House Mate Entitlement Service Design Document

Date: 11/03/2015
Author: Yingtian Wang

Introduction

This document provides the design for the House Mate Entitlement Service. House Mate Entitlement Service is an important component of House Mate System. It adds security to the system to prevent hacking or misuse of the system.

Overview

This document recaps the requirements first, then the use case diagram shows how to use this program. The implementation part of the document describe how the implementation meets the requirement, it presents the class diagram to show the Entitlement service system domain classes. Following with class dictionary, it describes the property, association, operation of the class in detail. In addition, the sequence diagram shows the flow of events for checking access.

Requirements

This section defines the requirements for the House Mate Entitlement Service.

The House Mate Entitlement Service entitles access control to the House Mate Model Service and House Mate Controller Service.

1. There are different types of instances of the Entitlement Service domain classes, such as Resources, Permissions, Roles, Users. This design use visitor pattern to check access of theses objects.

2. This system support processing Entitlement service API, and it use Abstract Factory Pattern to create instance of the Entitlements service domain classes.

3. Like House Mate Model Service and House Mate Controller Service. This design use Singleton pattern to make sure the only instance of House Mate Entitlement Service.

4. Roles can have permissions and other roles, the design use composite pattern to manage the relationship.

Use Cases

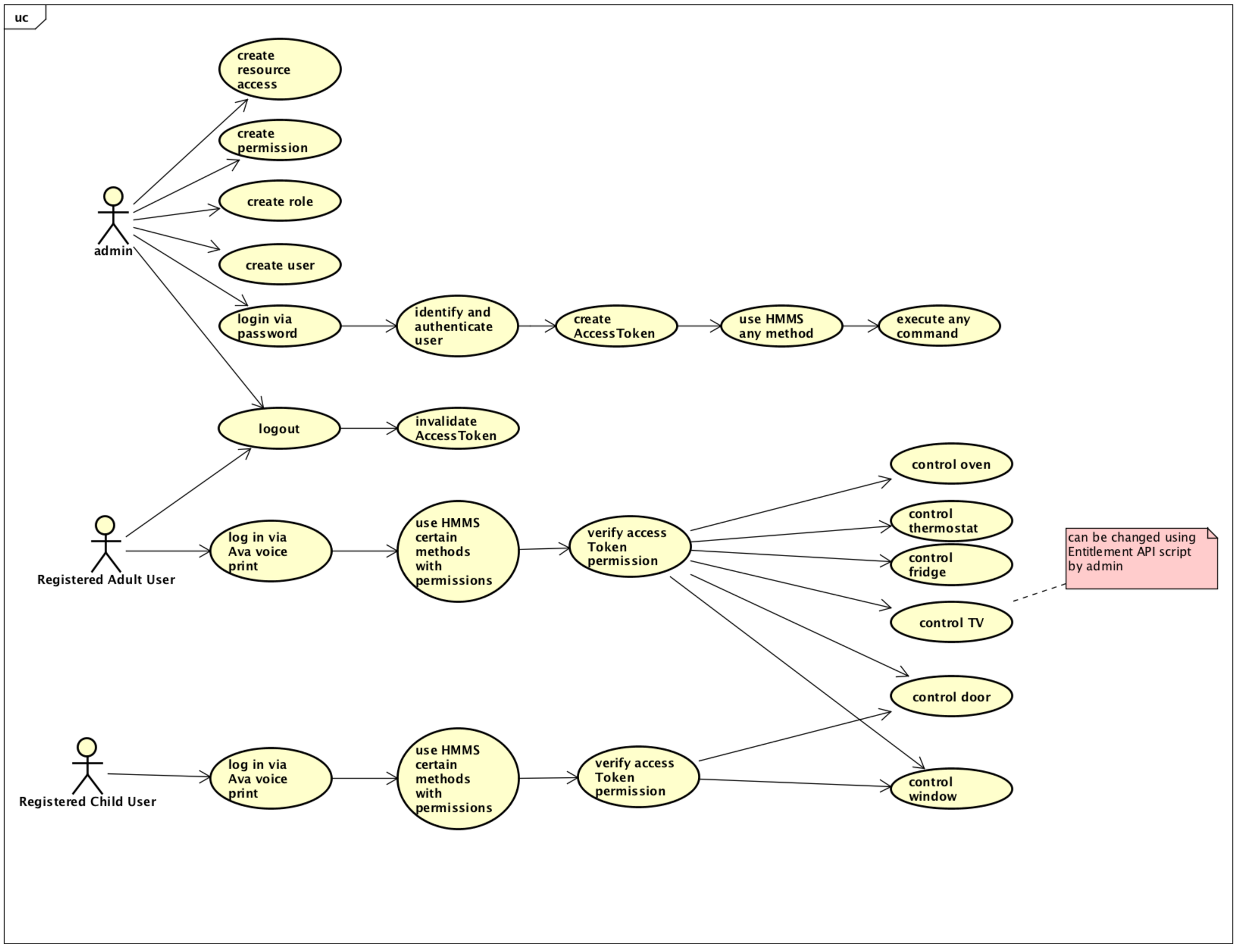
The following use case diagram shows the use case supported by the House Mate Entitlement Service System.

The Administrator uses Entitlement Service API script to create roles, permissions, resource roles, users. Once the Administrator logged in, it has the admin_role, after being verified the access token to confirm, the administrator can use any method in the House Mate Model Service.

The Registered adult user logged in to the service through Ava device voice print. Once the access token be verified, the registered user can have access to the device defined in the script.

It has less permissions than admin user, but can be changed by admin Entitlement API script.

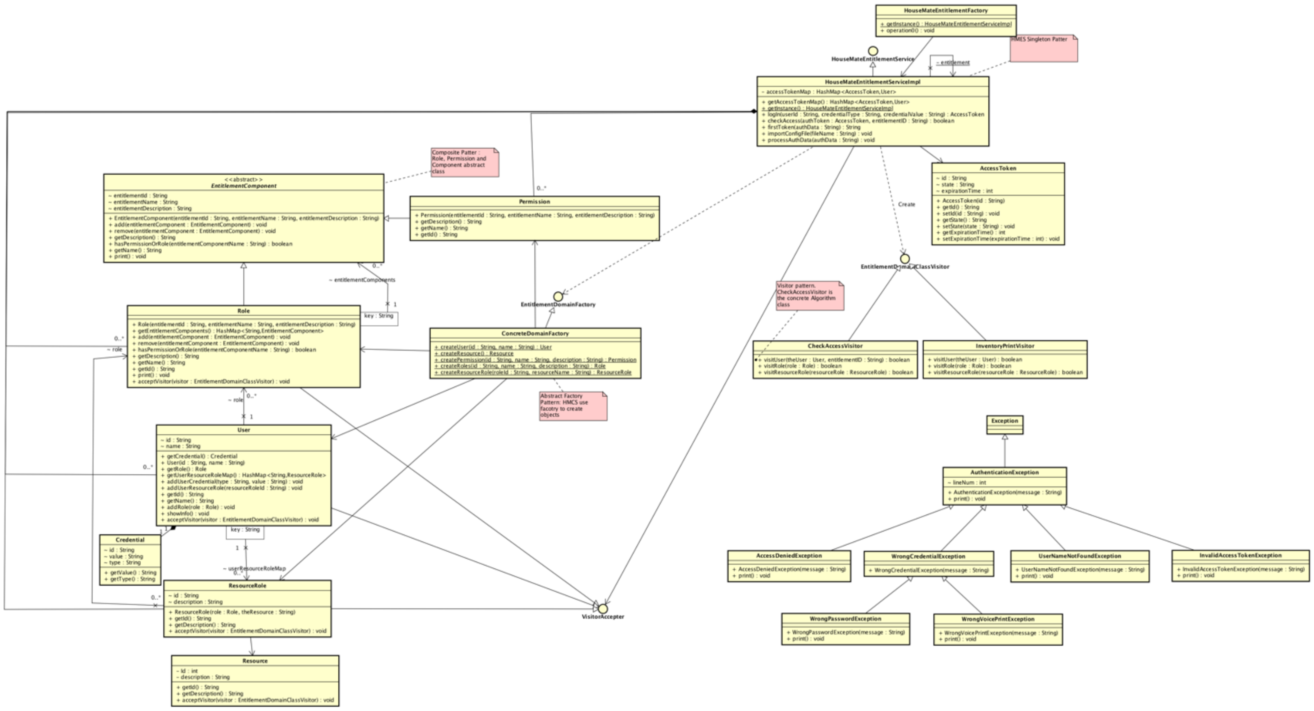
The Registered child user logged in to the service through Ava device voice print. Once the access token be verified, the registered user can have access to the device defined in the script. It has less permissions than adult user, but can be changed by admin Entitlement API script script.


uc
admin
Registered Adult User
Registered Child User
create resource access
create permission
create role
create user
login via password
identify and authenticate user
create AccessToken
use HMMS any method
execute any command
logout
invalidate AccessToken
log in via Ava voice print
use HMMS certain methods with permissions
verify access Token permission
control oven
control thermostat
control fridge
control TV
control door
control window
log in via Ava voice print
use HMMS certain methods with permissions
verify access Token permission
can be changed using Entitlement API script by admin

Implémentation

Class Diagram

Please find “HMES_Class_Diagram” in “designDoc” folder for more clear view of the class diagram.


HouseMateEntitlementFactory
+ getInstance() : HouseMateEntitlementServiceImpl
+ operation0() : void
HouseMateEntitlementService
- entitlement
HouseMateEntitlementServiceImpl
- accessTokenMap : HashMap<AccessToken,User>
+ getAccessTokenMap() : HashMap<AccessToken,User>
+ getInstance() : HouseMateEntitlementServiceImpl
+ logIn(userId : String, credentialType : String, credentialValue : String) : AccessToken
+ checkAccess(authToken : AccessToken, entitlementID : String) : boolean
+ firstToken(authData : String) : String
+ importConfigFile(fileName : String) : void
+ processAuthData(authData : String) : void
AccessToken
~ id : String
~ state : String
~ expirationTime : int
+ AccessToken(id : String)
+ getId() : String
+ setId(id : String) : void
+ getState() : String
+ setState(state : String) : void
+ getExpirationTime() : int
+ setExpirationTime(expirationTime : int) : void
Create
EntitlementDomainClassVisitor
CheckAccessVisitor
+ visitUser(theUser : User, entitlementID : String) : boolean
+ visitRole(role : Role) : boolean
+ visitResourceRole(resourceRole : ResourceRole) : boolean
InventoryPrintVisitor
+ visitUser(theUser : User) : boolean
+ visitRole(role : Role) : boolean
+ visitResourceRole(resourceRole : ResourceRole) : boolean
<<abstract>>
EntitlementComponent
~ entitlementId : String
~ entitlementName : String
~ entitlementDescription : String
+ EntitlementComponent(entitlementId : String, entitlementName : String, entitlementDescription : String)
+ add(entitlementComponent : EntitlementComponent) : void
+ remove(entitlementComponent : EntitlementComponent) : void
+ getDescription() : String
+ hasPermissionOrRole(entitlementComponentName : String) : boolean
+ getName() : String
+ print() : void
0..*
Permission
+ Permission(entitlementId : String, entitlementName : String, entitlementDescription : String)
+ getDescription() : String
+ getName() : String
+ getId() : String
0..*
~ entitlementComponents
1
key : String
EntitlementDomainFactory
Role
+ Role(entitlementId : String, entitlementName : String, entitlementDescription : String)
+ getEntitlementComponents() : HashMap<String,EntitlementComponent>
+ add(entitlementComponent : EntitlementComponent) : void
+ remove(entitlementComponent : EntitlementComponent) : void
+ hasPermissionOrRole(entitlementComponentName : String) : boolean
+ getDescription() : String
+ getName() : String
+ getId() : String
+ print() : void
+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void
0..*
~ role
ConcreteDomainFactory
+ createUser(id : String, name : String) : User
+ createResource() : Resource
+ createPermission(id : String, name : String, description : String) : Permission
+ createRoles(id : String, name : String, description : String) : Role
+ createResourceRole(roleId : String, resourceName : String) : ResourceRole
~ role
0..*
1
User
~ id : String
~ name : String
+ getCredential() : Credential
+ User(id : String, name : String)
+ getRole() : Role
+ getUserResourceRoleMap() : HashMap<String,ResourceRole>
+ addUserCredential(type : String, value : String) : void
+ addUserResourceRole(resourceRoleId : String) : void
+ getId() : String
+ getName() : String
+ addRole(role : Role) : void
+ showInfo() : void
+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void
0..*
1
1
Credential
~ id : String
~ value : String
~ type : String
+ getValue() : String
+ getType() : String
key : String
1
~ userResourceRoleMap
0..*
0..*
ResourceRole
~ id : String
~ description : String
+ ResourceRole(role : Role, theResource : String)
+ getId() : String
+ getDescription() : String
+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void
VisitorAccepter
Resource
~ Id : int
~ description : String
+ getId() : String
+ getDescription() : String
+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void
Exception
AuthenticationException
~ lineNum : int
+ AuthenticationException(message : String)
+ print() : void
AccessDeniedException
+ AccessDeniedException(message : String)
+ print() : void
WrongCredentialException
+ WrongCredentialException(message : String)
UserNameNotFoundException
+ UserNameNotFoundException(message : String)
+ print() : void
InvalidAccessTokenException
+ InvalidAccessTokenException(message : String)
+ print() : void
WrongPasswordException
+ WrongPasswordException(message : String)
+ print() : void
WrongVoicePrintException
+ WrongVoicePrintException(message : String)
+ print() : void

HMES Singleton Patter

Composite Patter : Role, Permission and Component abstract class

Abstract Factory Pattern: HMCS use facotry to create objects

Visitor pattern. CheckAccessVisitor is the concrete Algorithm class

Class Dictionary

This section specifies the class dictionary for the House Mate Entitlement Service.

HouseMateEntitlementService / HouseMateEntitlementImpl

HouseMateEntitlementService is to authenticate the user to check if they have access to a behavior in the House Mate Model Service and House Mate controller service.

Methods

Method Name	Signature	Description
getInstance	HouseMateEntitlementService: void	HouseMateEntitlementService is a singleton, this method return the instance of itself
logIn	+ logIn(userId : String, credentialType : String, credentialValue : String) : AccessToken	The logIn method provides verify the username and password or voice print and return the accesstoken if success
checkAccess	+ checkAccess(authToken : AccessToken, entitlementID : String) : boolean	Check if the access token has the right permission or role
processAuthData	+ processAuthData(authData : String) : void	This method process the authentication data file and use domain factory to generate Entitlement domain object

Associations

Association Name	Type	Description
EntitlementDomainFactory	Abstract Class	EntitlementDomainFactory is an abstract class that provides the method to generate domain classes.

AccessToken	Class	Entitlement Service has a hashmap to store the user and associated accessToken.
EntitlementDomainClassVisitor	Interface	Use EntitlementDomainClassVisitor to check access.
userMap	HashMap	HashMap to store user instatnces, associated with user class
entitlementMap	HashMap	HashMap to store role and permissions instatnces, associated with entitlement abstract class
resourceRoleMap	HashMap	HashMap to store resourceRole instatnces, associated with resourceRole class

House Mate Entitlement Factory

Based on the Architecture Document, this class is to provide a factory for accessing the HouseMateEntitlement singleton instance.

Methods

Method Name	Signature	Description
getInstance	HouseMateEntitlement	Return the HouseMateEntitlement singleton instance

EntitlementDomainFactory/ConcreteDomainFactory

EntitlementDomainFactory is an abstract factory that provides the methods to create domain class instances.

Method Name	Signature	Description
createUser	+ createUser(id : String, name : String) : User	Method to create user instance
createPermission	+ createPermission(id : String, name : String, description : String) : Permission	Method to create Permission instance
createRoles	+ createRoles(id : String, name : String, description : String) : Role	Method to create Role instance
createResourceRole	+ createResourceRole(roleId : String, resourceName : String) : ResourceRole	Method to create resourceRole instance

AccessToken

AccessToken class has an id, state and expiration time. The state indicates if it is active or expired. The access token can timeout due to inactivity. This class use Java UUID utility as an unique identifier for one access token.

EntitlementComponent

EntitlementComponent is an abstract class that has two subclasses: Permission and Role. This is part of composite pattern.

Method Name	Signature	Description
add	+ add(entitlementComponent : EntitlementComponent) : void	If not overridden, throw unsupportedOperationException.

remove	+ remove(entitlementComponent : EntitlementComponent) : void	If not overridden, throw unsupportedOperationException.
hasPermissionOrRole	+ hasPermissionOrRole(entitlementComponentName : String) : boolean	If not overridden, throw unsupportedOperationException

Role/Permission

Role is a subclass of EntitlementComponent, it has a hash map that can contain any role or permission inside itself. Permission is a subclass of EntitlementComponent, it defines a permission to access an appliance.

Methods

Method Name	Signature	Description
add	+ add(entitlementComponent : EntitlementComponent) : void	Add one entitlement component to this role
remove	+ remove(entitlementComponent : EntitlementComponent) : void	remove one entitlement component to this role
hasPermissionOrRole	+ hasPermissionOrRole(entitlementComponentName : String) : boolean	Check if a role has one particular permission
acceptVisitor	+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void	Accept the algorithm visitor class to implement the visitor pattern.

User

User is one of the Entitlement domain classes. It has the info of the credential and the role and resource role that is associate with the user.

Methods

Method Name	Signature	Description
addUserCredential	+ addUserCredential(type : String, value : String) : void	Add a type of credential(password or voiceprint) to the user.
addUserResourceRole	+ addUserResourceRole(resourceRoleId : String) : void	Add a resource role to user
addRole	+ addRole(role : Role) : void	Add a role to user
acceptVisitor	+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void	Accept the algorithm visitor class to implement the visitor pattern.

Resource

Resource refers to one of the concrete instance such as house and room.

Methods

Method Name	Signature	Description
acceptVisitor	+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void	Accept the algorithm visitor class to implement the visitor pattern.

Resource Role

Resource Role is an associated class that associate Resource and Role.

Methods

Method Name	Signature	Description
acceptVisitor	+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void	Accept the algorithm visitor class to implement the visitor pattern.

Credential

Credential is a class that contains username and password or voiceprint.

Methods

Method Name	Signature	Description
getType	+ getType() : String	Get the type of the credential, voice print or password
getValue	+ getValue() : String	Get the value of the credential

VisitorAccepter

VisitorAccepter is an interface that provides an acceptVisitor method for Entitlement domain classes.

Methods

Method Name	Signature	Description
acceptVisitor	+ acceptVisitor(visitor : EntitlementDomainClassVisitor) : void	Accept the algorithm visitor class to implement the visitor pattern.

EntitlementDomainClassVisitor

EntitlementDomainClassVisitor is an interface provides the methods to visit user, role, permission,and resource role.

Methods

Method Name	Signature	Description
visitUser	+ visitUser(theUser : User, entitlementID : String) : boolean	Visit an user instance
visitRole	+ visitRole(theUser : User, entitlementID : String) : boolean	Visit a role instance
visitPermission	+ visitPermission(theUser : User, entitlementID : String) : boolean	Visit a permission instance
visitResourceRole	+ visitResourceRole(theUser : User, entitlementID : String) : boolean	Visit a resource role instance

CheckAccessVisitor

CheckAccessVisitor is an algorithm class that implements. EntitlementDomainClassVisitor to check access of a user, role, permission and resource role.

Methods

Method Name	**Signature**	**Description**
visitUser	+ visitUser(theUser : User, entitlementID : String) : boolean	Check access of an user instance
visitRole	+ visitRole(theUser : User, entitlementID : String) : boolean	Check access of a role instance
visitPermission	+ visitPermission(theUser : User, entitlementID : String) : boolean	Check access of a permission instance
visitResourceRole	+ visitResourceRole(theUser : User, entitlementID : String) : boolean	Check access of a resource role instance

InventoryPrintVisitor

InventroyPrintVisitor is an algorithm class that implements EntitlementDomainClassVisitor to traverse a user, role, permission and resource role.

Methods

Method Name	**Signature**	**Description**
visitUser	+ visitUser(theUser : User,) : boolean	traverse an user instance
visitRole	+ visitRole(theUser : User,) : boolean	traverse a role instance
visitPermission	+ visitPermission(theUser : User) : boolean	traverse a permission instance

visitResourceRole	+ visitResourceRole(theUser : User) : boolean	traverse a resource role instance

AuthenticationException

This is the parent exception class for all HouseMateEntitlement exceptions. For any reason the authentication fails, an AuthenticationException should be thrown.

AccessDeniedException

If the user has no right permission or role to use the HouseMateModel method or try to use ava to control an appliance that is has no access to, an AccessDeneidException is thrown.

InvalidAccessTokenException

If the user has no AccessToken known by HouseMateEntitlement Service, or the AccessToken has expired, an InvalidAccessTokeException is thrown.

WrongCredentialException

This is the parent class for WrongPasswordException and WrongVoicePrint Exception.

WrongPasswordException

This Exception should be thrown when password check is wrong.

WrongVoicePrintException

This Exception should be thrown when the voice print is wrong.

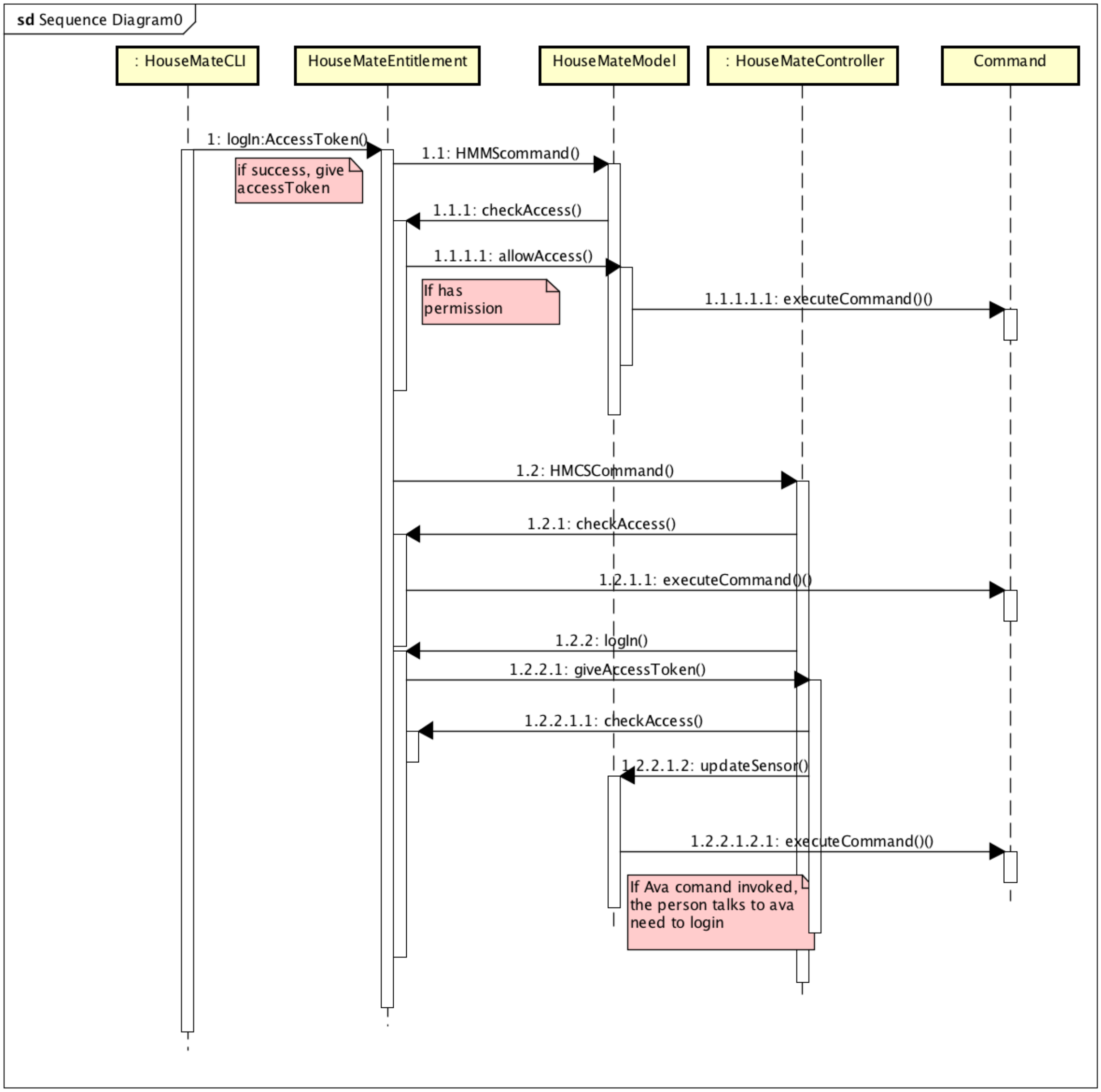
UserNameNotFoundException

When a username is not pre-defined in the authentication script, an UserNameNotFoundException is thrown.

Sequence Diagram

HouMateCLI as a user login in as an administrator to gain an access token, the HouseMateEntitlement check the username and password to see if it matches the hashed password. If success, it assign the user an access token associated roles and permissions.

When the user trys to user HouseMateModelService or HouseMateControllerService, the HouseMateEntitlement check the access token to see if this user has this permission.



Testing

There are three types of testing cases.

1. Test different types of roles. For each type of role, it should have the right access as defined in the auth data script. For example, if the script didn't give child adult light control access, when a child try to control light, an AccessDeniedException is thrown.

2. Test different types of failed login. UserNameNotFoundException is thrown when an username is not defined in the script. WrongPasswordExcetpion is thrown when password is wrong. Similarly, the WrongVoicePrintException is thrown when voice print doesn't match. In addition, if someone try to use HouseModelService without login first. InvalidAccessTokenException is thrown.

3. Test the controller. When someone talks to ava and the voice print match, he should automatic log in. And based on his role and permission to execute the commands.

Implementation Details

1. The AccessToken global unique identifier is implemented by JAVA built in UUID class.

2. The result report folder contains the result report to verify the functionality.

Risks

1. A user is automatically logged out when another user is try to login or when the program finish execution. It does not support multiple user login to the system at the same time.

2. The HouseMateEntitlement provides a method to show the hashed password, this is only for the convenience of showing the functionality of the assignment. This service is not provided for other purpose.