test_01: Passenger sign up

Purpose: To test whether a passenger can register with his/her email.
Prereq: No.
Test Data:
email_1 = bclinton@whitehouse.gov.us
password_1 = president
name_1 = Bill Clinton
phone_1 = 202-456-1111
address_1 = The White House 1600 Pennsylvania Avenue NW
Washington, DC 20500
credit card = 938589938

email_2 = jbonde@sp.gov.uk
password_2 = 007
name_2 = James Bonde
phone_2 = 760-753-7477
address_2 = 1620 Haydn Dr Cardiff By The Sea, CA 92007-2304
credit card = 956789938

Steps: 1. Passenger tap "Register" button to enter register screen.
2. Enter email, password, name, phone, address, credit card.
3. Tap "Picture" button to select a picture of the passenger from
the phone album, or take a picture right now.
4. Tap "Sign Up" button.
5. See whether the passenger could successfully sign up.

Notes and Questions:
1. The App should verify whether the email address the passenger
entered is a valid email address automatically.
2. The password should be confirmed -- entered twice.

test_02: Passenger log in

Purpose: To test whether the passenger can log in properly.
Prereq: Passenger already registered.
Test Data:
email_1 = bclinton@whitehouse.gov.us
password_1 = president

email_2 = jbonde@sp.gov.uk
password_2 = 007

email_3 = bill gates //invalid address
email_4 = george@bjtu //invalid address

Steps: 1. Launch the app RideWithMe in Android phone.
2. Input test email, and corresponding password.
3. Tap LogIn.
4. See whether the passenger could log in.

Notes and Questions:
1. The app should verify whether the email address the passenger
entered is a valid email address automatically.
2. The password should not be displayed -- display dots instead.
3. After the passenger logged in, the app should lead the
passenger to the "Pick Me Up" screen.

test_03: Pick the passenger (pick me up)

Purpose:	To test whether the app could display the passenger's location on the map; and whether the user could start his/her booking process.

Prereq:	Passenger already logged in.
Test Data:	The test is based on the user's GPS location, no test data needed.

Steps:	1. Passenger log in.
2. See whether the map displayed the passenger's location correctly.
3. Tap "Pick Me Up".
4. See whether the app could lead the user to the destination selection screen.

Notes and Questions:
1. The app should display the passenger's location based on GPS.

test_04: Book a ride

Purpose:	To test whether the passenger could select his destination by Google Maps search.
Prereq:	Passenger already logged in.

Test Data:	destination_1 = sanlitun
destination_2 = wudaokou
destination_3 = the summer palace
destination_4 = No Such a place in Beijing	//place that doesn't exist.

Steps:	1. Passenger type the destination into the Destination search box, and tap "Search".
2. See whether the map displayed the passenger's destination correctly.
3. Tap "Book a ride".
4. See whether the app could display a list of drivers for selection.

Notes and Questions:
1. The app should display the destination on map.
2. If the destination the passenger input a invalid address, the app should provide a notification to the passenger, and a way to correct his/her input.

test_05: Choose Car

Purpose:	To test whether the passenger could check the details of a driver, and choose a driver.
Prereq:	Passenger already logged in, and his/her destination is valid.

Test Data:	driver_1 = Barack Obama
driver_2 = George W. Bush
driver_3 = Bill Clinton
driver_4 = Ronald Reagon
driver_5 = Jimmy Carter
driver_6 = Gerald Ford

Steps:	1. Passenger select a driver by tap "Deatils" and check the driver's information.
2. Passenger tap the back key to check for another driver's details.
3. Repeat steps 1 and 2, until he find the driver best satisfied him/her.

4. The passenger tap "Choose" to choose this driver.

Notes and Questions:
1. The passenger should be able to view several nearby driver's details.

test_06: Alert the driver to arrive faster (Hurry up)

Purpose: To test whether the passenger could alert the driver to come to pick the passenger faster.

Prereq: Passenger has already chosen a driver, and the driver accepted.

Test Data: No need.

Steps: 1. Passenger tap "Hurry up" button to alert the driver.
2. Repeat step 1 several times.

Notes and Questions:
1. When the passenger tap the "hurry up" button, the driver should receive a "hurry up" message.

test_07: View Payment Notification and Rate The Driver

Purpose: To test whether the passenger could view the payment notification
Prereq: Destination arrived.

Test Data: No need.

Steps: 1. Passenger tap "I've Arrived" button. The app should display the payment information, and invite the passenger to rate the driver.
2. The passenger give the driver 4-star rating, and tap the "submit" button.
3. Start a new test process, give the driver 1, 2, 3, 4, 5 stars rating each time by repeat step 1 and 2.

Notes and Questions:
1. During the process the passenger gives the driver a rating, he should be able to
change his mind by edit the rating before he tap the "Submit" button.

==========
test_08: Driver login

Purpose: To test whether the driver could login.
Prereq: Driver has already registered physically in the office.

Test Data:

d_email_1 = rms@gnu.org
d_password_1 = rstallman

d_email_2 = torvalds@osdl.org
d_password_2 = ilovelinux

d_email_3 = esr@thyrsus.com
d_password_3 = eraymond

```
                            d_email_4    = guido@python.org
                            d_password_4 = gossum


Steps:
                            1. Driver launch the app, input his email address,
and password, tap "Sign In" button.

Notes and Questions:
              1. The driver should be able to login, and the app should led the
driver to the next screen.



test_09: Search for Passenger
-----------------------------
Purpose:              To test whether the driver could search for a p passenger.

Prereq:               Driver has already logged in.

Test Data:            No need.

Steps:
                            1. After driver logged in, he tap the "Look for
passenger" button.

Notes and Questions:
                      1. There should be a several passengers displayed on the
map, if they are available.


test_10: Bid for Book
---------------------
Purpose:              To test whether the driver could check and bid with the
passenger.

Prereq:               Driver has already logged in, and selected a passenger.

Test Data:
                            bid_amoutn_1 = 30 RMB
                            bid_time_1 = 10 Mins

                            bid_amoutn_2 = 50 RMB
                            bid_time_2 = 20 Mins

                            bid_amoutn_3 = 30 RMB
                            bid_time_3 = 15 Mins

                            bid_amoutn_4 = 30 RMB
                            bid_time_4 = 30 Mins

                            bid_amoutn_4 = a30 RMB             // invalid
                            bid_time_4 = 3o Mins        // invalid: letter 'o',
not number 'zero' by mistyping.

Steps:
                            1. The driver input bid amount and time he would
like to bid, tap the "Submit" button.

Notes and Questions:
```

1. The App should verify whether the driver's input is valid.

test_11: Pick Up Passenger

Purpose: To test whether the driver could notify the system that the passenger has been successfully picked up.

Prereq: Driver and the passenger have already reach aggreement on the bid.

Test Data: No need.

Steps:
1. After they reach the agreement, the driver goes to pick the passenger. When the passenger is picked, he tap the "Picked" button.

Notes and Questions:
1. A message should sent to the server when the driver tapped "Picked" button.

test_12: Drop Passenger (Successfully arrived)

Purpose: To test whether the driver could notify the system that the passenger has arrived his destination.

Prereq: The driver has successfully take the passenger to his destination.

Test Data: No need.

Steps:
1. After they reach the destination, the driver taps the "Destination Reached" button.

Notes and Questions:
1. A message should sent to the server when the driver tapped "Destination Reached" button.

test_13: View Payment Notification

Purpose: To test whether the driver could view the payment notification.

Prereq: The passenger has reached his destination. driver has already tap the "Destination Reached" button to notify the server.

Test Data: No need.

Steps:
1. After the driver tapped the "Destination Reached" button, (as in test_12), a payment notification should be displayed on the screen.

Notes and Questions:
1. The content of the payment notification should include the amount of money the driver earned in this trip.

test_14: Rate the Passenger

Purpose: To test whether the driver could rate the passenger.

Prereq: The passenger has reached his destination. driver has already tap the "Destination Reached" button to notify the server.

Test Data:
rate_1 = 4 stars
rate_2 = 3 stars
rate_3 = 4.5 stars

Steps:
1. After the driver tapped the "Destination Reached" button, (as in test_12), a rating-bar should appear on the screen for the driver to give his rating.
2. The driver taps the rating-star to give his rating.
3. Tap the "Submit" button.

Notes and Questions:
1. The rating should be transferred to the server.

==========
test_15: Login (Administrator)

Purpose: To test whether the administrator could login.

Prereq: No.

Test Data:
admin_1 = lombe
passwd_1 = king4all

admin_2 = sebastian
passwd_2 = hola88

admin_3 = yufeng
passwd_3 = caesar3

---
intruder_1 = bill // intruder, should not login
passwd_1 = anything

intruder_2 = larry // intruder, should not login
passwd_2 = easypass

Steps:
1. Each administrator enter his user name and password, and click "login", to see whether he could successfully log in.
2. Each intruder enter his user name and password, and click "login", to see whether he could successfully log in --- obviously he should NOT.

Notes and Questions:
1. In order to stop the robot intrusion using brute force, we may add a random validation code.
2. We'd better stop an administrator to log in if he

entered the wrong password for continuous 5 times.

test_16: Add driver

Purpose: To test whether the administrator could add a driver to the database.

Prereq: The administrator has already logged in.

Test Data:

```
driver_id_1       = tblair@bigben.gov.uk
driver_passwd_1   = thesunneverfall
driver_address_1  = London W27JU UK
billing_address_1 = P.O. Box 60519
car_no._1         = uk8f37
bank_account_1    = 3894793884

driver_id_2       = dcameron@bigben.gov.uk
driver_passwd_2   = staycalm
driver_address_2  = London W73JU UK
billing_address_2 = P.O. Box 60659
car_no._2         = uurf37
bank_account_2    = 3894793688

driver_id_3       = gbrown@bigben.gov.uk
driver_passwd_3   = carryon
driver_address_3  = London W7k9U UK
billing_address_3 = P.O. Box 60788
car_no._3         = ek67k3
bank_account_3    = 3894793234
```

Steps:

1. The administrator click "Add Driver" (or "Register the driver"), there should be a web page to enter the driver information.
2. Enter the driver information and click "Submit".

Notes and Questions:

1. Each field of the driver's information should be unique. The system should have a way to check if there's any duplications. (But it is reasonable for the husband and his wife to have the same address and bank account number.)

test_17: Update the driver information

Purpose: To test whether the administrator could update the driver's information in the database.

Prereq: The administrator has already logged in, and some drivers have been added.

Test Data:

```
admin_1  = lombe
passwd_1 = king4all

driver_id_1          = tblair@bigben.gov.uk
new_driver_address_1 = London W49JU UK          //
```
the driver changed his address

```
                            driver_id_2          = dcameron@bigben.gov.uk
                            new_car_no._2        = uurf74
                            new_bank_account_2   = 38947933455        // the
driver changed his car and his bank account number.

                            driver_id_3          = gbrown@bigben.gov.uk
                            driver_passwd_3      = ilovedriving
// the driver reseted his password.

Steps:
                            1. The administrator select a driver and click
"Edit".
                            2. Edit the driver's information and click "Submit".

Notes and Questions:
                      1. The new information should be updated smoothly, and sync
back on the driver's phone.


test_18: Remove a driver
------------------------
Purpose:              To test whether the administrator could remove a driver in
the database.

Prereq:               The administrator has already logged in, and some drivers
have been added.

Test Data:
                            admin_1  = lombe
                            passwd_1 = king4all

                            driver_id_1  = tblair@bigben.gov.uk
                            driver_id_2  = dcameron@bigben.gov.uk


Steps:
                            1. The administrator select a driver and click
"Remove".
                            2. The administrator confirm his operation by
clicking "Yes".

Notes and Questions:
                      1. There should be a way to prevent the administrator to
remove drivers by accident. Maybe the system should require him to input his
password again.


test_19: View a driver
------------------------
Purpose:              To test whether the administrator could view a driver in
the database.

Prereq:               The administrator has already logged in, and some drivers
have been added.

Test Data:
                            admin_1  = lombe
                            passwd_1 = king4all

Steps:
                            1. The administrator select a driver and click
```

"view".

Notes and Questions:
1. The administrator should also be able to sort the driver by rating, address, car number, bank account. (we may consider it as an additional feature.)

test_20: Update a passenger

Purpose: To test whether the administrator could update a passenger in the database.

Prereq: The administrator has already logged in, and there are some passengers registered.

Test Data:
admin_1 = lombe
passwd_1 = king4all

passenger_1 = bclinton@whitehouse.gov.us
password_1 = president
password_new = oldstory //change the password for the passenger.
name_1 = Bill Clinton
phone_1 = 202-456-1111
address_1 = The White House 1600 Pennsylvania Avenue NW Washington, DC 20500
credit card = 938589938

passenger_2 = jbonde@sp.gov.uk
password_2 = 007
name_2 = James Bonde
phone_2 = 760-753-7477
phone_new = 760-753-4545 //change the phone number for the passenger.
address_2 = 1620 Haydn Dr Cardiff By The Sea, CA 92007-2304
credit card = 956789938

Steps:
1. The administrator select a passenger and click "edit".
2. The administrator update the passenger's information and click "Submit".
3. Confirm his operation by clicking "Yes".

Notes and Questions:
1. Usually the passenger's information is managed by the passenger using the Android application. The administrator have the power to change information for administration purpose.

test_21: Remove a passenger

Purpose: To test whether the administrator could remove a passenger in the database.

Prereq: The administrator has already logged in, and there are some passengers registered.

Test Data:
admin_1 = lombe

passwd_1 = king4all

passenger_1 = bclinton@whitehouse.gov.us
passenger_2 = jbonde@sp.gov.uk

Steps:
1. The administrator select a passenger and click "remove".
2. Confirm his operation by clicking "Yes".

Notes and Questions:
1. The system should have a way to prevent the administrator removing a passenger by accident. Maybe the administrator should input his password again.

test_22: View a passenger

Purpose: To test whether the administrator could view a passenger in the database.

Prereq: The administrator has already logged in, and some passengers have already registered.

Test Data:
admin_1 = lombe
passwd_1 = king4all

Steps:
1. The administrator select a passenger and click "view".

Notes and Questions:
1. The administrator should also be able to sort the passenger by rating, address, bank account. (we may consider it as an additional feature.)