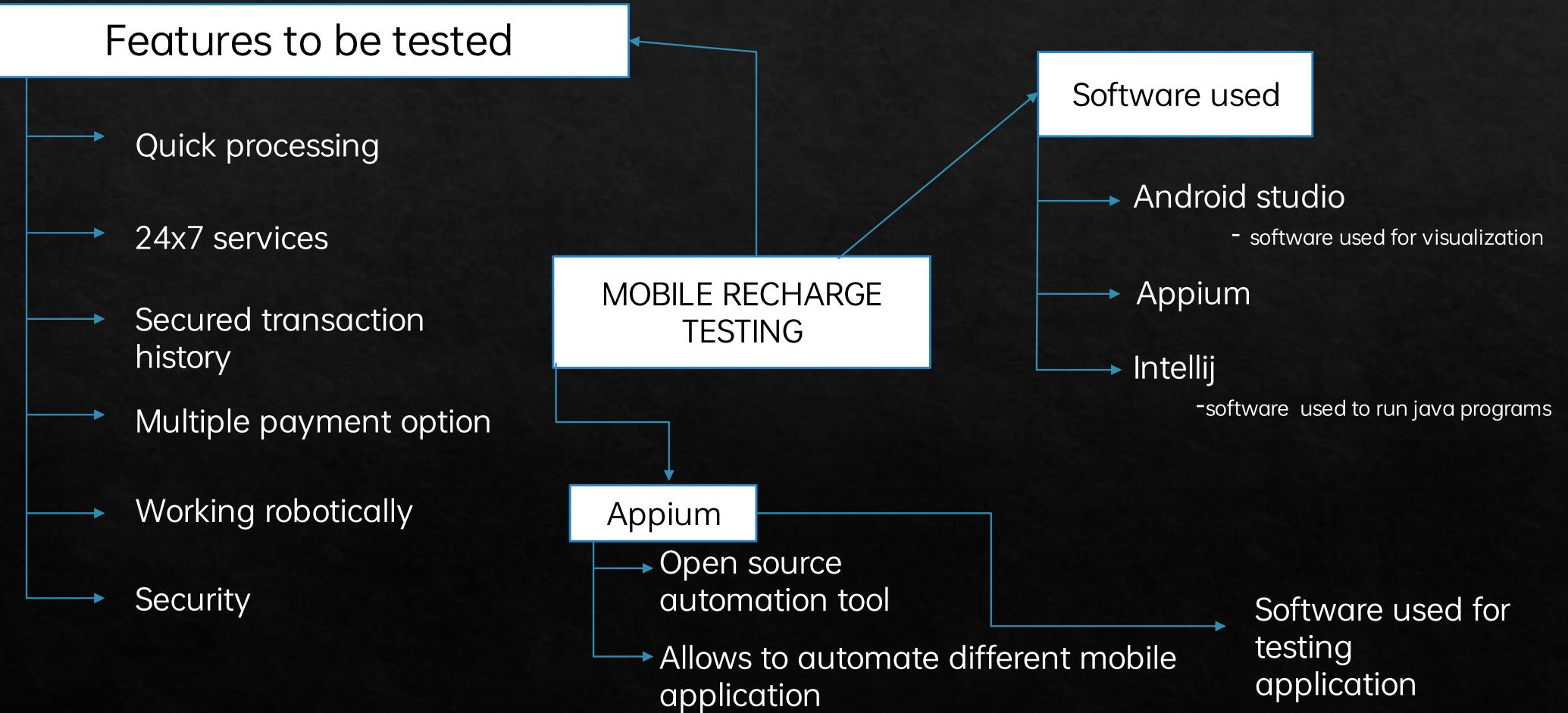
- ☒ To develop the testing environment, Appium software is installed in the windows and IntelliJ is another software, a JDK , which provides us environment to run the java programs.
- ☒ For visualization, android studio is installed.
- ☒ To determine whether the application performs as per the requirement under different load conditions.
- ☒ To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
- ☒ To determine whether the existing client-server configuration setup provides the required optimum performance level.
- ☒ The functionalities that are involved in the online recharge system involves below :

- Login to application with valid username and password.
- enter mobile no.
- check weather it is valid or not.
- enter recharge amount
- for that check recharge plan.
- check do you have required balance in your account or not.
- generate OTP.
- proceed to payment option.
- send amount deduction message.
- send recharge successful message.

# FLOW CHART



# CONCEPT MAP



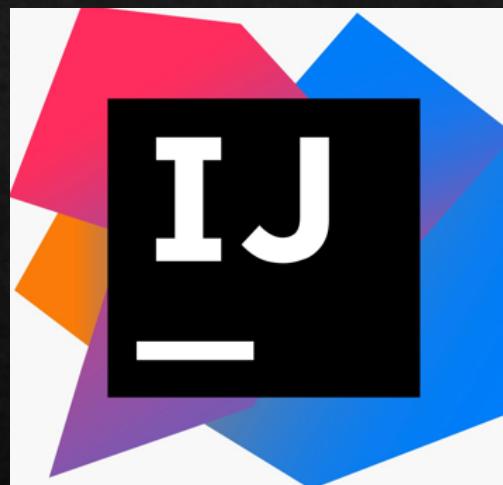
# TEST CASES

- Login to application with valid username and password.
- enter mobile no.
- check weather it is valid or not.
- enter recharge amount
- for that check recharge plan.
- check do you have required balance in your account or not.
- generate OTP.
- proceed to payment option.
- send amount deduction message.
- send recharge successful message.

# APP TOOLS



ANDROID  
STUDIO

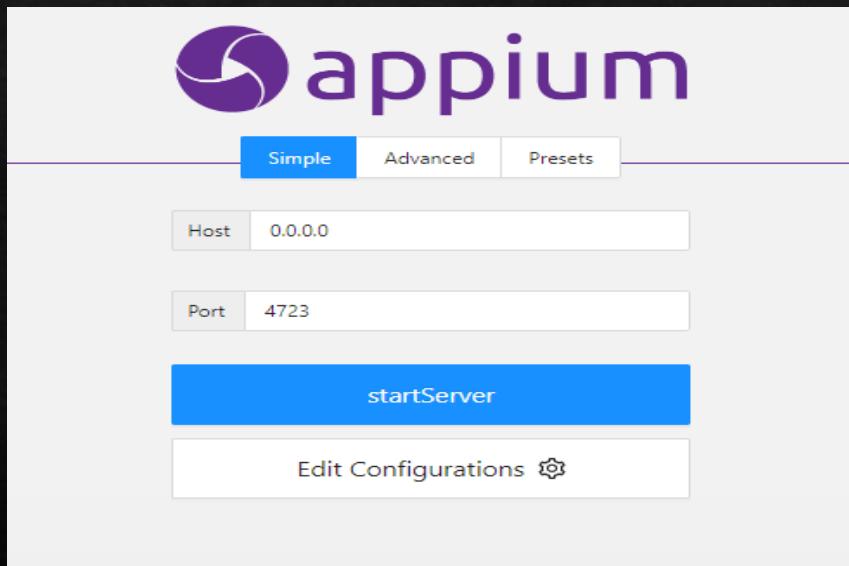


IntelliJ



APPIUM  
STUDIO

# APPIUM SERVER



BEFORE STARTING SERVER

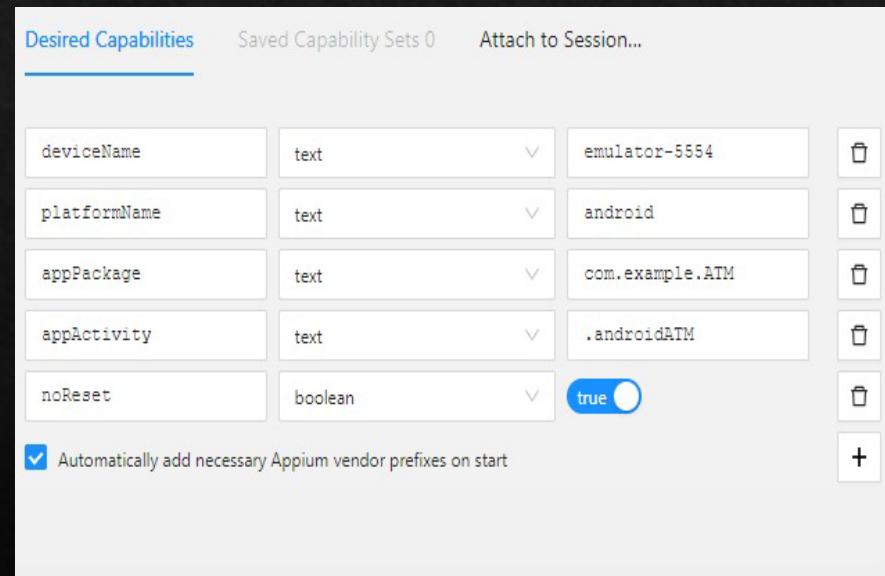
The screenshot shows a terminal window with the Appium logo icon. The window title is 'File View Help'. The main area displays the following log output:

```
[Appium] Welcome to Appium v1.22.3
[Appium] Non-default server args:
[Appium]   relaxedSecurityEnabled: true
[Appium]   allowInsecure: {
[Appium]     }
[Appium]   denyInsecure: {
[Appium]     }
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

AFTER STARTING SERVER

# IMPLEMENTATION & TESTING SCREENSHOOTS

- Give the Desired Capabilities in Appium Inspector.
- It will start the session and gives you java code to start testing.
- Paste the code in IntelliJ IDE and Start the Testing.



# IMPLEMENTATION OF TESTING

Appium Server    Select Cloud Providers

Remote Host: 127.0.0.1    Remote Port: 4723

Remote Path: /    SSL:

> Advanced Settings

Desired Capabilities    Saved Capability Sets 0    Attach to Session...

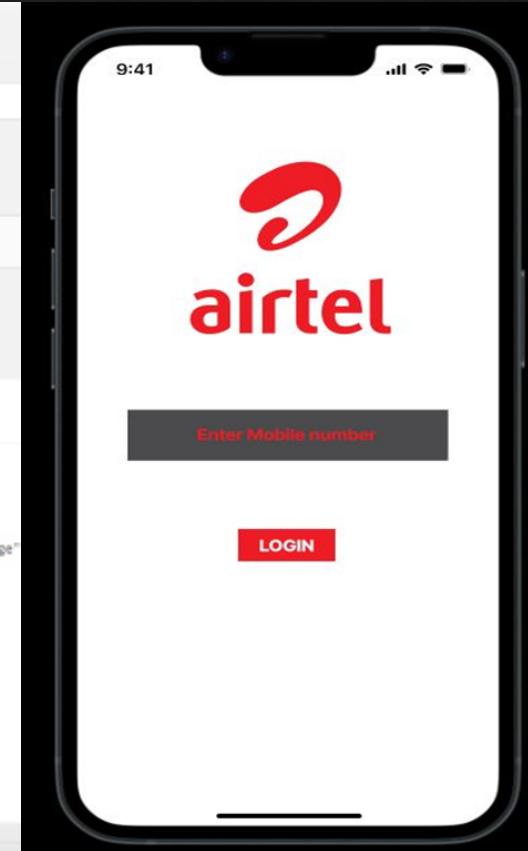
IMPLEMENTATION OF TESTING

deviceName	text	emulator-5554	<input type="button" value=""/>
platformName	text	android	<input type="button" value=""/>
appPackage	text	com.example.NoboileREcharge	<input type="button" value=""/>
appActivity	text	.SplashScreen	<input type="button" value=""/>

Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
[{"deviceName": "emulator-5554", "platformName": "android", "appPackage": "com.example.NoboileREcharge", "appActivity": ".SplashScreen"}]
```



# IMPLEMENTATION OF TESTING

Appium Server Select Cloud Providers

Remote Host: 127.0.0.1 Remote Port: 4723

Remote Path: / SSL

> Advanced Settings

Desired Capabilities Saved Capability Sets 0 Attach to Session...

deviceName	text	emulator-5554
platformName	text	android
appPackage	text	com.example.MobileRecharge
appActivity	text	.SplashScreen

Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{  
  "deviceName": "emulator-5554",  
  "platformName": "android",  
  "appPackage": "com.example.MobileRecharge",  
  "appActivity": ".SplashScreen"  
}
```

The screenshot shows the 'recharge' app interface on an iPhone 14. The top status bar indicates the time is 9:41 and the battery level is at 100%. The main screen displays a list of recommended mobile plans:

- ₹239** Unlimited calls, 1.5 GB per day, 28 days validity. Plan details: **Free Hellotunes**, view details.
- ₹265** Unlimited calls, 1.5 GB per day, 30 days validity. Plan details: **Free Hellotunes**, view details.
- ₹666** Unlimited calls, 1.5 GB per day, 84 days validity. Plan details: **Apollo 24|7 Circle**, view details.
- ₹148** 15 GB data, existing validity. Plan details: **Xstream App (Claim Hoichoi and Win)**, view details.

# IMPLEMENTATION OF TESTING

Appium Server Select Cloud Providers

Remote Host: 127.0.0.1 Remote Port: 4723

Remote Path: / SSL

Advanced Settings

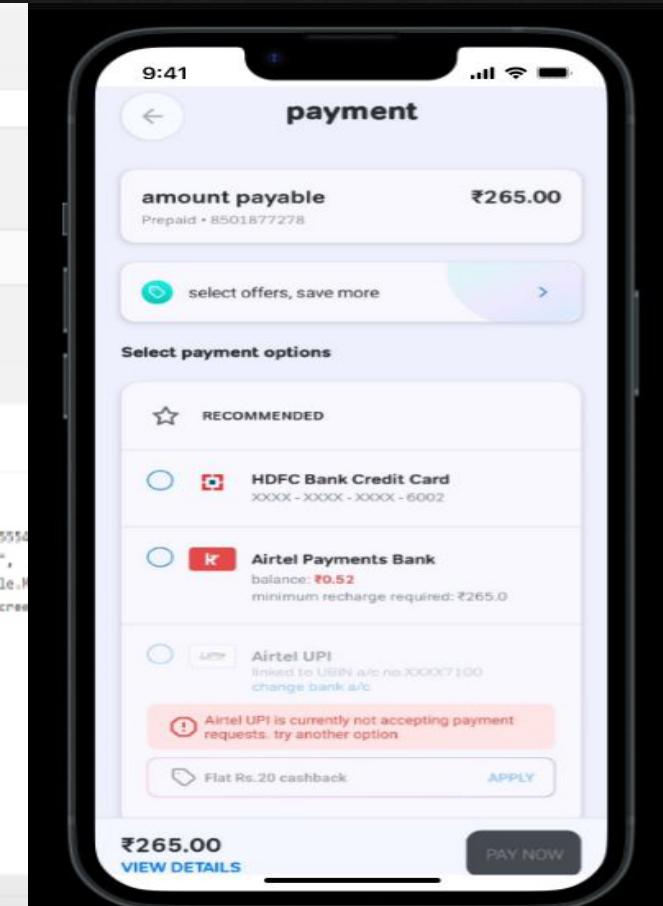
Desired Capabilities Saved Capability Sets 0 Attach to Session...

deviceName	text	emulator-5554
platformName	text	android
appPackage	text	com.example.MobileRecharge
appActivity	text	.SplashScreen

Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{  
  "deviceName": "emulator-5554",  
  "platformName": "android",  
  "appPackage": "com.example.MobileRecharge",  
  "appActivity": ".SplashScreen"  
}
```



# IMPLEMENTATION OF TESTING

Appium Server    Select Cloud Providers

Remote Host: 127.0.0.1    Remote Port: 4723

Remote Path: /    SSL:

> Advanced Settings

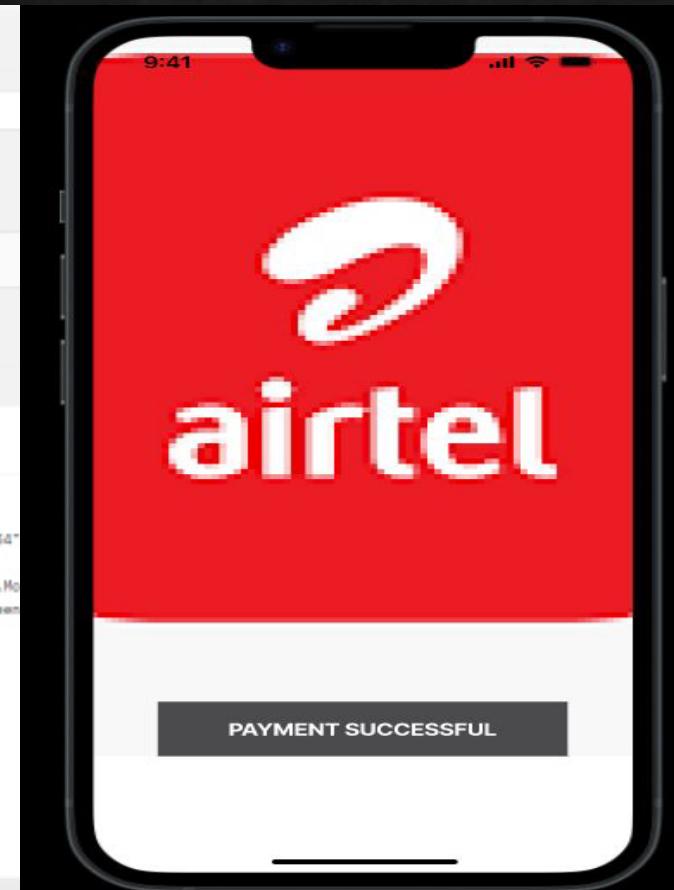
Desired Capabilities    Saved Capability Sets 0    Attach to Session...

deviceName	text	emulator-5554
platformName	text	android
appPackage	text	com.example.MoboilesZohar
appActivity	text	.splashScreen

Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{  
  "deviceName": "emulator-5554",  
  "platformName": "android",  
  "appPackage": "com.example.MoboilesZohar",  
  "appActivity": ".SplashScreen"  
}
```



# IMPLEMENTATION OF TESTING

Appium Server    Select Cloud Providers

Remote Host: 127.0.0.1    Remote Port: 4723

Remote Path: /    SSL:

> Advanced Settings

Desired Capabilities    Saved Capability Sets 0    Attach to Session...

deviceName	text	emulator-5554
platformName	text	android
appPackage	text	com.example.mobilesexchange
appActivity	text	.splashscreen

Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{  
  "deviceName": "emulator-5554",  
  "platformName": "android",  
  "appPackage": "com.example.mobilesexchange",  
  "appActivity": ".splashscreen"  
}
```



# TESTCASE OUTCOMES

TEST CASES	POSITIVE/NEGATIVE
Enter valid email or mobile no.	positive
Enter invalid mobile no.	negative
Click on submit button	positive
Select the valid recharge plan	positive
Enter recharge amount	positive
Check required balance in account	positive
Generate OTP	positive
Enter invalid OTP	negative
Proceed to payment option	positive
Enter invalid payment amount	negative
Payment deduction message	positive
Payment successful message	positive

# CONCLUSION

This project has been developed successfully and the performance of the system has been found satisfactory. Use of this interface helps customer in having immediate information about available recharge tariff and do recharge without wasting their precious time. User friendly Interface also for the admin to add and delete tariff information.

THANK YOU