Software Engineering Assignment 3 – Team 1

Analysis Classes

Entity and Boundary classes for relevant Use Cases -

Phillip Witkin

Asad Kabir

--- Entity classes ---

|  |  |  |
| --- | --- | --- |
| **Entity Class:** | **Attributes** | **Description** |
| Table | Table\_ID  Server\_Section  Maximum\_Occupancy  Status | From the views of table layout, described below, individual tables can be selected for each type of employee’s relevant actions.  The Server has the most involvement with and individual table. It is through a table that Orders are taken, served to customers, and eventually paid. Server\_section and Order\_status are relevant attributes.  The Hostess can interact with a specific table when seating new customers. The availability attribute is directly involved.  The Bussboy can select an individual table to be cleaned, and use his view of the table to clean it. A combination of Occupancy Availability manage the status regarding if table has been cleaned.  The Manager can potentially view any of this information for any Table in the restaurant. This allows them to see specific information, and perform some actions regarding specific tables.  The table is a ubiquitous and persistent element of these employees using RAS, and most important staff actions are directly associated with a table. |

|  |  |  |
| --- | --- | --- |
| **Entity Class:** | **Attributes** | **Description** |
| Table Layout | Number\_Of\_Tables  Tables[] | The Table Layout class contains a collection of all of the individual Table objects. It will hold very little additional information aside from the number of tables (which might be redundant, since that can be determined by the collection of tables, but might serve a purpose if individual tables are linked together) |

--- Boundary Classes ---

|  |  |  |
| --- | --- | --- |
| **Boundary Class:** | **View** | **Description** |
|  | Table Layout View | The table layout represents an initial point of contact for the staff upon logging in, which serves as a basic interface throughout using RAS. The Server, Hostess, Bussboy and Manager each have specialized views of the Table Layout.  From there, individual tables can be selected. |
| Server Interface | Table Layout View-Server specialization  **ViewTable**  Order related cases   * (team 2) | The Server has a view of the table layout emphasizing which tables are in their sections, and those tables whose orders are ready. |
| Hostess Interface | Table Layout View-Hostess specialization:  **CheckTableAvailability**  **SeatTable,**  Waitlist and Reservation cases (team 3) | The Hostess has a view which shows which tables are available for seating.  It also provides the ability to adjust the Reservations and Waitlist |
| Bussboy Interface | Table Layout View-Bussboy specialization:  **CleanTable** | The Bussboy has a view which shows tables which are empty but dirty and need to be cleaned. |
| Manager Interface | Table Layout View-  Manager specialization:  **ViewTable**  **AddTable,**  **RemoveTable,**  **CreateServerSection,**  **ModifySeating,**  AdjustCheck,  Add Employee  Remove Employee  Menu Cases (team 3) | The Manager can potentially view any of the information available to other employees, as well as perform their actions.  They can also select some special actions regarding Table configuration, such as assigning them to a server section, and adding or removing the number of tables in the restaurant.  Employee management is also an option from their basic interface. |

--- Control Classes ----

|  |  |  |
| --- | --- | --- |
| **Control Classes:** | **Actor:** | **Description** |
| **ViewTable** | Server,  Manager | Allows the server to see the details regarding a specific table. It brings up all information about the table and party currently seated there, including the Order associated with a Table, if there is one. If not, an Order can be submitted.  When initiated by the manager, it provides the same functionality, but also allows for the check for an order to be modified. |
| **CheckTableAvailability** | Hostesss | Shows the hostess a list of available tables to seat a new party. Allows the hostess to filter based on the number of people to be seated. |
| **SeatTable,** | Hostess | Allows the Hostess to select a specific table, and seat a party of customers there. It changes the status of the table, so that a Server whose section includes that table is aware that a new party has been seated, and must be attended to, so that an Order can be placed. |
| **CleanTable** | Bussboy | Allows the busboy to select a dirty table and clean it, so that it is available for seating. |
| **AddTable** | Manager | Adds a new table to the restaurant’s floor.  It can then be assigned to a server. |
| **RemoveTable** | Manager | Removes a table from the restaurant’s floor. |

Table Management Subsystem

The table subsystem is used directly by all the employees other than the cook. It is utilized to generate the basic interfaces that they all use to perform their duties, when they see a table layout. The Hostess needs to interact with it in order to CheckTableAvailability, and then to use SeatTable. The Bussboy also uses it to CleanTable, as well as see what tables need to be cleaned.

The Manager uses it to AddTable and RemoveTable, and the Bussboy uses it to CleanTable.

**Services offered:**

* The Employee Management subsystem:
  + CreateServerSections
  + ModifySeating.

When it does this, it offers services to the Employee Management subsystem, because the server sections are associated with a server.

* Order subsystem:
  + Because every Order is associated with a Table. The server is the main actor who uses the Order subsystem
  + ModifyCheck

The manager must interact with the Table Management subsystem if they ModifyCheck, since the check is associated with a Table as well as an Order

**Services requested:**

* Order subsystem:
  + ViewTable

Both the Manager and Server can use it to ViewTable, which might have an Order associated with it



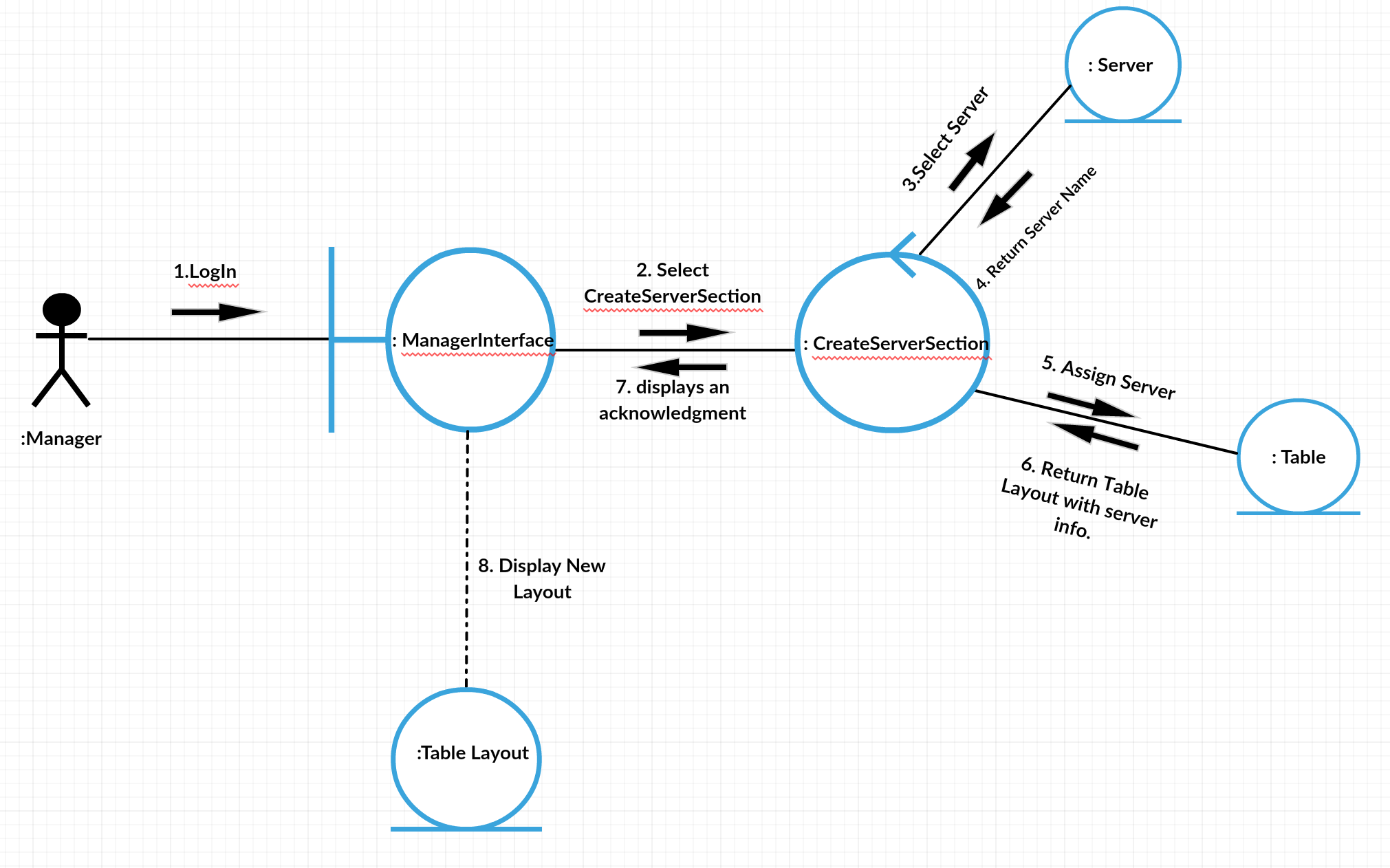




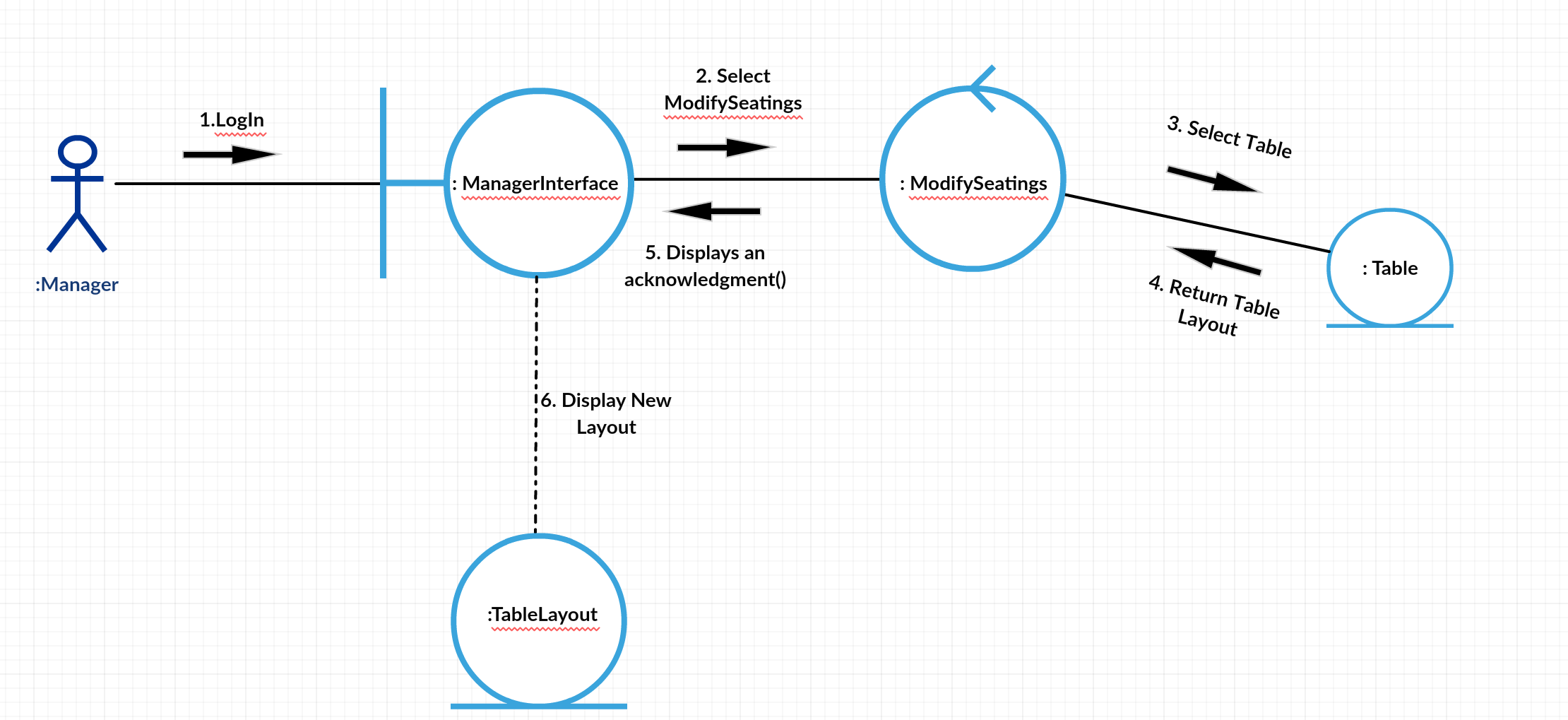




|  |  |  |
| --- | --- | --- |
| **Control Classes:** | **Actor:** | **Description** |
| **CreateServerSection,** | Manager | Groups a set of tables together, marking them as belonging to the same section associated with a specific Server. This indicates that the same Server will be responsible for managing the orders of all the tables in that section. |
| **ModifySeating,** | Manager | Allows the manager to combine or tables to accommodate larger parties. Tables which have been merged can also be split up again. |

CreateServerSection Collaboration diagram

Asad Kabir – Control Classes (Entity and Boundary class above)

ModifySeating Collaboration Diagram

Christina Sadhu

Software engineering

***Entity class: Attribute***  ***Description***

Employee Employee\_id For the employees, when they come in to work

Log in/ log out they have their employee id to clock in and out

Email of the system. By using their employee id at work

Name that’s how they are identified to do everything

Phone number that Is needed to be accomplished at work.

***Boundary class: view*** ***Description***

Manager interface: Manager\_id once the manager logs in to the system he/she

Employee\_id will be able to have access to all the employees

View login/logout information such as their employee id and when

Employees availability they have logged in and out. Also, it will allow the manager to see what employee has available time for them to attend to another customer or if he/she is short staff and need more people to come in for extra help in the restaurant

***Control class Description***

Add staff – new employees new profile After managing the employee’s id and their

New employee id hours (when the clock in and out at work) the manager would be able to overlook all the information when he/she logs in to the system. They will be able to control all the employee’s information if needed. Also, the manager can also add new employees to the restaurants and also assigned them employees ids and the necessary training that they need for the restaurants

Remove staff ***–***employee’s termination, once the new employees are added the

Remove from the system completely manager also has to remove all of the old employees’ id and profile from the system because they no longer work there. The new employees have replaced the old employees’ positions and the unnecessary information is not needed in the system anymore.

Check staffing- employee\_Id once the manager logs in to the

Name system they gain access on the

Email entire restaurant. They are also

Able to check on their staff and make sure

They are doing the necessary jobs that they were assigned to. When the manager logs in he would be able to check on the staff by seeing who clocked in and out by seeing their employee id with their name in the system

***Employee management subsystem***

Along with the server, the manager also needs to have access to all of the employee’s information, check on the staff to see who is here and working and make sure the system is updated with all of the new employee’s information and have the old employee’s information’s removed from the system. The manager also needs to view the entire work space so he/she can assist other employees if help is needed somewhere in the restaurant. This will help run the restaurant in a much better needed way because they then have the necessary number of employees to seat and attend to customers as quickly as possible. It also helps the managers to add new staff if it is needed and remove staff that is not to any use for the restaurant.

