SquareDesk
Authentication Service Requirements

Author: Eric Gieseke

Date: 10/30/2014

Introduction

This document provides the requirements for the SquareDesk Authentication Service.

Overview

SquareDesk is a new service that allows people to rent out their home as office space. SquareDesk allows people to make additional income by renting out portions of their home as office space. People who need a place to work and want to get out of their home, or find a quieter place to work than Starbucks, can find a place through SquareDesk.

SquareDesk makes it easy for people to register and list their homes as office space. A candidate office space provider simply navigates to the SquareDesk site, registers, provides details about the space they have for rent, and SquareDesk does the rest.

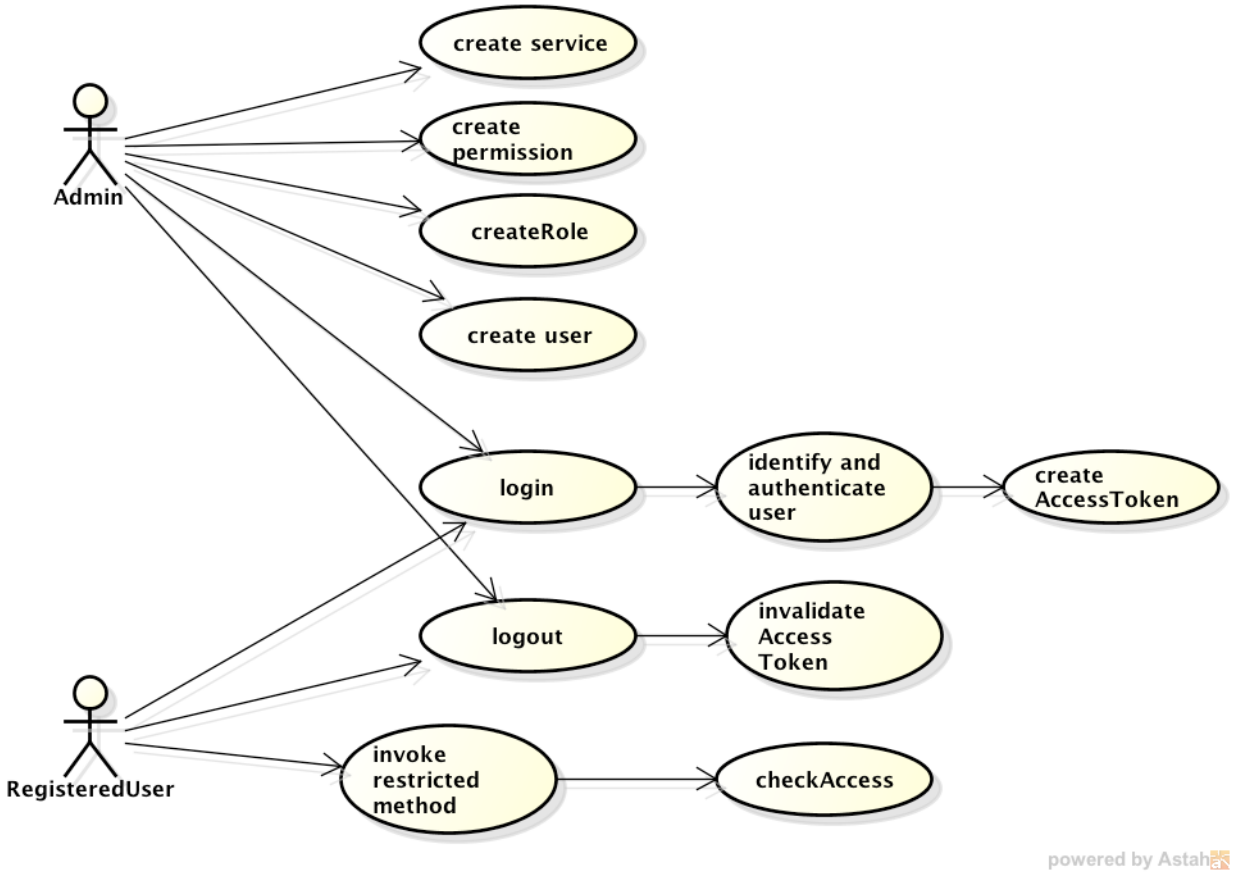
People looking for office space, go to the SquareDesk web site and search for office space near their location. Search criteria can help narrow the search to spaces that work for them. For example: includes WIFI, allows pets, and includes free coffee. Select the days and time you need the space, location, and price range. SquareDesk will locate all the SquareDesk available spaces that meet the criteria.

With the list of available places to work, the renter chooses the one that they like best and then reserves and pays for the space. Reservations and payment are seamlessly managed by the SquareDesk system. SquareDesk takes a small commission (10%) and the rest is credited to the account of the office space provider.

Office space providers and renters can rate each other to help ensure a good experience for all.

Authentication Service

The Authentication Service is responsible for controlling access to the SquareDesk restricted service interfaces. The Authentication Service provides a central point for managing Users, Services, Permissions, Roles, and Access Tokens.



The Authentication Service supports 2 primary actors or roles: The Administrator and Registered Users. The Administrator is responsible for managing the Services, Permissions, Roles, and Users maintained by the Authentication Service. The Registered User is the consumer of the the Authentication Service, using it to login, logout, and indirectly to access restricted service methods.

Authentication Service API

The Authentication Service API is needed to support the following functions:

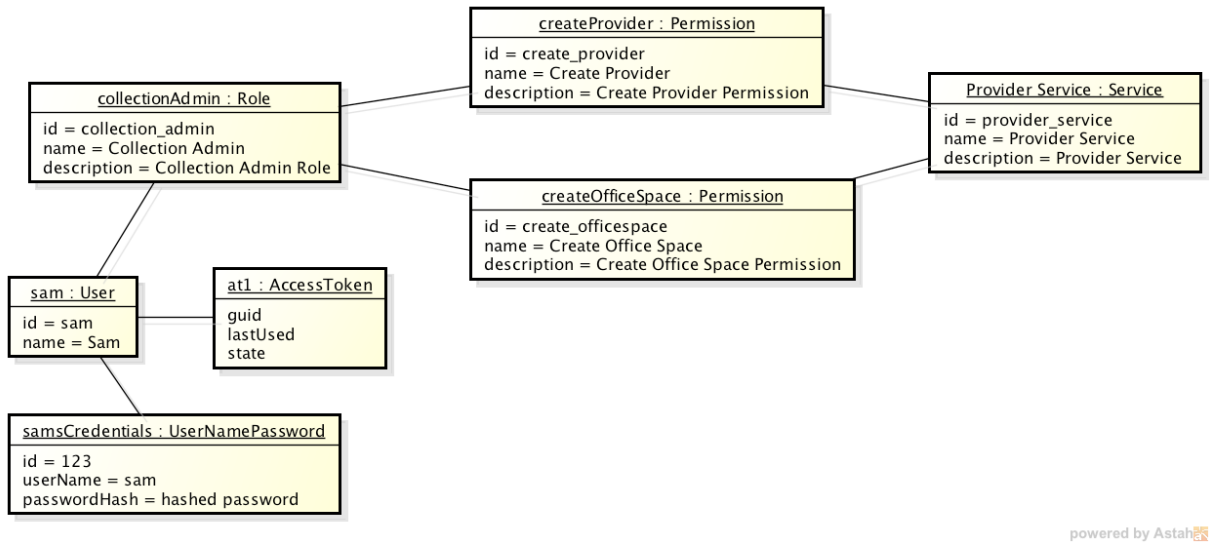
- Creating Services:
    - Service aggregates a set of permissions specific to the Service

    - A Service has a unique ID, a name, and a description.
    - A Service represents any of the SquareDesk services, including the Provider and Renter Service, and the Authentication Service.
    - Restricted access
- Creating Permissions:
    - Permissions represent an entitlement required to access restricted functionality of the SquareDesk application. Restricted service methods require that the user accessing the method have the permission associated with the method.
    - Permissions are specific to a Service.
    - Permissions have a unique id, name, and description.
    - A Service may have one or more permissions.
    - A User may be associated with zero or more permissions.
    - Restricted access
- Creating Roles:
    - Roles are composites of Permissions.
    - Roles provide a way to group Permissions and other Roles.
    - Like Permissions, Roles have a unique id, name, and description.
    - Users may be associated with Roles, where the user has all permissions included in the Role or sub Roles.
    - Roles help simplify the administration of Users by providing reusable and logical groupings of Permissions and Roles.
    - Restricted Access
- Creating Users:
    - Users represent registers users of the SquareDesk application.
    - Users have an id, a name and a set of Credentials. Credentials include a username and a password. To help protect the password, the password should be hashed.
    - Users are associated with 0 or more Roles or Permissions.
    - Restricted Access
- Login:
    - The Login process provides users AccessTokens that can then be used to access restricted Service Methods.
    - Login accepts a User's credentials (username, password).
    - A check is made to make sure that the username exists, and then that the hash of the password matches the known hashed password.
    - If authentication fails, an AuthenticationException should be thrown.
    - If authentication succeeds, an AccessToken is created and returned to the caller.
    - The accessToken binds the User to a set of permissions that can be used to grant or deny access to restricted methods.
    - AccessTokens can timeout with inactivity.
    - AccessTokens have a unique id, an expiration time, and a state (active or expired).
    - Access tokens are associated with a User and a set of Permissions.

- Logout:
    - Logout marks the given Access Token as invalid.
    - Subsequent attempts to use the AccessToken should result in a InvalidAccessTokenException.
- Invoking Restricted Methods:
    - All restricted methods defined within the SquareDesk application should accept a AccessToken
    - Each Restricted method should validate that the AccessToken is non null and non empty.
    - The Restricted method should pass the accessToken to the Authentication Service with the permission required for the method.
    - The AuthenticationService should check to make sure that the AccessToken is active, and within the expiration period, and then check that the user associated with the AccessToken has the permission required by the method.
    - The AuthenticationServce should through a AccessDeniedException or InvalidAccessTokenException if any of the checks fail.

Exceptions should include useful information to help users understand the nature of the Exception.

The following instance diagram shows how the User, Role, Permission, Service, AccessToken, and User Credentials are related.



Caption: Sample instance diagram for Authentication Service API.

Restricted methods include restricted methods on the Authentication, Provider and Renter Services.

Sample Authentication Data:

 # define service

 # define_service, <service_id>, <service_name>, <service_description>

define_service, renter_api_service, Renter API Service, Renter Management and Access

define_service, provider_api_service, Provider API Service, Provider Management and Access

define_service, authentication_service, Authentication Service, Manage Authentication Configuration and Control Access to Restricted Service Interfaces

 # define permissions

 # define_permission, <service_id>, <permission_id>, <permission_name>, <permission_description>

define_permission, provider_api_service, create_provider, Create Provider

define_permission, provider_api_service, create_officespace, Create Office Space Permission, Permission to create a new office space

 # define roles

 # define_role, <role_id>, <role_name>, <role_description>

define_role, provider_role, Provide Role, All permissions required by providers

 # add entitlement (permission or role) to role

 # add_entitlement_to_role, <role_id>, <entitlement_id>

add_entitlement_to_role, provider_role, create_provider

add_entitlement_to_role, provider_role, create_officespace

 # create_user

 # create_user <user_id>, <user_name>, <password>

create_user, sam, Sam, secret

 # add role to user

 # add_role_to_user <user_id>, <role>

add_role_to_user sam, provider_role