

System Admin Manual

<SNAPVALET>

<Team No- 03>

Name	Role
Brian Vanover	Project Manager, Developer
Brian Bousman	Operational Concept Engineer, Developer
Ditong Ding	System Architect, Developer.
Molly Karcher	IIV & V, Quality Focal Point, Developer.
Patrick Horng	Feasibility Analyst, Developer
Ridhima Manjrekar	Requirements Engineer, Life Cycle Planner, Developer.

<03/29/15>

Version History

Date	Author	Version	Changes made	Rationale
04/08/15	RM	2.0	Added expected information in each section	To comply with Instructional ICM-Sw standard

Table of Contents

VERSION HISTORY	III
TABLE OF CONTENTS	IV
TABLE OF TABLES.....	V
TABLE OF FIGURES	VI
1. Introduction	1
1.1 System Overview	1
1.2 System Requirements	1
2. Deployment Instructions.....	Error! Bookmark not defined.
2.1 Get Started on Local Machine	Error! Bookmark not defined.
2.2 Cloud Deployment.....	Error! Bookmark not defined.
3. Database Schema	Error! Bookmark not defined.
4. Transaction Summary Instructions	5

Table of Tables

No table of figures entries found.

Table of Figures

No table of figures entries found.

1. Introduction

1.1 System Overview

<< *State the brief purpose of the system to which this manual applies.*

Please note that if your project does not need a user manual, but a system administrator's manual, you can tailor this template to fit your project. On the other hand, if you need both End user's manual and System administrator's manual, please separate into two documents. >>

1.2 System Requirements

<< *Describe the minimum hardware and software (Operating System, etc.) requirements for the system. >>*

1.2.1 Hardware Requirements

Hardware requirements are: Android Smart phone (Customer and Valet), PC/ Laptop (Valet Company)

1.2.2 Software Requirements

Android Operating System , Modern browser with JavaScript enabled

1.2.3 Other Requirements

Credit Card

2. Deployment Instructions

2.1 Get Started on Local Machine

1. Install Play Framework
2. Install Eclipse Luna
3. Install Scala IDE Plugin
4. Install eGIT plugin
5. Clone GIT repo
6. Activator Eclipse
7. Install MySQLServer
8. Install SQL Workbench
9. Update application config file for connection and username/password
10. Run the design db folder SQL Scripts
11. Activator Run

2.2 Cloud Deployment

1. Sign up and login to your account on Clever Cloud.
2. On the left hand console click on 'add an application.'
3. Select "No, the source code of your application is hosted on Github."
 - You can also select the github option, but problems may occur, and therefore this option is not recommended.
4. Select 'Play 2 and Java.' In the instance page, click next.
5. Name you app and select Montreal, Canada.
6. Select MySQL database add-on.
7. Choose what option (max connection limit, max database size) for your DB.
8. Name your MySQL add-on, and change your zone to North America.
9. Click next on the environment variables page.
10. Copy the two lines of git code in the deployment page.
11. Before you deploy your code, go back to the source for the Play Framework project.
12. In the conf folder, select the app.conf file and change the DB configuration settings.
 - On the left console on clever cloud, select your DB add-on.
 - Click on Configuration.
 - Copy all of the credentials: Host, User, Password and replace them in the app.conf.
 - Also copy the Database name for later use in configuring the database.
13. Run the two copied git commands from clever cloud to deploy the application.
14. Now to configure the database use the mysql command line under "You can use this to connect to your database" in Configuration.
15. Find the create.sql script in the source code under design->database folder.
16. Remove the first two lines in the script and add: "use (Database name you copied previously);".
17. Push this script to the cloud DB by running:
mysql command code < database1.sql
18. You're DONE, check if the application is running smoothly by clicking on the domain name of your app.

3. Database Schema

```
create database snapvalet;
use snapvalet;
```

```
create table valet_company
(
    valet_company_ID int(32) primary key auto_increment,
    user_name varchar(255) not null unique,
    user_password varchar(255) not null,
    valet_company_name varchar(255) not null unique
    /* ... */
);
```

```
create table valet
(
    valet_ID int(32) primary key auto_increment,
    user_name varchar(255) unique,
    user_password varchar(255),
    valet_name varchar(255),
    valet_state tinyint(8) not null,
    /* ... */
    valet_company_ID int(32) not null,
    foreign key (valet_company_ID) references valet_company(valet_company_ID)
);
alter table valet auto_increment=10000001; /* make auto increment start from 10,000,001 */
```

```
create table customer
(
    customer_ID int(32) primary key auto_increment,
    user_name varchar(255) not null unique,
    user_password varchar(255) not null,
    customer_name varchar(255) not null
    /* Other information needed for payment*/
    /* ... */
);
```

```
/* add a column current_valet_fee */
```

```
create table location
(
    location_ID varchar(32) primary key,
    location_name varchar(255) not null,
    location_latitude double not null,
    location_longitude double not null,
    location_address varchar(255) not null,
    /* Other information needed for check position; Check google location API and use their fields*/
    /* ... */
    valet_company_ID int(32), /* null means it belongs to no valet company */
    current_valet_ID int(32),
    current_valet_fee float,
    foreign key (valet_company_ID) references valet_company(valet_company_ID),
    foreign key (current_valet_ID) references valet(valet_ID)
);
```

```
create table valet_request
```

```

(
    valet_request_ID int(32) primary key auto_increment,
    valet_request_status tinyint(8) not null,
    ticket_number varchar(255) not null,
    timestamp datetime not null,
    payment_method tinyint(8) not null, /*Card/Cash*/
    tips float not null,
    valet_fee float not null,
    /* CreditCardInfo/... [can be null] [may be better to fetch these information from Customer table, for
security]*/
    /* ... */
    customer_ID int(32) not null,
    location_ID varchar(32) not null,
    valet_company_ID int(32) not null, /*for paying money*/
    foreign key (customer_ID) references customer(customer_ID),
    foreign key (location_ID) references location(location_ID),
    foreign key (valet_company_ID) references valet_company(valet_company_ID)
);

create table valet_check_in
(
    valet_ID int(32) not null,
    location_ID varchar(32) not null,
    timestamp datetime not null, /*only work with MySQL*/
    valet_fee float not null,
    foreign key (valet_ID) references valet(valet_ID),
    foreign key (location_ID) references location(location_ID),
    primary key (valet_ID, location_ID, timestamp)
);

create table valet_company_located_in
(
    valet_company_ID int(32) not null,
    location_ID varchar(32) not null,
    timestamp datetime not null, /*only work with MySQL*/
    foreign key (valet_company_ID) references valet_company(valet_company_ID),
    foreign key (location_ID) references location(location_ID),
    primary key (valet_company_ID, location_ID, timestamp)
);

```

4. Transaction Summary Instructions

4.1 Transaction Query

Per Mona's request:

Transaction Query SQL statements will take the aggregate total of fee+tips for each valet company for a particular week.

Parameter to change is the TimeDate values (ie. "2015-04-06" AND "2015-04-11") to view aggregate totals for different weeks.

Transaction Query is:

```
SELECT valet_company.valet_company_name, SUM(valet_request.tips + valet_request.valet_fee)  
FROM snapvalet.valet_company, snapvalet.valet_request  
WHERE valet_request.timestamp BETWEEN "2015-04-06" AND "2015-04-11" AND  
valet_request.valet_company_ID=valet_company.valet_company_ID AND  
valet_request.valet_request_status=3  
GROUP BY valet_company_name;
```