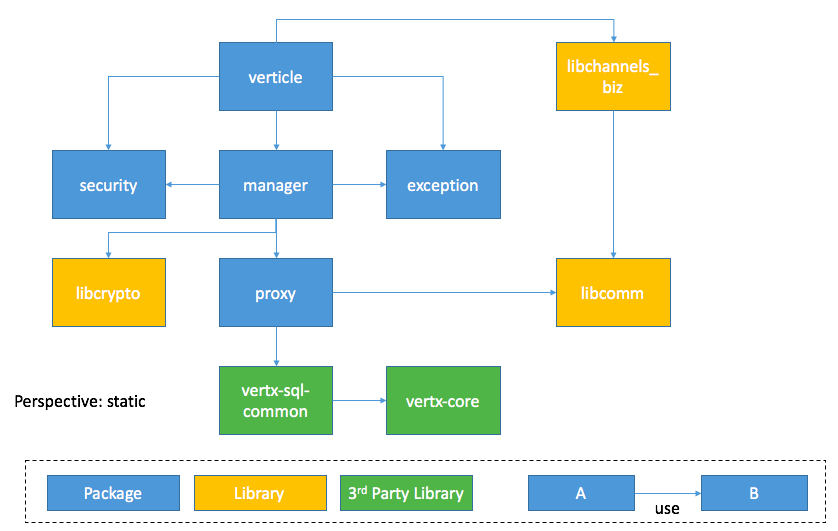
|  |
| --- |
| Sure-Park Server Design |
| TEAM 2 – NOT YET |
|  |
|  |
|  |

### 

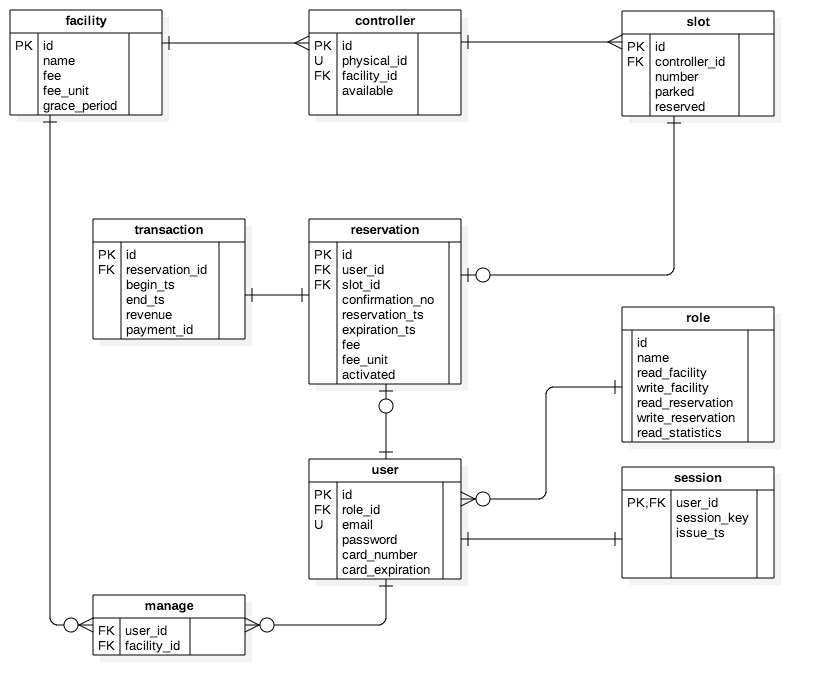
# Static View

## Package and Library Use Relationship View



|  |  |  |
| --- | --- | --- |
| **Entity** | **Type** | **Responsibility** |
| verticle | package | - Listen request and response to it via logical communication channels defined by libchannels\_biz.  - Subscribe logical communication channels defined by libchannels\_biz fot catching interested messages.  - Use manager to process received messages.  - Use exception for responding error situation to requester. |
| manager | package | - Perform business logic like making a reservation and getting facility information. |
| proxy | package | - Perform network communication and database I/O |
| security | package | - Define session which contains global information for authenticated user. Session contains user’s information and authority.  - Define privileges which represent authority of performing particular actions like READ\_RESERVATION and so on. |
| libcrypto | library | - Provide string encryption and decryption methods. |
| libcomm, libchannels\_biz | library | - Provide communication abstraction layer. Details for these library are described at **[TEAM2\_DOC\_04] Protocol\_Design**. |

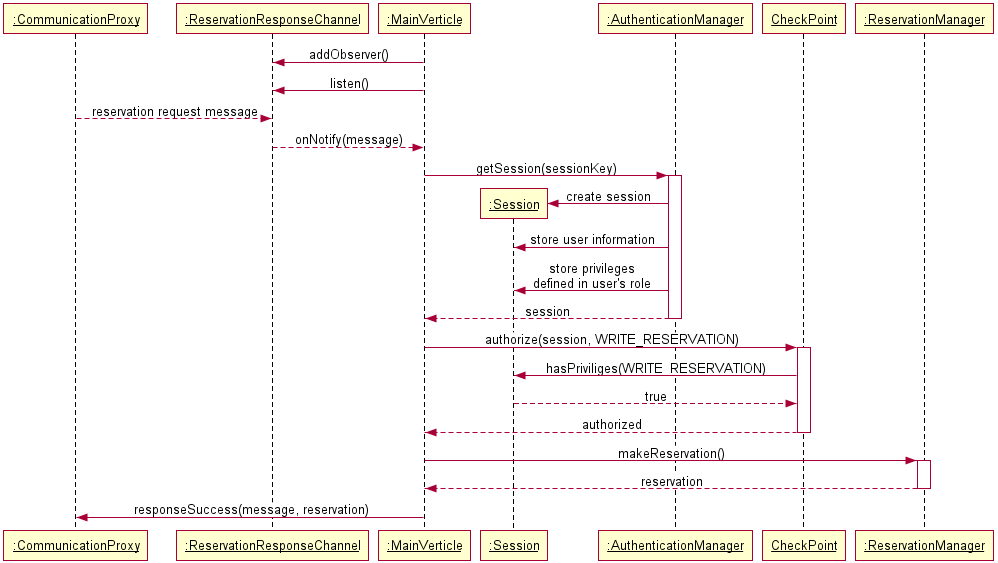
## Database Schema (ERD)



This ERD was primarily designed for supporting multi-owners, multi-facilities, and multi-controllers. An owner can have multi-facilities and an attendant can be assigned to more than one facility. This relationship can be managed by **manage** table. When driver makes a reservation, reserved facility’s configuration values such as fee and grace period is going to be stored into **reservation** table. These configuration values can be referred when charging fee to driver’s card. The **role** table defines role and its privileges which represent authorities to perform particular operations.

# Dynamic View

## Basic Request-Response Sequence Diagram



This sequence diagram shows basic flow related to request-response channel.There are some equipments which promote security such as **CheckPoint**, **Session**, and **AuthenticationManager** which is a sort of Single Access Point. Every request from an application comes with sessionKey which was issued as a result of login request. **AuthenticationManager** create **Session** corresponding to sessionKey and store authenticated user’s global information and user’s privileges into **Session**. **Session** can be used by **CheckPoint** to determine whether user related session has enough privileges to perform particular operations. **MainVertice** permit to access business logic like **ReservationManager** only when **CheckPoint** ensure that this session has required privileges.