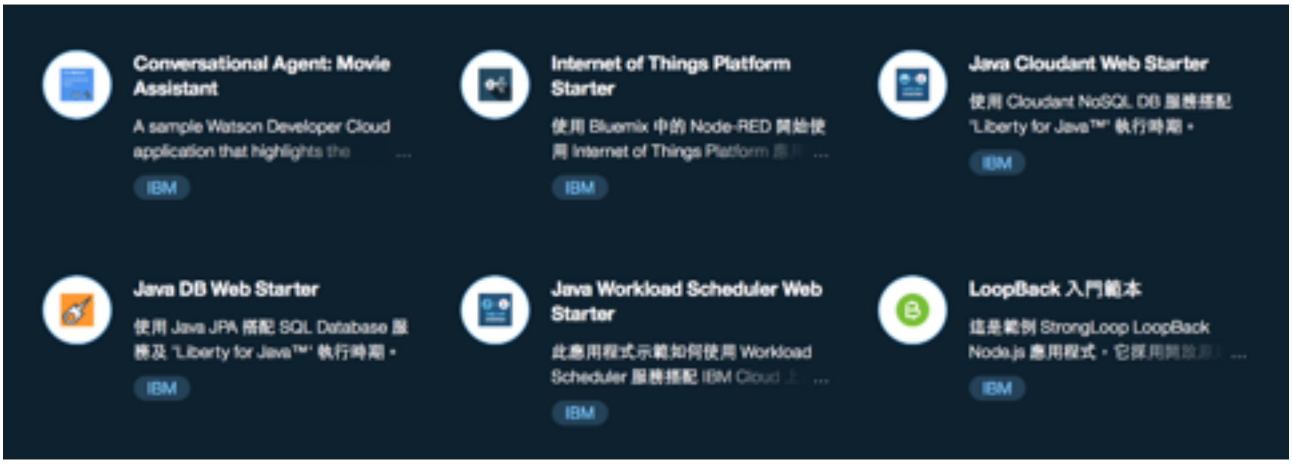
CMPE 272 mini project on bluemix

Github:
https://github.com/kingpeterson/CMPE272-FoodForNeed

Contributors:Po-Tsung Wang, Chien-Wei Lan

Thanks for IBM that provides a powerful platform for constructing application.

We choose **Java DB Web Starter** as boilerplate to build our project: Food for the need.



Conversational Agent: Movie Assistant
A sample Watson Developer Cloud application that highlights the ...
IBM

Internet of Things Platform Starter
使用 Bluemix 中的 Node-RED 開始使用 Internet of Things Platform 服... ...
IBM

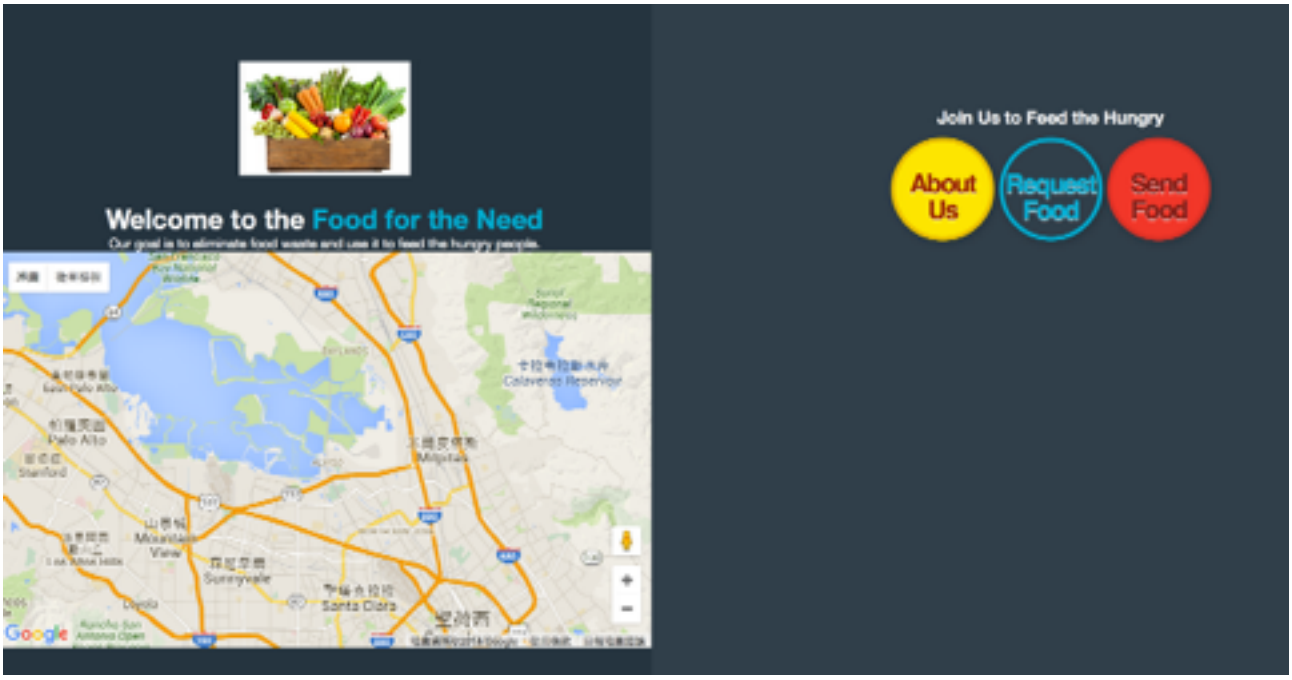
Java Cloudant Web Starter
使用 Cloudant NoSQL DB 服務搭配 'Liberty for Java™' 執行時期。
IBM

Java DB Web Starter
使用 Java JPA 搭配 SQL Database 服務及 'Liberty for Java™' 執行時期。
IBM

Java Workload Scheduler Web Starter
此應用程式示範如何使用 Workload Scheduler 服務搭配 IBM Cloud 上 ...
IBM

LoopBack 入門範本
這是範例 StrongLoop LoopBack Node.js 應用程式。它採用開... ...
IBM

The goal of our project is reducing food waste at retail level and deliver the expired food and out of date promotion goods to people or organizations who need.

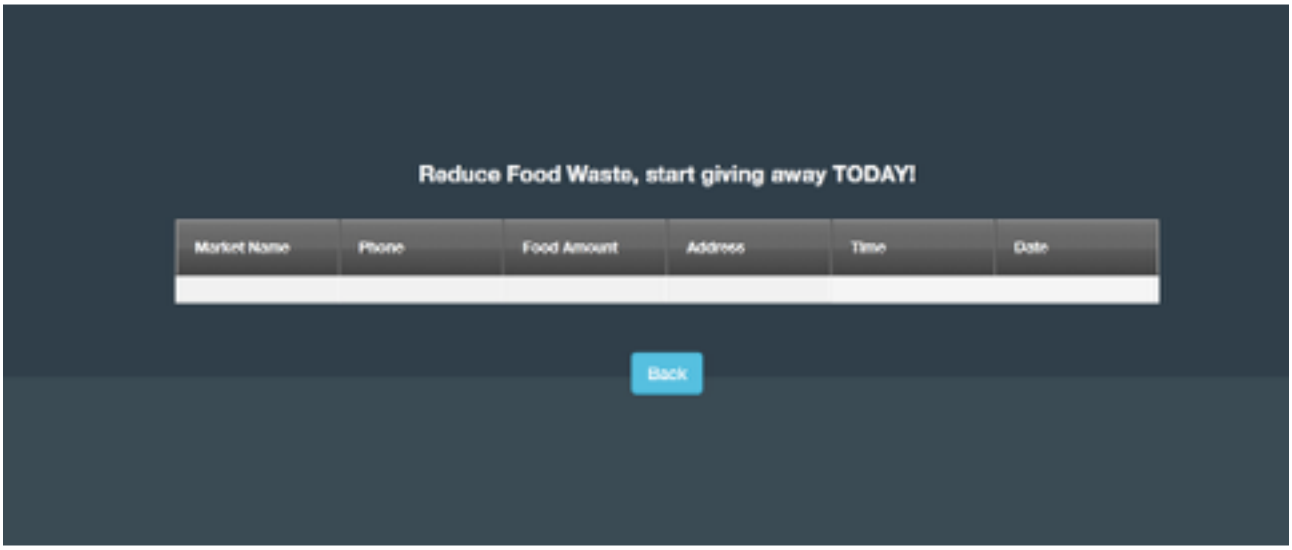


Welcome to the Food for the Need
Our goal is to eliminate food waste and use it to feed the hungry people.

Join Us to Feed the Hungry

You can donate food by just giving

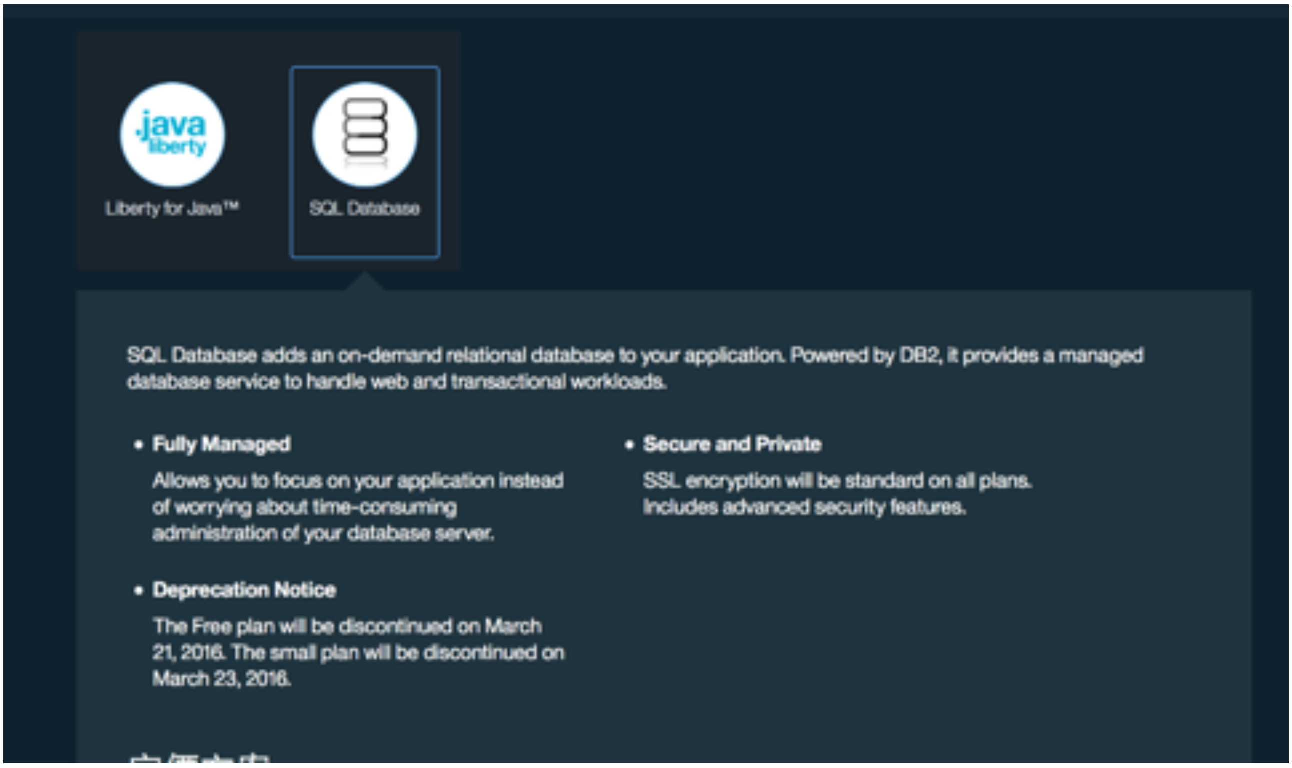
address, date and time. We will do the rest to arrange truck to deliver the food and goods. Also, you request for food, you can pick up by yourself at the grocery stores.



Reduce Food Waste, start giving away TODAY!

Market Name	Phone	Food Amount	Address	Time	Date

At the back-end, **Java DB Web Starter** provides **SQL Database** service, help us to record all the requests and donations. To combine with google map api, we can transform address to geocode and show information on the google map, an intuitive way to find and donate food by the famous location-based application!



SQL Database adds an on-demand relational database to your application. Powered by DB2, it provides a managed database service to handle web and transactional workloads.

- **Fully Managed**
  Allows you to focus on your application instead of worrying about time-consuming administration of your database server.
- **Deprecation Notice**
  The Free plan will be discontinued on March 21, 2016. The small plan will be discontinued on March 23, 2016.
- **Secure and Private**
  SSL encryption will be standard on all plans. Includes advanced security features.

Database detail:

```
CREATE TABLE market (
  marketId      INT  NOT NULL,
  marketName    VARCHAR(255) NOT NULL,
  city          VARCHAR(50)  NOT NULL,
  phone         VARCHAR(50)  NOT NULL,
  latitude      float  NOT NULL,
  longitude     float  NOT NULL,
  addressLine1  VARCHAR(50)  NOT NULL,
  addressLine2  VARCHAR(50)  DEFAULT NULL,
  state         VARCHAR(50)  DEFAULT NULL,
  country       VARCHAR(50)  NOT NULL,
  postalCode    VARCHAR(15)  NOT NULL,
  PRIMARY KEY  (marketId)
);
INSERT INTO market (marketId, marketName, city, phone, latitude, longitude, addressLine1, state,
country, postalCode) VALUES
(1, 'Safeway', 'San Jose', '(408) 292-4010', 37.334794, -121.887576, '100 S Second St',
'California', 'USA', 95113),
(2, 'Safeway', 'Santa Clara', '(408) 244-6873', 37.345185, -121.936453, '2605 The Alameda Ave.',
'California', 'USA', 95050),
(3, 'Safeway', 'San Jose', '(408) 882-0999', 37.322932, -121.911945, '1300 W. San Carlos',
'California', 'USA', 95126);

CREATE TABLE customers (
  customerId       INT NOT NULL,
  customerName     VARCHAR(255)   NOT NULL,
  contactLastName  VARCHAR(50)   NOT NULL,
  contactFirstName VARCHAR(50)   NOT NULL,
  phone            VARCHAR(50)   NOT NULL,
  latitude         float  NOT NULL,
  longitude        float  NOT NULL,
  addressLine1     VARCHAR(255)   NOT NULL,
  addressLine2     VARCHAR(255)   DEFAULT NULL,
  city             VARCHAR(50)   NOT NULL,
  state            VARCHAR(50)   DEFAULT NULL,
  postalCode       VARCHAR(15)   DEFAULT NULL,
  country          VARCHAR(50)   NOT NULL,
  PRIMARY KEY (customerId)
);
INSERT INTO customers (customerId, customername, contactlastname, contactfirstname, phone,
latitude, longitude, addressLine1, city, state, country, postalCode) VALUES
(1, 'Heritage Home','Wang', 'Peter','(408) 294-1238', 37.344251, -121.894151, '435 N 3rd St', 'San
Jose', 'California', 'USA', 95112),
(2, 'Loaves & Fishes Family Kitchen', 'Harris', 'Devon', '(408) 998-1600',37.356510, -121.846991,
'2150 Alum Rock Ave', 'San Jose', 'California', 'USA', 95116),
(3, 'Second Harvest Food Bank - Cypress Center', 'Smith', 'Josh', '(408) 266-8866', 37.413259,
-121.952596, '4001 N 1st St', 'San Jose', 'California', 'USA', 95134);

CREATE TABLE products (
  productId        INT  NOT NULL,
  productName      VARCHAR(70)  NOT NULL,
  productLine      VARCHAR(50)  NOT NULL,
  productScale     VARCHAR(10)  NOT NULL,
  productVendor    INT  NOT NULL,
  productDescription  VARCHAR(255)       NOT NULL,  -- 64KB
  quantityInStock     SMALLINT     NOT NULL,  -- Allow negative
  PRIMARY KEY (productId),
  FOREIGN KEY (productVendor) REFERENCES market (marketId)
    ON DELETE RESTRICT ON UPDATE CASCADE
);
ALTER TABLE products ADD CONSTRAINT foreign_key_name FOREIGN KEY (productVendor)
REFERENCES market(marketId) ON UPDATE RESTRICT;

CREATE TABLE productlines (
  productLine      VARCHAR(50)   NOT NULL,
  textDescription  VARCHAR(255) DEFAULT NULL,
  image            BLOB          DEFAULT NULL,  -- 64 KB
  PRIMARY KEY (productLine),
  FOREIGN KEY (productLine) REFERENCES products (productLine)
);
ALTER TABLE productLines ADD CONSTRAINT foreign_key_name FOREIGN KEY (productLine)
REFERENCES products(productLine) ON UPDATE RESTRICT;

CREATE TABLE orders (
  orderNumber     INT          NOT NULL,
  orderDate       DATE         NOT NULL,
  requiredDate    DATE         NOT NULL,
  shippedDate     DATE         DEFAULT NULL,
  status          VARCHAR(15)   NOT NULL,  -- use ENUM
  comments        VARCHAR(255)         DEFAULT NULL,
  customerId      INT NOT NULL,
  PRIMARY KEY  (orderNumber),
  FOREIGN KEY (customerId) REFERENCES customers (customerId)
);
ALTER TABLE orders ADD CONSTRAINT foreign_key_name FOREIGN KEY (customerId)
REFERENCES customers(customerId) ON UPDATE RESTRICT;

CREATE TABLE orderDetails (
  orderNumber      INT             NOT NULL,
  productId        INT       NOT NULL,
  quantityOrdered  SMALLINT          NOT NULL,  -- [0, 65535]
  orderLineNumber  TINYINT           NOT NULL,   -- [0,255]
  PRIMARY KEY  (orderNumber,productId),
  FOREIGN KEY (orderNumber) REFERENCES orders (orderNumber),
  FOREIGN KEY (productId) REFERENCES products (productId)
);
ALTER TABLE orderDetails ADD CONSTRAINT foreign_key_name FOREIGN KEY (orderNumber)
REFERENCES orders(orderNumber) ON UPDATE RESTRICT;
ALTER TABLE orderDetails ADD CONSTRAINT foreign_key_name FOREIGN KEY (productId)
REFERENCES products(productId) ON UPDATE RESTRICT;
```