

# *ShareDesk (Part 1)* Design Document

CSCI E-97 Assignment 2 9/18/2014

ShareDesk Office Provider Service

Date: September 18, 2014

Author: Philip Lin

Reviewer(s): Daniel Little

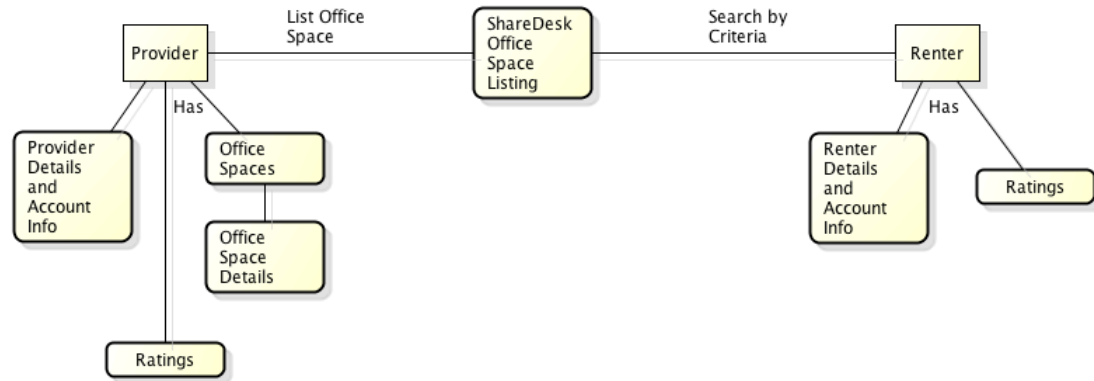
## Introduction

Sites and applications exist that allow people to share their homes, apartments, or other physical spaces that they have with others. This document describes a home sharing application of this sort called ShareDesk. The concept of ShareDesk is to provide a means for individuals to rent out places that they have as office space. The providers can make money from their unused space, and people looking for a place to work have an alternative to traditional office spaces, bookstores, coffee shops or their own homes.

This document describes the first part of the implementation of the ShareDesk application – the definition of an API for modeling office space providers and office spaces within the ShareDesk application.

## Overview

Office space providers need a way to list their available spaces for rent to prospective renters, while renters need a way to search for office spaces that suit their working needs. ShareDesk provides a solution for this by connecting providers and renters through a searchable database of listings. In addition to giving providers a way to list their spaces, and renters a way to search those listings, ShareDesk also takes care of reservations and payments through a secure online system.



ShareDesk structure overview.

Since ShareDesk is the service that facilitates transactions between providers and renters, a small commission is charged for each payment made.

Demand for this service will exist because ShareDesk makes the whole process of listing/finding a space and making a monetary transaction easy, convenient, and fast.

Profits can be sustained as long as transactions continue to occur on the site, and the money earned from those transactions exceeds the costs needed to host and maintain the site.

# Requirements

The ShareDesk Office Provider Service API supports the following requirements:

## 1. Create, read, update, delete instances of Office Space Providers.

People who wish to provide their office spaces register with ShareDesk, and an instance of an Office Space Provider is created for each registered provider. Office Space Provider instances are maintained by the ShareDesk application. Each Office Space Provider has a set of attributes and associations that can be read and updated when necessary. These attributes and associations contain data about the provider. Instances of Provider can be deleted as well, when requested or when necessary.

A system of ratings should also be implemented, such that renters can submit ratings about about office providers and vice versa.

## 2. Create, read, update, and delete instances of Office Space.

Office space instances are created by and associated with a provider, where they will be listed by the ShareDesk application. Office Space instances should be maintained within the ShareDesk application. Each Office Space instance has a set of attributes and associations that can be read and updated when necessary. These attributes and associations contain data about the office space that is being rented out, information that prospective renters use to choose their spaces. Instances of Office Space can be deleted as well, when requested or when necessary.

A system of ratings should also be implemented, such that renters can submit ratings about office spaces.

## 3. Support specification of the Office Space Provider details

An Office Space Provider is registered with the site and includes these properties and associations.

Identifier (GUID)

Name

Contact Info

Picture

Ratings

Account

#### 4. Support specification of the Office Space details

An Office Space has the following descriptive attributes that renters can search for when determining what space they would like to rent:

Features

Location

Capacity Info

Images

Common Access

Rates

Facility Type

Ratings

# Use Cases

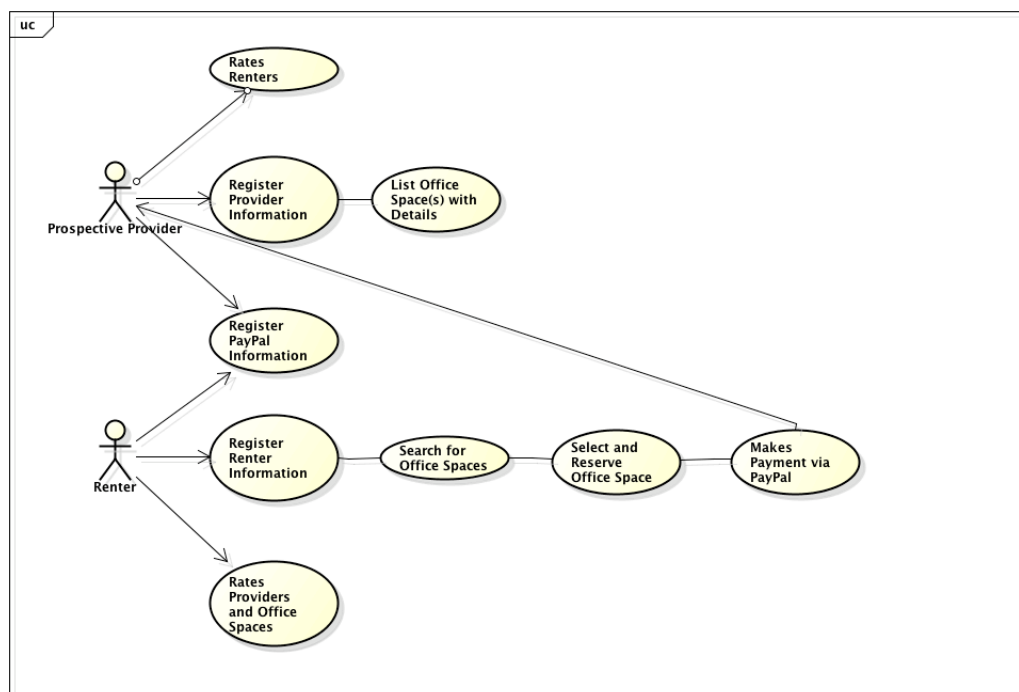
The ShareDesk application supports 2 primary use cases:

## 1. Provider

Office space providers go to the ShareDesk site, register themselves and their available spaces, and provide details about themselves as well as their available spaces. ShareDesk then connects them with prospective renters, and the provider will receive rent payments from the renter minus the 10% commission ShareDesk gets for each transaction. Providers can then rate the renters they work with.

## 2. Renter

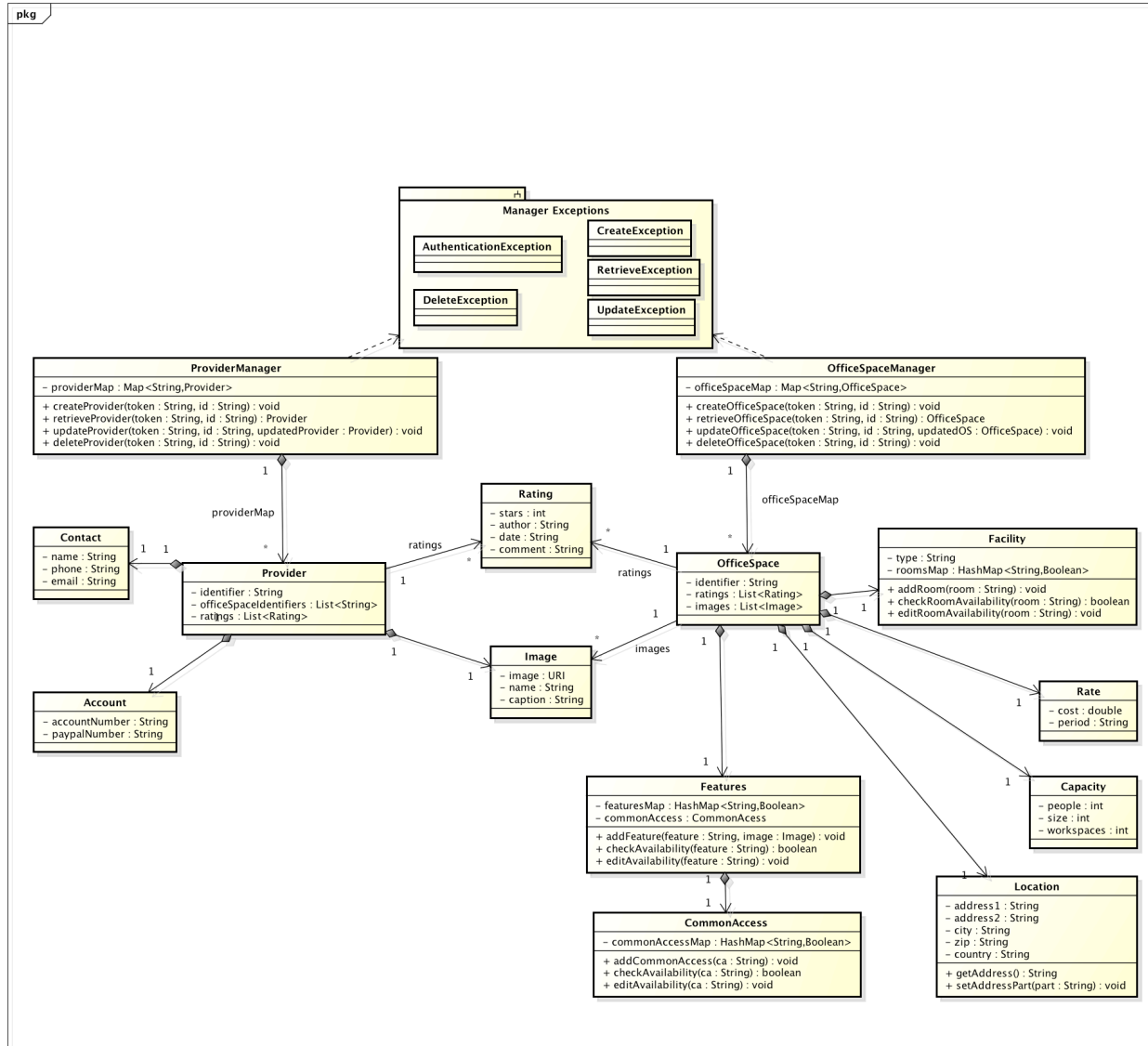
Renters looking for an office space go to the ShareDesk web site, and search for office spaces using a specific set of search criteria. ShareDesk then finds the available spaces that meet the renter's criteria. The renter chooses the space that best fits their needs, and reserves and pays for the space through the ShareDesk payment system. Renters can rate the providers as well as the office space.



Use Case Diagram.

# Implementation

## Class Diagram



## Class Dictionary

This section specifies the class dictionary for the ShareDeskOffice Space Provider API. The classes should be defined within the package “cscie97.asn2.sharedesk.provider”.

### Account

The Account class keeps track of information used to handle transactions between the renter and provider through PayPal on the ShareDesk client.

#### Methods

Method Name	Signature	Description
getPayPal	() : int	Returns the PayPal account number.
setPayPal	(paypalAccount : int) : void	Sets the PayPal account number.

#### Properties

Property Name	Type	Description
accountNumber	String	Private unique non-mutable number that identifies the account. Format is 7-digit number.
paypalNumber	String	Private unique number of the PayPal account associated with renter or provider. Format is 8-digit number.

## Capacity

The Capacity class keeps track of capacity information of the Office Space, including the maximum amount of people, size of the space, and the number of available workspaces.

### Methods

Method Name	Signature	Description
getPeople	() : int	Returns the maximum capacity.
setPeople	(people : int) : void	Sets the maximum capacity.
getSize	() : int	Returns the size.
setSize	(size : int) : void	Sets the size.
getWorkspaces()	() : int	Returns the number of individual workspaces.
setWorkspaces()	(workspaces : int) : void	Sets the number of individual workspaces.

### Properties

Property Name	Type	Description
people	int	Private maximum capacity of the office space.
size	int	Private size of the office space. Units in square feet.
workspaces	int	Private number of individual workspaces available in the office space.

## Common Access

The Common Access class keeps track of areas or items that are available for common use in the facility where an Office Space is located. A common access could be a kitchen, coffee, game room, etc.

### Methods



Method Name	Signature	Description
addCommonAccess	(ca : String) : void	Adds a common access area or item to the map of common access areas and items.
checkCommonAccessAvailability	(ca : String) : boolean	Returns whether a particular common access area or item is available using the map.
editCommonAccessAvailability	(ca : String, availability : Boolean) : void	Changes the availability of a particular common access item in the map.
getCommonAccessMap	() : Map<String, Boolean>	Returns the map of common access areas and items.

### ***Properties***

Property Name	Type	Description
commonAccessMap	Map<String, Boolean>	Private Map maintaining a list of possible common access areas and items by String keys, with a Boolean indicator of the availability of the common access for the office space.

### **Contact**

The Contact class keeps track of contact information for a renter or provider, including phone number and email address.

### ***Methods***

Method Name	Signature	Description
getName	() : String	Returns the name.
setName	(name : String) : void	Sets the name.
getPhone	() : String	Returns the phone number.
setPhone	(phone : String) : void	Sets the phone number.
getEmail	() : String	Returns the email address.
setEmail	(email : String) : void	Sets the email address.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
name	String	Private name of renter or provider.
phone	String	Private phone number of the renter or provider. Format is “(###) ###-####” for US phones.
email	String	Private email of the renter or provider. Format is “__@__”

### **Facility**

The Facility class is used to describe the office space as a type of facility with one or more rooms. The facility type could be house, garage, warehouse, etc.

### ***Methods***

<b>Method Name</b>	<b>Signature</b>	<b>Description</b>
addRoom	(room : String) : void	Adds a room to the map of possible rooms in the facility.
checkRoomAvailability	(room : String) : boolean	Returns whether a particular room is available in the facility using the map.
editRoomAvailability	(room : String, availability : Boolean) : void	Changes the availability of a particular room in the map.
getRoomsMap	() : Map<String, Boolean>	Returns the map of possible rooms in the facility.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
type	String	Private indicator of the type of facility the office space is.

roomsMap	HashMap<String, Boolean>	Private Map maintaining the possible rooms in a facility with String keys and Boolean indicator of the availability of the room in the facility.
----------	--------------------------	--

## Features

The Features class keeps track of features that are available in an Office Space for rent. Features in an office space could include WIFI, parking, or pets being allowed. The set of Common Access areas are also a part of the features of an office space.

## Methods

Method Name	Signature	Description
addFeature	(feature : String) : void	Adds a feature to the map of features in the office space.
checkFeatureAvailability	(feature : String) : boolean	Returns whether a particular feature is available in the office space using the map.
editFeatureAvailability	(feature : String, availability : Boolean) : void	Changes the availability of a particular feature item in the map.
getFeaturesMap	() : Map<String, Boolean>	Returns the map of features.

## Properties

Property Name	Type	Description
featuresMap	Map<String, Boolean>	Private Map maintaining a list of possible features in an office space with String keys and a Boolean indicator of the availability of the feature in the Office Space.
commonAccess	CommonAccess	Private CommonAccess object that is an attribute of the Features class.

## Image

The Image class describes an image that is associated with a provider or office space, and includes a name, caption, and an URI link to the image file.

### Methods

Method Name	Signature	Description
getImage	() : URI	Returns the URI image.
setImage	(image : URI) : void	Sets the URI image.
getName	() : String	Returns the image name.
setName	(name : String) : void	Sets the image name.
getCaption	() : String	Returns the image caption
setCaption	(caption : String) : void	Sets the image caption.

### Properties

Property Name	Type	Description
image	URI	Private image that pictures a provider, renter, or (a part of) an office space. A URI address.
name	String	Private name of an image.
caption	String	Private caption describing an image.

## Location

The Location class describes the address various identifying attributes such as street, city, zip and country of an available Office Space.

### Methods

Method Name	Signature	Description
-------------	-----------	-------------

getAddress	() : String	Returns the full address, formatted.
editAddress	(part : String) : void	Edits a specified portion of the address, based on the input: address1, address2, city, state, zip, or country.

## ***Properties***

Property Name	Type	Description
address1	String	Private address line 1.
address2	String	Private address line 2.
city	String	Private city name.
state	String	Private state name.
zip	String	Private zip code. Format is #####.
country	String	Private country name.

## **OfficeSpaceManager**

The OfficeSpaceManager class manages a map of all the OfficeSpace objects, as well as the create, retrieve, update, and delete operations regarding instances of OfficeSpace.

## ***Methods***

Method Name	Signature	Description
createOfficeSpace	(token : String, id : String) : void	Creates a new Office Space object, as well as new objects for attributes of OfficeSpace.
retrieveOfficeSpace	(token : String, id : String) : OfficeSpace	Returns the OfficeSpace object identified by the input identifier from the map of office spaces.
updateOfficeSpace	(token : String, id : String, updatedOS : OfficeSpace) : void	Updates the OfficeSpace object identified by the input identifier from the map of office spaces with a different OfficeSpace object (DTO).
deleteOfficeSpace	(token : String, id : String) : void	Deletes the OfficeSpace object identified

		by the input identifier from the map of office spaces.
getOfficeSpaceMap	() : Map<String, OfficeSpace>	Returns the Map of OfficeSpace objects.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
officeSpaceMap	Map<String, OfficeSpace>	Private Map maintaining a list of all registered instances of OfficeSpace from providers that are available for rent or currently being rented out.

## **OfficeSpace**

The OfficeSpace class represents an office space with certain attributes that can be rented from a provider to a renter in ShareDesk.

### ***Methods***

<b>Method Name</b>	<b>Signature</b>	<b>Description</b>
getIdentifier	() : String	Returns the identifier.
getRatings	() : List<Rating>	Returns the list of ratings.
getImages	() : List<Image>	Returns the list of images.
getRate	() : Rate	Returns the Rate object.
getLocation	() : Location	Returns the Location object.
getFeatures	() : Features	Returns the Features object.
getCapacity	() : Capacity	Returns the Capacity object.
getFacility	() : Facility	Returns the Facility object.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
identifier	String	Private unique non-mutable identifier that is

		used to identify individual office spaces. Format is 8 digit number.
ratings	List<Rating>	Private List maintaining a list of all ratings received for an office space from renters who have rented it.
images	List<Image>	Private List maintaining a list of all images associated with an office space.

## ***Associations***

<b>Association Name</b>	<b>Type</b>	<b>Description</b>
rate	Rate	Private object used to define the rate of rental of the office space using by cost per unit time.
location	Location	Private object used to define the location of the office space, using a standard US address.
features	Features	Private object used to define the features that are available or not available in the office space. New features can be added as necessary.
capacity	Capacity	Private object used to define the capacity of the office space, in number of people, size, and number of workspaces.
facility	Facility	Private object used to define the facility type, which includes the rooms in the space and their availability for use.

## **ProviderManager**

The ProviderManager class manages a map of all the Provider objects, as well as the create, retrieve, update, and delete operations regarding instances of Provider.

### ***Methods***

<b>Method Name</b>	<b>Signature</b>	<b>Description</b>
createProvider	(token : String, id : String) : void	Creates a new Provider object, as well as

		new objects for attributes of Provider.
retrieveProvider	(token : String, id : String) : Provider	Returns the Provider object identified by the input identifier from the map of providers
updateProvider	(token : String, id : String, updatedProvider : Provider) : void	Updates the Provider object identified by the input identifier from the map of providers with a different provider object (DTO).
deleteProvider	(token : String, id : String) : void	Deletes the Provider object identified by the input identifier from the map of providers.
getProviderMap	() : Map<String, Provider>	Returns the Map of Provider objects.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
providerMap	Map<String, Provider>	Private Map maintaining a list of all registered instances of Provider representing office space providers.

### ***Associations***

<b>Association Name</b>	<b>Type</b>	<b>Description</b>
identifier	String	Private unique non-mutable identifier that is used to identify individual Providers. Format is 8 digit number.
providerMap	Map<String, Provider>	Private Map maintaining a list of all registered Providers.
ratings	List<Rating>	Private List maintaining a list of all ratings received for a provider from a renter.

## **Provider**

The Provider class represents a Provider with certain attributes and a



list of OfficeSpace identifiers that keep track of OfficeSpace objects associated with the provider in ShareDesk.

### ***Methods***

<b>Method Name</b>	<b>Signature</b>	<b>Description</b>
getIdentifier	() : String	Returns the identifier.
getOfficeSpaceIdentifiers	() : List<String>	Returns the list of office space identifiers.
getRatings	() : List<Rating>	Returns the list of ratings.
getContact	() : List<Image>	Returns the Contact object.
getAccount	() : Rate	Returns the Account object.
getImage	() : Location	Returns the Image object.

### ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
identifier	String	Private unique non-mutable identifier that is used to identify individual office spaces. Format is 8 digit number.
officeSpaceIdentifiers	List<String>	Private List maintaining a list of all the identifiers of OfficeSpace objects associated with this Provider.
ratings	List<Rating>	Private List maintaining a list of all ratings received for an office space from renters who have rented it.

### ***Associations***

<b>Association Name</b>	<b>Type</b>	<b>Description</b>
contact	Contact	Private object used to define contact information, which is the name, phone, and email of the provider.
account	Account	Private object used to define the account information of the provider, including PayPal account number.
image	Image	Private object used to define the profile image of the provider, with the image in URI form.

## Rate

The Rate class keeps track of the rate to rent an office space determined by the office space provider. The class also keeps track of the dates the space is available or being rented.

### Methods

Method Name	Signature	Description
getRate	() : double	Returns the daily rental rate for an office space.
setRate	(rate : double) : void	Sets the daily rental rate for an office space.
getDaysRented	() : int	Returns the number of days that an office space is being rented out for.
setDaysRented	(daysRented : int) : void	Sets the number of days to rent out an office space.

### Properties

Property Name	Type	Description
rate	double	Private daily rental rate for an office space. Format is dollars per day.
daysRented	int	Private int of the number of days space is rented.

## Rating

The Rating class describes feedback items that providers give to renters or that give to providers or office spaces. Ratings include stars as well as comments.

## ***Methods***

<b>Method Name</b>	<b>Signature</b>	<b>Description</b>
getStars	() : int	Returns the number of stars in the rating.
setStars	(stars : int) : void	Sets the number of stars in the rating.
getAuthor	() : String	Returns the author of the rating
setAuthor	(author : String) : void	Sets the author of the rating.
getComment	() : String	Returns the comment of the rating.
setComment	(comment : String) : void	Sets the comment of the rating.
getDate	() : String	Returns the date the rating was made.
setDate	(date : String) : void	Sets the date the rating was made.

## ***Properties***

<b>Property Name</b>	<b>Type</b>	<b>Description</b>
stars	int	Private number of stars in the rating. Rating can be 1-5 stars.
author	String	Private name of the author of the rating. Should be the name given for a Renter or Provider.
comment	String	Private comment justifying the rating given. Format is a sentence or paragraph of text.
date	String	Private date the comment was authored.

## Implementation Details

The create, retrieve, update, and delete operations for individual instances of Provider and OfficeSpace are defined in the ProviderManager and OfficeSpaceManager classes, respectively.

When a Provider object is initially created with the create method, the following objects associated with Provider are also created in the constructor:

- Two Lists (for Ratings and OfficeSpace identifiers)

- Account object

- Contact object

- Image object

Likewise, when an OfficeSpace object is initially created with the create method, the following objects associated with OfficeSpace are also created in the constructor:

- Two Lists (for Ratings and Images)

- Capacity object

- Features and CommonAccess objects

- Location object

- Rate object

All of these objects created in the constructor initially contain empty fields, but are updated with actual information by using class setters and the retrieve and update methods of the manager classes.

In this way, correct information can be added provider and office space instances after creation.

## Testing

A test driver class called `TestDriver`, which includes a main method, is used to test the functionality of the API. The test driver should be able to create sample Office Space Providers and sample Office Spaces, as well as retrieve, update, and delete them as well.

Exceptions should also be handled for authentication errors, as well as errors when creating, retrieving, updating, or deleting `OfficeSpace` or `Provider` objects. These errors happen when objects don't exist when called for, or when attributes of objects are improperly edited or accessed.

The `TestDriver` class should be defined within the package `"cscie97.asn2.test"`.

## Risks

Because of the in-memory implementation, the number of Providers and Office Spaces is limited by the memory allocated to the JVM. Providers and Office Spaces also must be implemented and updated properly as to avoid making references to objects or fields that don't exist, something that can be checked before any operations are actually done. Authentication must be done in order to ensure that any changes to the system are authorized and account information is correct and usable.