cancelFood(UID, serve\_date, meal){

r = Reserve.find(UID, serve\_date, meal);

if(r == null){

return("unsuccessful");

} else {

Reserve.delete(r);

return("unsuccessful");

}

}

reserve(UID,serve\_date,meal){

is\_done = Reserve.insert(UID,serve\_date,meal);

if( is\_done )

return("successful");

else

reutrn("fail");

}

increase\_credit(UID, credit) {

u = User.find(UID);

bool = make\_bank\_transaction(credit);

if(bool){

User.update(UID, credit);

result = IncreaseCreditLog.create(UID, credit);

return ("successful");

} else {

return ("unsuccessful");

}

}

make\_bank\_transaction(amount) {

transacted = transact(amount)

if(transacted == amount) {

return true;

} else {

return false;

}

}

transfer\_credit(src\_id, dest\_id, amount) {

src = User.find(src\_id);

dst = User.find(dest\_id);

if(src.credit < amount) {

return false;

} else {

src.credit -= amount;

dst.credit += amount;

src.save();

dst.save();

return true;

}

}

getActivityReport(start\_date, end\_date, UID, reportType){

if(reportType == increase){

result = IncreaseCreditReport.findMatching(UID, start\_date, end\_date);

} else {

result = TransportCreditReport.findMatching(UID, start\_date, end\_date);

}

return result;

}

getFoodSchedule(UID, week\_start\_date, week\_end\_date){

u = User.find(UID);

g = u.gender;

result = Serve.findMatching(UID, start\_week\_date, end\_week\_date, g);

return result;

}

login ( UID , password ){

valid = User.find ( UID , password );

if (valid)

return ("successful") ;

else

return ("fail") ;

}

logout(UID){

u = User.find(UID);

if(u == null){

return("logout unsuccessful!");

} else {

return("logout successful!");

}

}

registerUser(UID, password, name, gender){

u = User.find(UID);

if(u == null){

User.insert(UID, password, name, gender);

return("successfully registered");

} else {

return("user exists!");

}  
}

determine\_food(foodName,price){

f = Food.find(foodName);

if(f == null){

Food.insertFood(foodName,price,false);

return ("new food inserted successfully");

} else {

Food.updateFood(foodName,price,false);

return ("food updated successfully");

}

}

determine\_food\_paln(){

Serve.insertFood(foodName,date,meal);

if (Food.has(foodName)){

return (“done!”);

} else {

return (“failed to do so!”);

}

}

get\_food\_statistics( serve\_date, meal , gender ){

food\_status\_table = Serve.findMatching( serve\_date, meal , gender );

return ( food\_status\_table );

}

food\_statistics\_search( meal , serve\_date, gender){

reserve\_table = Reserve.findMatching(meal , serve\_date, gender);

retuen ( reserve\_table );

}

approve\_food\_price(listOfFoodRequests[]){

foodPrices = Food.findPrice(listOfFoodRequests[]);

foreach(food : listOfFoodRequests){

if(Exist){

appropriatePrice = getAppropriatePrice(foodName);

if(appropriatePrice == foodPrice){

Food.updateApproved(foodName);

}

} else {

return false;

}

}

}