Systems Analysis

and Design

Instructor : Huang, Chuen-Min

**Teamwork2 ver.1**

Group 1

|  |  |
| --- | --- |
| ID | Name |
| A10523013  B10423013 | Miko  Tiffany |
| B10423038 | John |
| B10423041 | Dan |
| B10523009 | Jerry |
| B10523024 | Steven |
| B10523048 | Andy |
| B10523062  B10523063 | Ken  Patrick |
| Date 2018/05/29 | |

Content

**1. Please explain the Law of Demeter (LoD) by using of your project. 3**

**2. coupling 6**

**3. cohesion8**

**4. connascence10**

**5. CRC card or the class diagram12**

**6. Structured English and an activity diagram to specify the algorithm.14**

**7. Evaluate project in terms of cohesion, coupling, and connascence perspective16**

**8. Adopt RDBMs to your project, please describe the referential integrity18**

**9. Using the steps of normalization, create a model19**

**10. Draw the new class diagram based on your suggested changes24**

**11. Clustering、Index26**

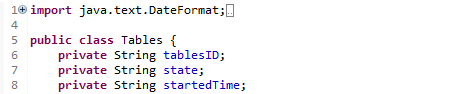
**12. Teamwork Responsibility29**

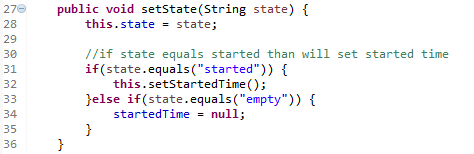
1. Please explain the **Law of Demeter (LoD)** by using of your project.

**Lod1**

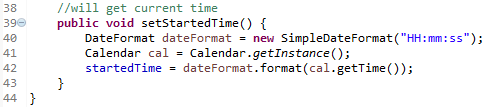
**(1) to itself (O itself)**

**Class Tables**





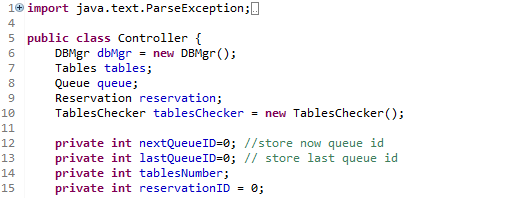
Call method defined in this class to set started time.



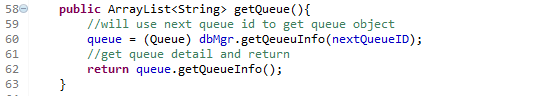
**Lod2**

**(2) to objects contained in attributes of itself or a superclass (Any objects created/instantiated within M)**

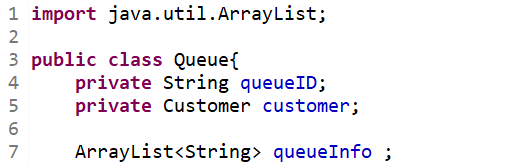
**Class Controller**

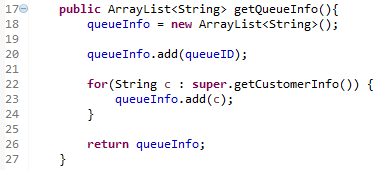


Controller has queue’s attribute of object and then use method in queue class



**Class Queue**

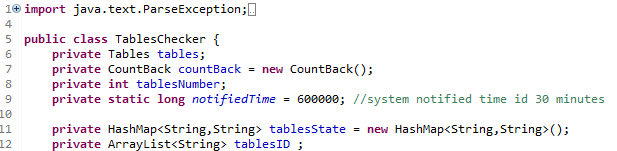


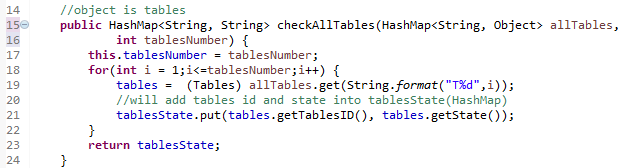


**Lod3**

**(3) to an object that is passed as a parameter to the method (M’s parameters)**

**Class TablesChecker**



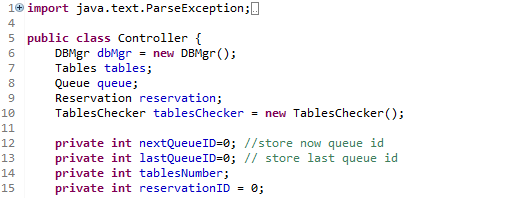


checkAllTables will get HashMap(object) and use it.

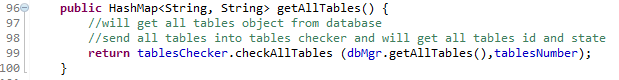
**Lod4**

**(4) to an object that is created by the method (O's direct component objects)**

**Class Controller**



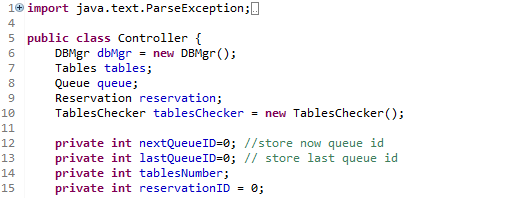
Controller create tablesChecker and use checkAllTables() method to get table id and table state.

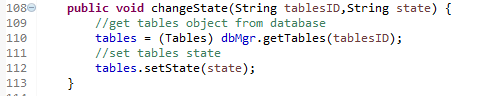


1. There are six (or seven) types of interaction coupling, each falling on different parts of a good-to-bad continuum. Choose three pieces of your project to describe what types of the coupling they belong to.

**2.1 data**

**Class Controller**

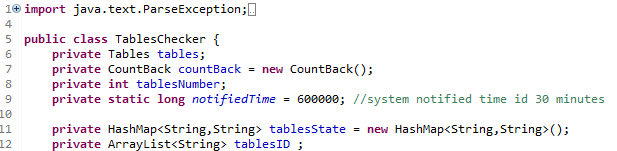


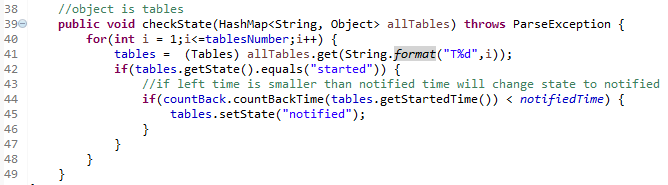


changeState will get tableID and state as parameter, and will use them.

**2.2 stamp**

**Class TablesChecker**

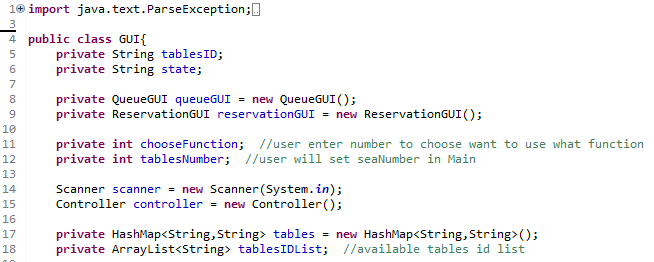


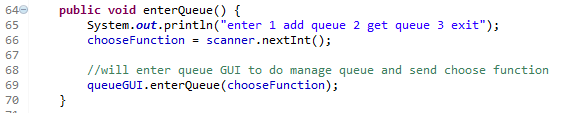


checkState() will get HashMap(object) and use part method get().

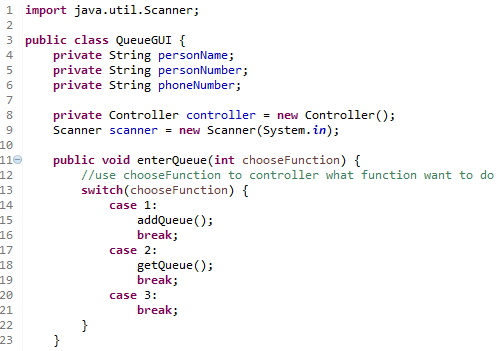
**2.3 control**

**Class GUI**





**Class QueueGUI**

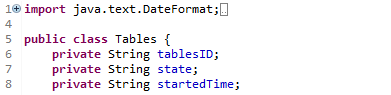


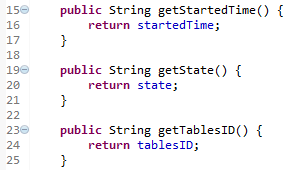
enterQueue() get chooseFunction, different chooseFunction will execute different method.

1. There are seven types of method cohesion, choose three pieces of your project to describe what types of the cohesion they belong to.

**3.1 funtional**

**Class Tables**

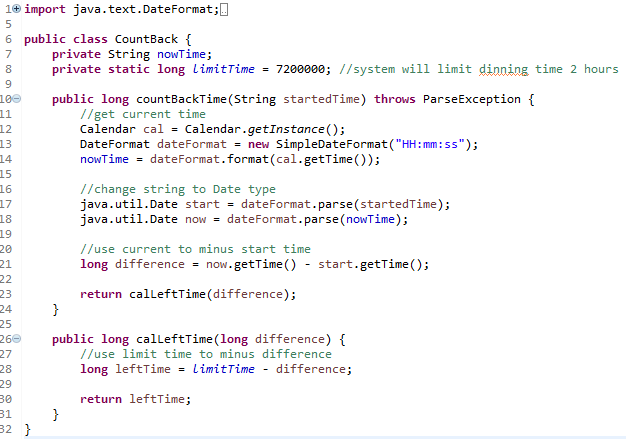




There are three method, those method performs a single problem-related task.

**3.2 sequential**

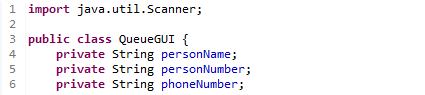
**Class CountBack**

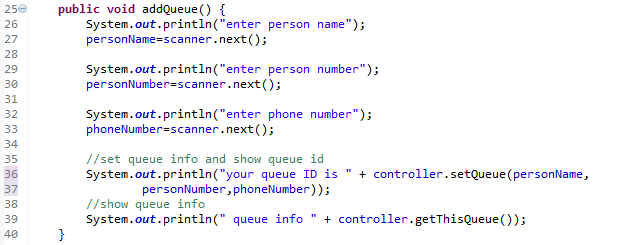


Different method combines two functions in which the output from the first one in used as the input to the second one.

**3.3 procedual**

**Class QueueGUI**



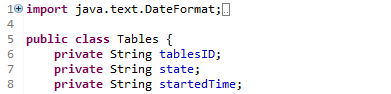


Call controller’s setQueue() will get queueID, and call controller’s getThisQueue() will get queue info, they both related to queue but they are weakly related functions.

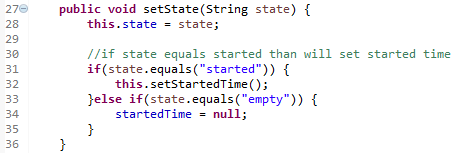
1. Connascence generalized the ideas of cohesion and coupling, use three pieces of your project to describe what types of the connascence they belong to.

**4.1 name**

**Class Tables**

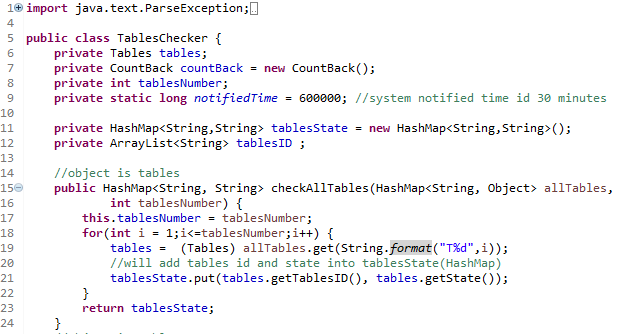


setState() will get state, and if state name change, then any use state have to change name.



**4.2 type**

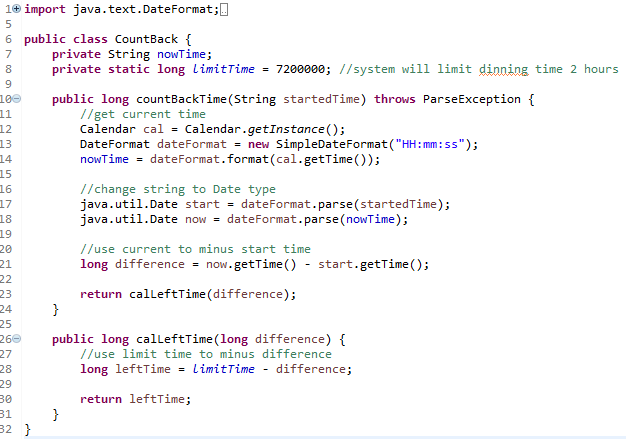
**Class TablesChecker**



If checkAllTables’s parameter of type changes, the parameter declaration will have to change.

**4.3 Convention**

**Class CountBack**

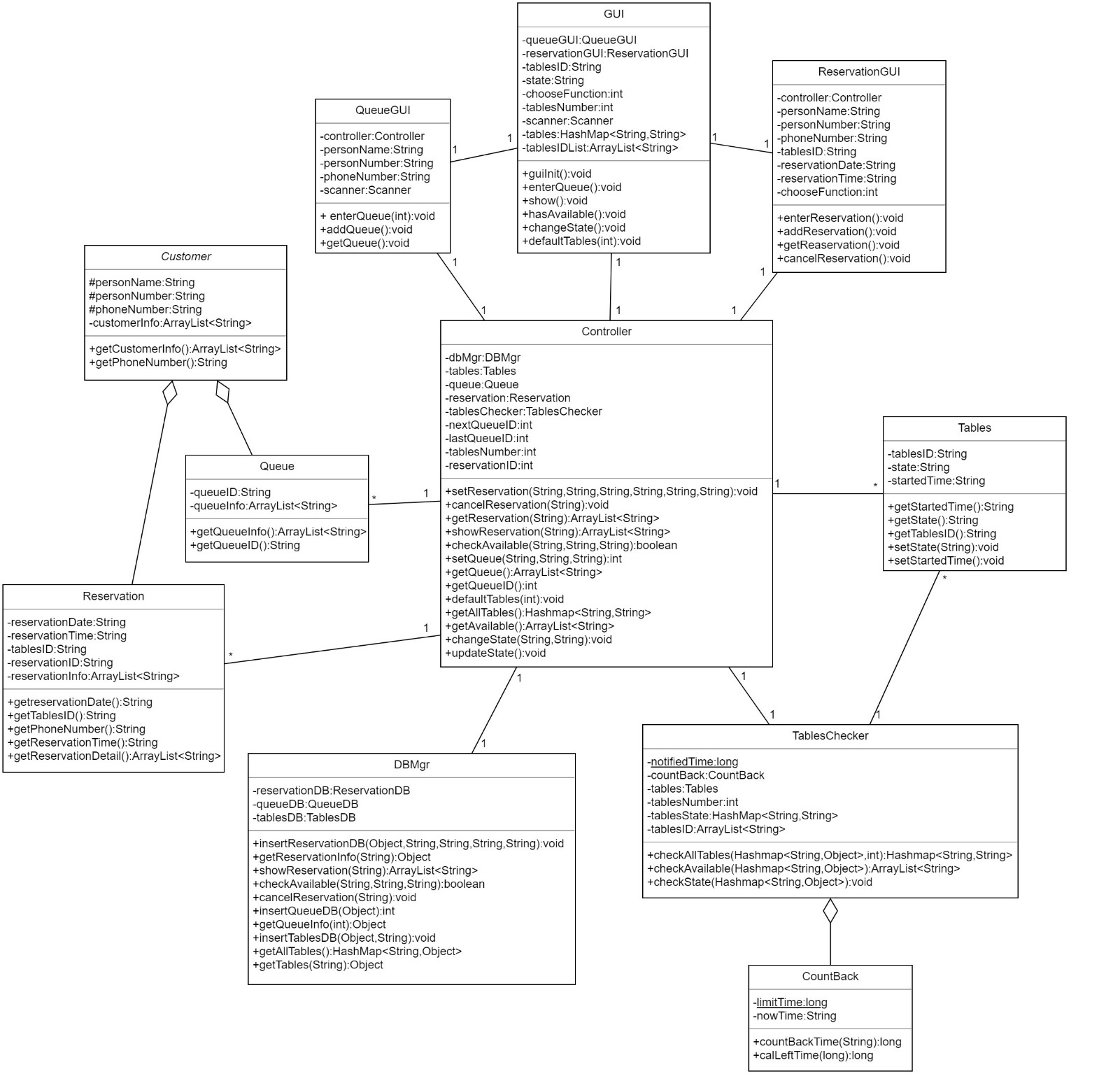


If limitTime number change, will affect other use this attribute.

1. Use one class from your project that can create a set of invariants and add them to the CRC card or the class diagram.

|  |  |  |  |
| --- | --- | --- | --- |
| Class Name: TablesChecker | ID:2 | | Type: Concrete, Domain |
| Description: If user want to show all tables state will use checkAllTables method and if user want to show available tables will use checkAvailable method and our system will update tables object every 6 seconds and will use checkState method. | | | Associated Use Cases: Arrange Tables |
| Respossibilites | | Collaborators | |
| CheckAllTables  CheckAvailable  CheckState | | Tables  Talbes  Tables, CountBack | |

|  |
| --- |
| Attributes:  TablesID(1..1)(String) {TableID = Tables.getTablesID()}  TablesState(1..1)(String) {TablesState = Tables.getState()}  TablesNumber(1..1)(Int)  Tables(1..1)(Tables)  NotifiedTime(1..1)(Long)  CountBack(1..1)(CountBack) |
| Realationship:  Generalization(a-kind-of):  Aggregation(has-parts): CountBack(1..1)  Other Association: Tables(1..1) |

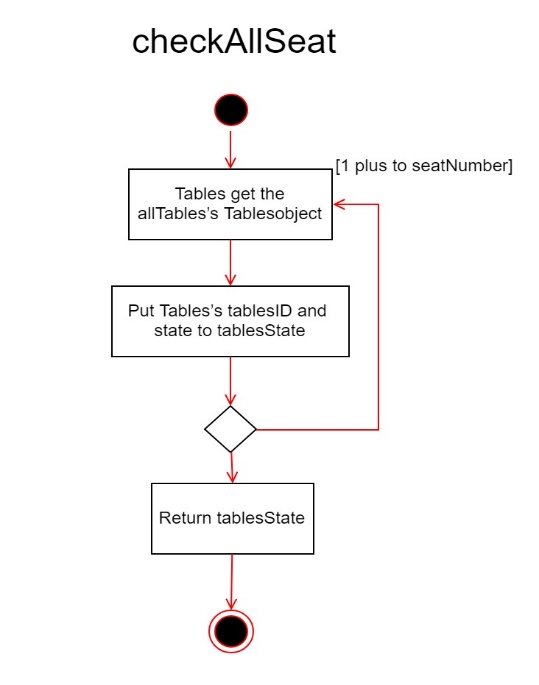
Class Diagram

1. Use a method of a class from your project that can create a contract and describe its algorithm specification. Specify the pre- or post- condition and use both Structured English and an activity diagram to specify the algorithm.

|  |  |  |
| --- | --- | --- |
| Method Name: checkAllTables | Class Name: TablesChecker | ID:2 |
| Clients(consumers): Controller | | |
| Associated Use Cases:  Arrange Tables | | |
| Description of Responsibilities:  It will get all allTables(HashMap) include tables id(key) and tables object(value), will use tables object to get tables id and state, use seatState(HashMap) to store tables id(key) and state(value), will return seatState(HashMap). | | |
| Arguments Received:  allTables:HashMap<String,Object>,tablesNumer:int | | |
| Type of Value Returned:  tableState:Hashmap<String, String> | | |
| Pre-Conditions:  TablesDB.includes(Tables) | | |
| Post-Conditions:  tableState.sum(Tables.getTablesID(), Tables.getState()) | | |

Method Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method Name: checkAllTables | Class Name: TablesChecker | | | ID:1 |
| Contract ID:2 | Programmer:Tiffany | | Data Due: 05/22/18 | |
| Programming Language:  Java | | | | |
| Triggers/Events:  User wants to show all tables state. | | | | |
| Arguments Received:  Data Type: | | Notes: | | |
| Hashmap<String, Object> | | allTables(tablesID,Tables Object) | | |
| String | | tablesNumber | | |
| Messages Sent & Argument Passed:  ClassName.MethodName: | | Data Type: | Notes: | |
| tables.getTablesID() | | String |  | |
| tables.getState() | | String |  | |
| Arguments Returned:  Data Type: | | Notes: | | |
| Hashmap<String,String> | | tablesState(tablesID,tablesState) | | |
| Algorithm Specification:  For 1 plus to tablesNumber  Tables get the allTables’s Tables object  Put Tables’s tablesID and state to tablesState  Return tablesState | | | | |
| Misc.Notes:  None | | | | |

****

1. Please evaluate any piece of your project in terms of cohesion, coupling, and connascence perspective.

**Coupling-**

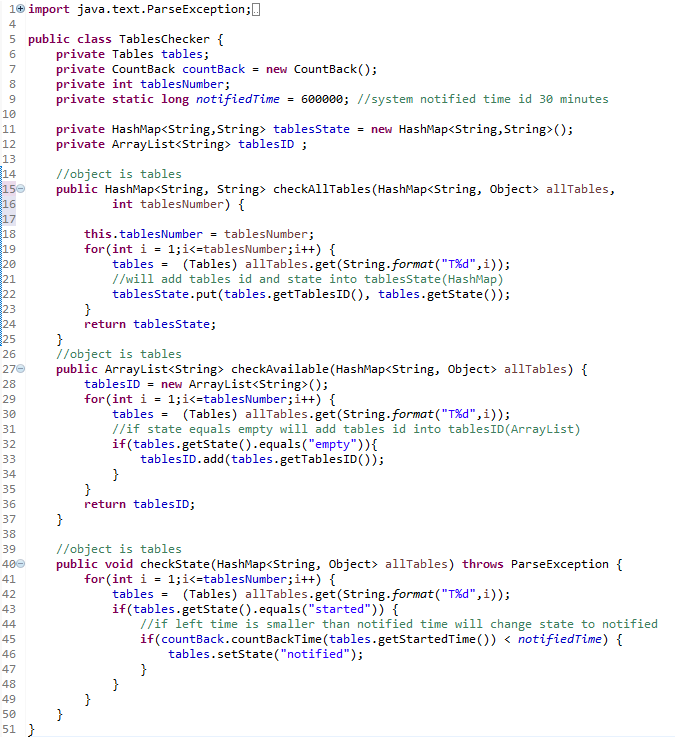
In TablesChecker class we just use stamp, data coupling, it’s best two coupling.

**Cohesion-**

In Tables class, all method is functional cohesion, it’s the best cohesion.

**Connascence-**

In TablesChecker class, if method received parameter’s type change, which use this parameter have to change, and TablesChecker define static long notifiedTime, if the value change, it will affect which use this attribute.



convention

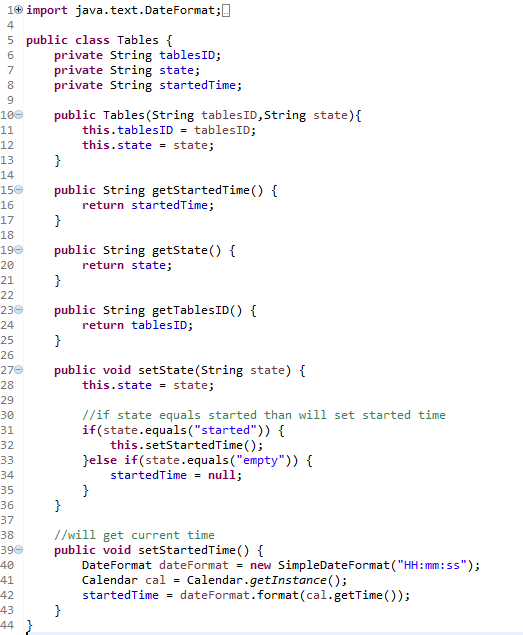
stamp

stamp

stamp

data

stamp



functional

functional

functional

functional

functional

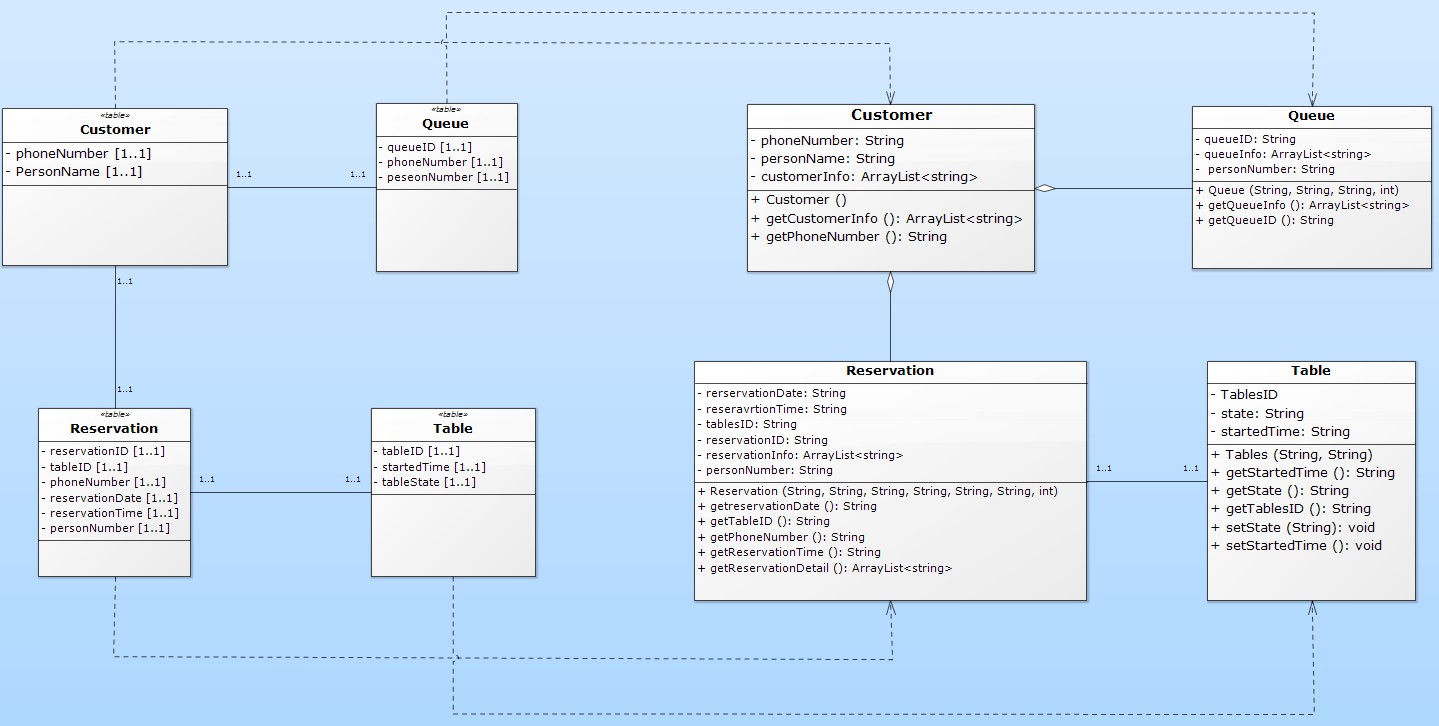
1. Assume that you are going to adopt RDBMs to your project, please describe the referential integrity.

Rule 1.Map all concrete-problem domain classes to RDBMS tables. Also, if an abstract Problem Domain class has multiple direct subclasses, map the abstract class to a RDBMS table

Rule 2.Map single-valued attributes to columns of tables

Rule 4.Map single-valued aggregation and association relationships to a cloumn that can store the key of the related table,i.e.,add foreignkey to the table.Do this for both sides of the relationship.

Problem Domain Class



RDBMS Tables

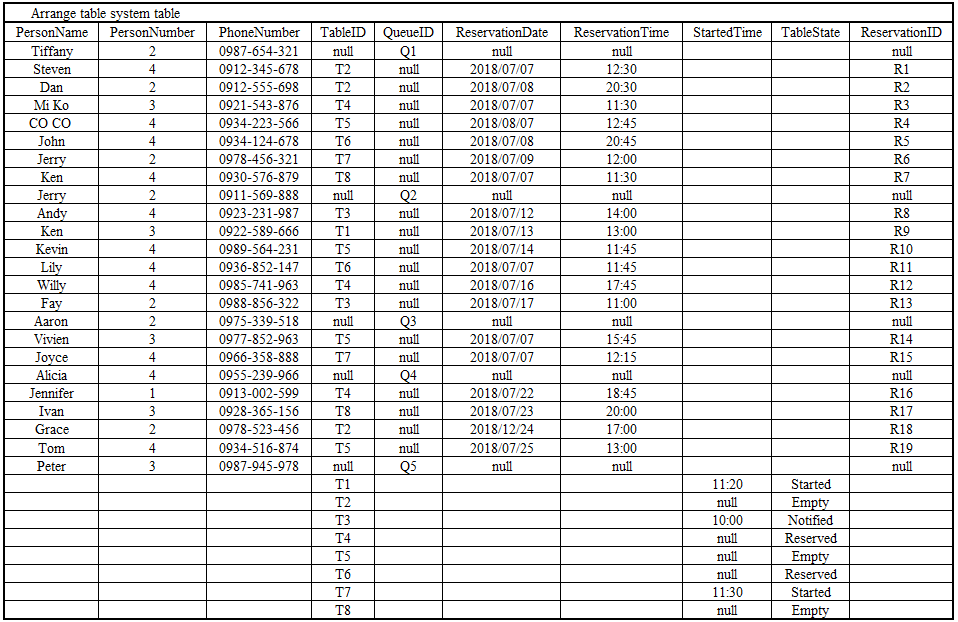
1. Using the steps of normalization, create a model that represents the file of your project in third normal form. Please make necessary assumptions to explain why the tables are related.

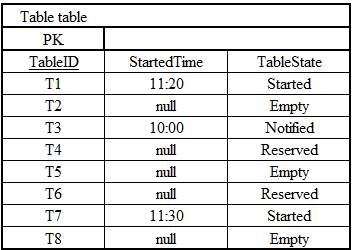
|  |
| --- |
| Zero Normal Form Class Diagram |
| C:\Users\miko1\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Classdiagram1.png |

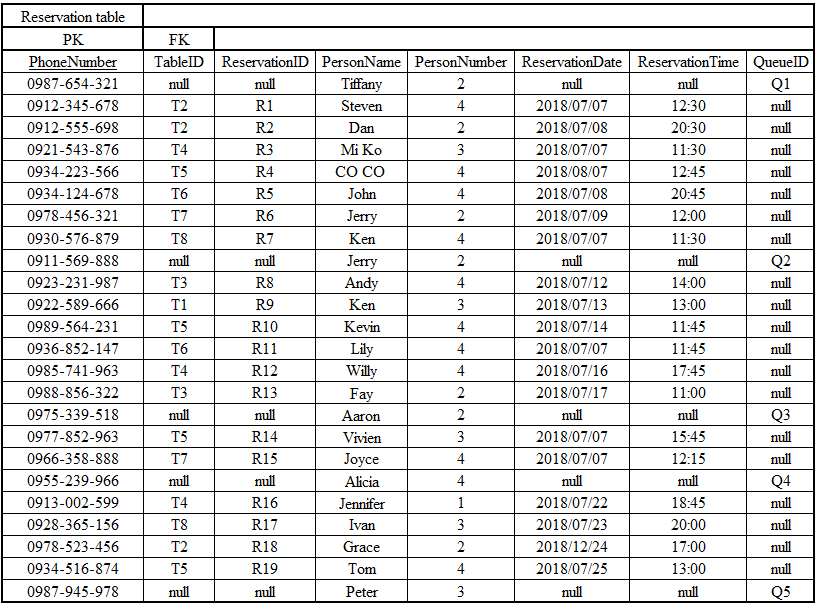
|  |
| --- |
| First Normal Form Class Diagram |
|  |

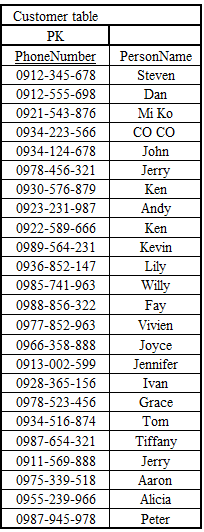
|  |
| --- |
| Second、Third Normal Form Class Diagram |
|  |

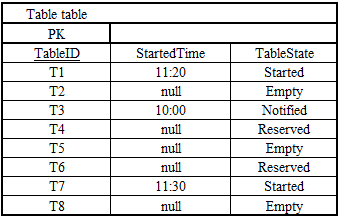
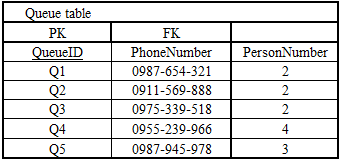
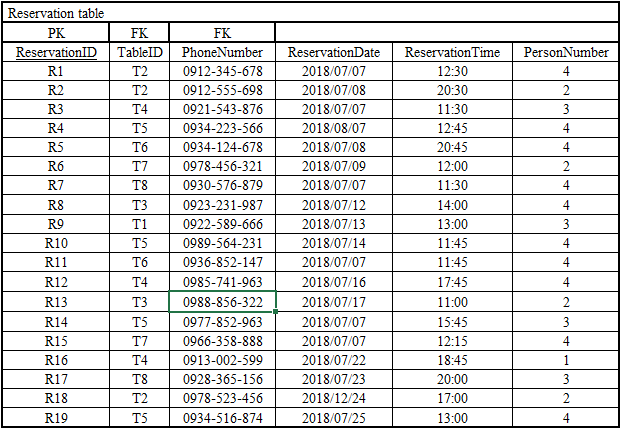
NF0



NF1

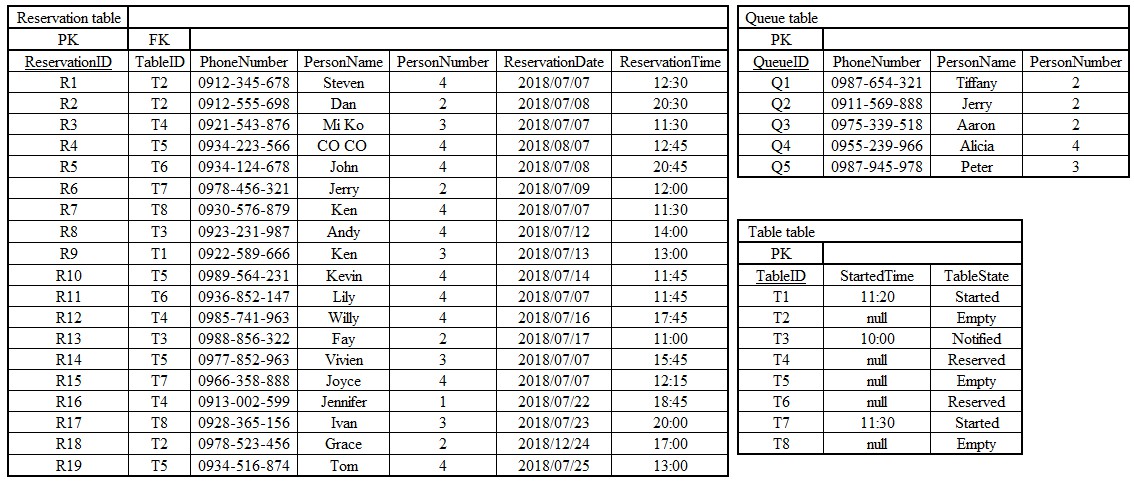


NF2、NF3

When the second normalization is performed, part of the dependency is removed. After the second normalization found out that the data table did not have a transitive dependency, so when the data table do the second normalization, it is also the third normalization.

1. Describe how you would denormalize the model that you created in question 9. Draw the new class diagram based on your suggested changes.

When we are searching booking and queuing information, if we want to know the personal information and the number of people that customer gives us, must connect to the customer information table through the telephone number before we can know. However, every time you seach it, you must do the same action. Therefore, we decided to use Denormalization to integrate the customer information table into the queuing and the reservation table.

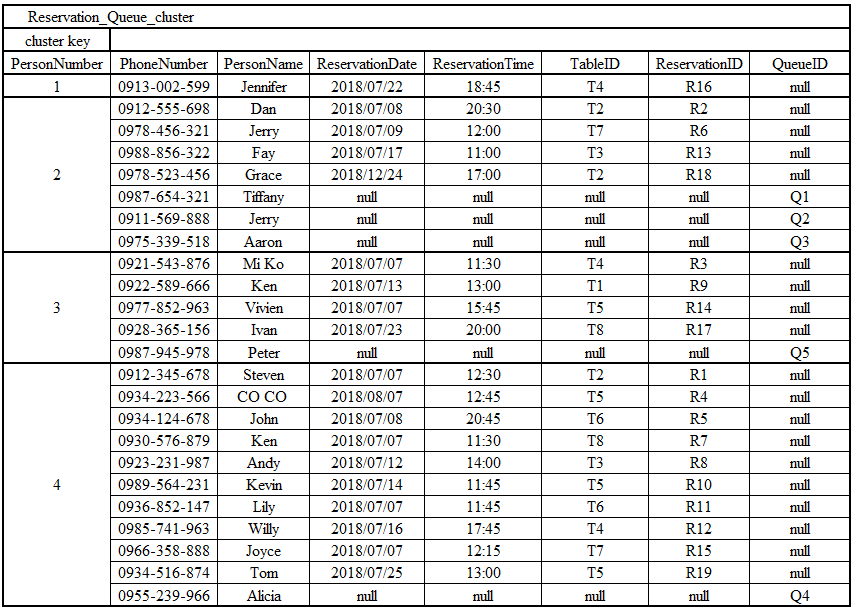


|  |
| --- |
| Class Diagram |
|  |

1. Examine the model that you created in question 10. Develop the inter-file clustering and index strategies. Describe how your clustering strategy will improve the performance of the database. List possible indices you would recommend and describe the reasons

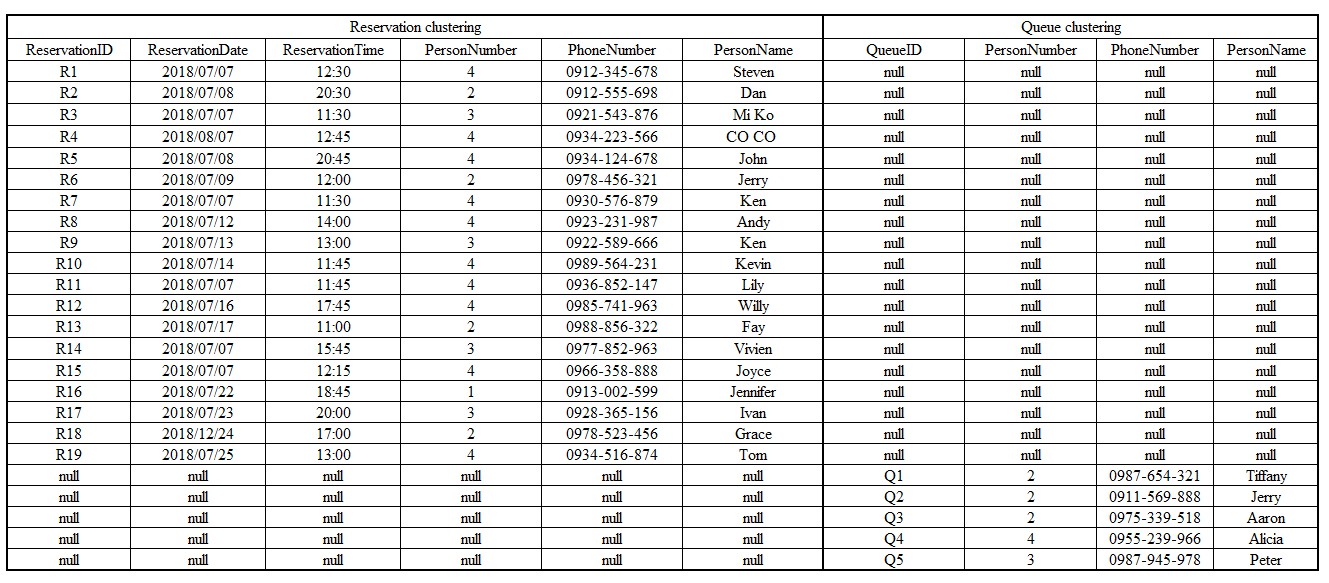
Intra – file Clustering

We use PersonNumber as our cluster\_key, which reduces time we spend on storing data.



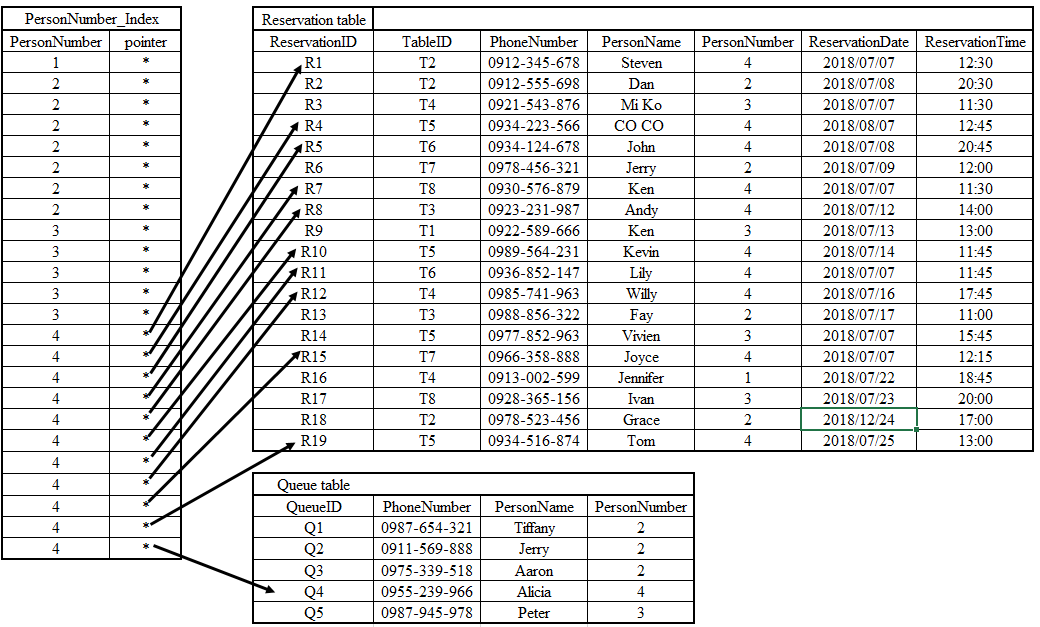
inter-file clustering

When we make a reservation, the system will also save the information from the customer spontaneously. And, so is when we wait in the line. Therefore, we make two cluster out of the situations.



Index

We used PersonNumber as our Index to let our system to seach the data of customers faster



# 12.Teamwork Responsibility：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Name | Responsibility | Percentage |
| A10523013 | Miko | Q8、Q9、Q10、Q11 | 100% |
| B10423013 | Tiffany | Q1、Q2、Q3、Q4、Q5、Q6、Q7 | 100% |
| B10423038 | John | Q1、Q2、Q3、Q4 | 100% |
| B10423041 | Dan | Q8、Q9、Q10、Q11 | 100% |
| B10523009 | Jerry | Q5、Q6、Q7 | 100% |
| B10523024 | Steven | Q1、Q2、Q3、Q4 | 100% |
| B10523048 | Andy | Q5、Q6、Q7 | 100% |
| B10523062 | Ken | Q1、Q2、Q3、Q4 | 100% |
| B10523063 | Patrick | N/A | 0% |