# Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testing Iteration:** 2 | | **Date:** 13/05/2017 | **Owner:** Kent Huang, Daniel Park , Terence Leung, Bill Xue | | |
| **Test / Scenario** | **Input** | **Expected Result** | **Actual Result** | **Passed** | **Date fixed** |
| Test that all activities load | Activity selected | All activities load without fail | All activities load | Y | N/A |
| Customer app: test page back-redirection when user is logged into the home page | press the back button on home page | exit the app cause the base page should be the homepage when user is logged in | exit the app, base page become the homepage when user is logged in | Y | 03/May |
| Customer app: test page back-redirection from login page to the welcome page | press back button on the login page | redirect back to welcome page | redirect back to welcome page | Y | N/A |
| Customer app: test page back-redirection from order page to home page | press back button on the order page | redirect back to home page | redirect back to home page | Y | N/A |
| Customer app: test page back-redirection from order confirmed page to home page | press back button on order confirmed page | redirect back to home page | redirect back to home page | Y | 03/May |
| Customer app: app should be locked in portrait mode | viewing any activity | activity locked in portrait mode | activity locked in portrait mode | Y | 07/May |
| Customer app: password field in login and register page be masked | type in characters in password field | masked password shown up | masked password shown up | Y | 03/May |
| Customer app home activity: scrollview | viewing customer app home activity | header and footer should be lock in relative position, while the middle part (company information) should be scrollable | header and footer (logo and contact info) are locked while the middle part being scrollable | Y | N/A |
| Company app: test page back-redirection when user is logged into the home page | press the back button on home page | exit the app cause the base page should be the homepage when user is logged in | startup page become homepage when user is logged in | Y | 04/May |
| Company app: test page back-redirection after user presses logout | press back button on login page right after the user logged out | exit the app | exit the app | Y | 04/May |
| Company app: test tab(New order, Order in progress, Driver and Profile) redirections on dispatch page | press another tab | switch to a new view for specified tab | switched to a new view for specified tab | Y | 04/May |
| Company app: password field in login and register page be masked | type in characters in password field | masked password shown up | masked password shown up | Y | 04/May |
| Customer app: input validation: login | email field left empty | unable to login | toast: “invalid email or password” | Y |  |
| Customer app: input validation: login | email not in the database:  hello@gmail.com | unable to login | toast “invalid email” | Y |  |
| Customer app: input validation: login | incorrect email:  adfsd | toast displaying incorrect email | toast: “invalid email or password” | Y |  |
| Customer app: input validation: login | correct email, incorrect password.  email: [testcust@gmail.com](mailto:testcust@gmail.com)  password: (empty) | unable to login | toast: “invalid email or password” | Y |  |
| Customer app: server validation: login | correct email, incorrect password.  email: [testcust@gmail.com](mailto:testcust@gmail.com)  password: 123456 | unable to login | toast: “invalid password” | Y |  |
| Customer app: server validation: login | correct **customer** email, correct password.  email: [testcust@gmail.com](mailto:testcust@gmail.com)  password: 123asd | logged into home page;  customer view: there should be an “order now button” | logged into home page;  there’s an “order now” button | Y |  |
| Customer app: server validation: login | correct **restaurant** email, correct password.  email: [testrest@gmail.com](mailto:testrest@gmail.com)  password: 123asd | logged into home page;  restaurant view: there should be a “request driver” button | logged into home page;  there’s a “request driver” button | Y |  |
| Customer app: logout function | logout from profile page | log out to welcome page, “back” button should **not** take user back to the home page | logged out to welcome page, “back” button exits the app | Y |  |
| Customer app: activities based on user type : **guest** | login as guest | profile icon is a guest icon;  order page is not prefilled;  has “order now” button on home page;  profile page showing guest info;  profile page has edit profile and change password buttons disabled; | profile icon is a guest icon;  order page is not prefilled;  has “order now” button on home page;  profile page shows “guest”  profile page has edit profile and change password buttons disabled; | Y |  |
| Customer app: activities based on user type : **customer** | login with a customer account | profile icon is a customer icon;  has “order now” button on home page;  order page is prefilled with customer’s info;  profile page showing customer info;  profile page has all buttons enabled; | profile icon is a customer icon;  has “order now” button on home page;  order page is prefilled with customer’s info;  profile page showing customer info;  profile page has all buttons enabled; | Y |  |
| Customer app: activities based on user type : **restaurant** | login with a restaurant account | profile icon is a restaurant (chef) icon;  has “request driver” button on home page;  “request driver” button takes user to the request\_driver\_activity;  order page is prefilled with restaurant info;  profile page showing restaurant info;  profile page has all buttons enabled; | profile icon is a restaurant icon;  has “request driver” button on home page;  “request driver” button takes user to the request\_driver\_activity;  order page is prefilled with restaurant info;  profile page showing restaurant info;  profile page has all buttons enabled; | Y |  |
| Customer app: sign up (Register) activity | sign up without filling anything in | unable to sign up, a toast giving warnings and stay on the same activity; | toast “failed to sign up”;  email edittext has an error sign;  stayed on the same activity; | Y |  |
| Customer app: sign up (Register) activity | incorrect email address, everything else not filled  **email used: aaaaa** | unable to register;  also highlight on the email edittext showing it’s not a correct email; | unable to register,  error message on email edittext: “invalid email address” | Y |  |
| Customer app: sign up (Register) activity | weak password:  user enters a password that’s less than 6 characters long.  **password used:**  **123** | highlight password edittext field showing it’s a weak password | error message on first password edittext: “password must contain at least 6 characters” | Y |  |
| Customer app: sign up (Register) activity | second password is different than the first one (both valid passwords).  **password 1: 123456**  **password 2: 123123** | advise user that the two passwords entered are not the same | error message on second password edittext: “password does not match” | Y |  |
| Customer app: sign up (Register) activity | name edittext left blank | highlight name edittext for user to enter a name | error message on name edittext: “Enter a name” | Y |  |
| Customer app: sign up (Register) activity | **name field input: a** | name should pass since anything that’s not an empty string is allowed | passed | Y |  |
| Customer app: sign up (Register) activity | **name field input: !** | name should pass since anything that’s not an empty string is allowed | passed | Y |  |
| Customer app: sign up (Register) activity | phone number field left empty | highlight phone number edittext for user to enter a phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 1** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 123** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 123--** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 123-1** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 12345678901234567890** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | **phone number field: -** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: .** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: (778)1** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | **phone number field: (778)-1-1** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | **phone number field: ()-1-1** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: 5-(1)** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: (123)-2.** | invalid phone number | error message on phone edittext: “Invalid phone number” | Y |  |
| Customer app: sign up (Register) activity | **phone number field: (123)2-2** | pass | passed | Y |  |
| Customer app: sign up (Register) activity | address field and address detail left empty | pass | passed | Y |  |
| Customer app: edit profile activity | edit profile informations: name, phone, address, address detail | new profile info is updated to the database and reflected back to the profile page;  same validation rules used on registration page also apply here | passed.  able to edit profile and profile info is updated to db and reflected back to the profile page. | Y |  |
| Customer app: profile - change password activity | wrong current password, correct new passwords.  **input used here:**  **pw:abcdef**  **new pw: 111111**  **new pw:111111** | unable to change password | toast: “Error Invalid Credential” | Y |  |
| Customer app: profile - change password activity | correct current password, new passwords don’t match.  **input used here:**  **pw:123456**  **new pw: 111111**  **new pw:111112** | unable to change password | toast: “Please enter identical passwords” | Y |  |
| Customer app: profile - change password activity | correct current password, new passwords don’t match.  **input used here:**  **pw:123456**  **new pw: 111111**  **new pw:111111** | password updated | password updated.  toast: “password updated” | Y |  |
| Customer app: customer: place new order | same validation rules. Name, phone, address and order detail have to be filled, email and address detail can be left empty | same validation rules. Name, phone, address and order detail have to be filled, email and address detail can be left empty.  Order should be send out when the required fields are filled, user will go to the “current orders” activity | passed | Y |  |
| Customer app: restaurant: request driver | same validation rules. Restaurant name, phone, address; ready minutes, total amount; customer name, phone, address and order detail all have to be filled, other fields can be empty | same validation rules. Restaurant name, phone, address; ready minutes, total amount; customer name, phone, address and order detail all have to be filled, other fields can be empty.  Driver request should be send out when the required fields are filled, user will go to the “current orders” activity | passed | Y |  |
| Customer app: new order & request driver | correct input fields, user press “submit” button | dispatcher receive notification of “new order” | dispatcher receive notification of “new order” | Y |  |
| Customer app: new order & request driver | correct input fields, user press “submit” button | order sent to Firebase, dispatcher should be able to see the order |  |  |  |