cancelFood (UID ,serve\_date , meal){

u = User.find(UID);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

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

if (reserved\_food == null){

return("unsuccessful");

} else {

Reserve.delete(reserved\_food);

return("successful");

}

}

reserveFood (UID ,serve\_dat e, meal){

u = User.find(UID);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

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

if (is\_done){

return("successful");

} else {

return("unsuccessful ");

}

}

increaseCredit (UID , credit) {

u = User.find(UID);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

Is\_done = makeBankTransaction(credit);

if( is\_done){

u.credit += credit ;

User.update(UID);

IncreaseCreditLog.create(UID, credit);

return ("successful");

} else {

return ("unsuccessful");

}

}

transferCredit (src\_id , dest\_id , amount) {

u = User.find(src\_id);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

src = User.find(src\_id);

dst = User.find(dest\_id);

if ( src == null || dst == null){

return ("unsuccessful");

} else if ( src.credit < amount) {

return ("unsuccessful");

} else {

src.credit -= amount;

dst.credit += amount;

User.update(src);

User.update(dst);

TransferCreditLog.insert (src, dst , amount);

return ("successful");

}

}

getActivityReport(UID , start\_date, end\_date , report\_type){

u = User.find(UID);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

if (reportType == increase){

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

} else {

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

}

return result\_table;

}

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

u = User.find(UID);

u.last\_activity = getCurrent\_date\_time();

User.update(u);

g = u.gender;

result = Serve.findMatching(UID, week\_start\_date, week\_end\_date, g);

return result;

}

login ( UID , password ){

u = User.find ( UID );

if (u.password == password){

u.is\_login = true ;

u.last\_activity = getCurrent\_date\_time();;

User.update(u);

return ("successful") ;

}else{

Return(“unsuccessful”)

}

}

logout(UID){

u = User.find(UID);

if(u != null && u.is\_login){

return("successful!");

} else {

return("unsuccessful!");

}

}

registerUser(UID, password, name, gender){

u = User.findAdmin()

u.last\_activity = getCurrent\_date\_time();;

if( u != null ){

User.update(u);

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

Return(“successful”)

} else{

Return (“unsuccessful”)

}

}

determineFood(food\_name,price){

u = User.findTheKitchen()

u.last\_activity = getCurrent\_date\_time();

f = Food.find(food\_name);

if(f == null){

Food.insert(food\_name,price,false);

return ("new food inserted successfully");

} else {

Food.update(food\_name,price,false);

return ("food updated successfully");

}

}

determineFoodPlan(food\_name , serve\_date , meal){

u = User.findTheKitchen()

u.last\_activity = getCurrent\_date\_time();

f = Food.find(food\_name);

if ( f == null ) {

return("unsuccessful!");

} else{

Serve.insert(food\_name, serve\_date,meal);

return("successful!");

}

}

getFoodStatistics( serve\_date, meal , gender ){

u = User.findTheKitchen()

u.last\_activity = getCurrent\_date\_time();

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

return ( food\_status\_table );

}

foodStatisticsSearch( meal , serve\_date, gender){

u = User.findTheKitchen()

u.last\_activity = getCurrent\_date\_time();

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

retuen ( reserve\_table );

}

approveFoodPrice(list\_of\_food\_requests[]){

u = User.findAdmin()

u.last\_activity = getCurrent\_date\_time();

foodPrices = Food.findPrice(listOfFoodRequests[]);

index = 0

foreach(food : listOfFoodRequests){

index++

exist = Food.find(food)

if(exist){

appropriatePrice = getAppropriatePrice(foodName);

if(appropriatePrice == foodPrice){

Food.updateApproved(foodName);

} else {

Result\_list[i] = “unsuccessful”

}

}

Return(result\_list)

}