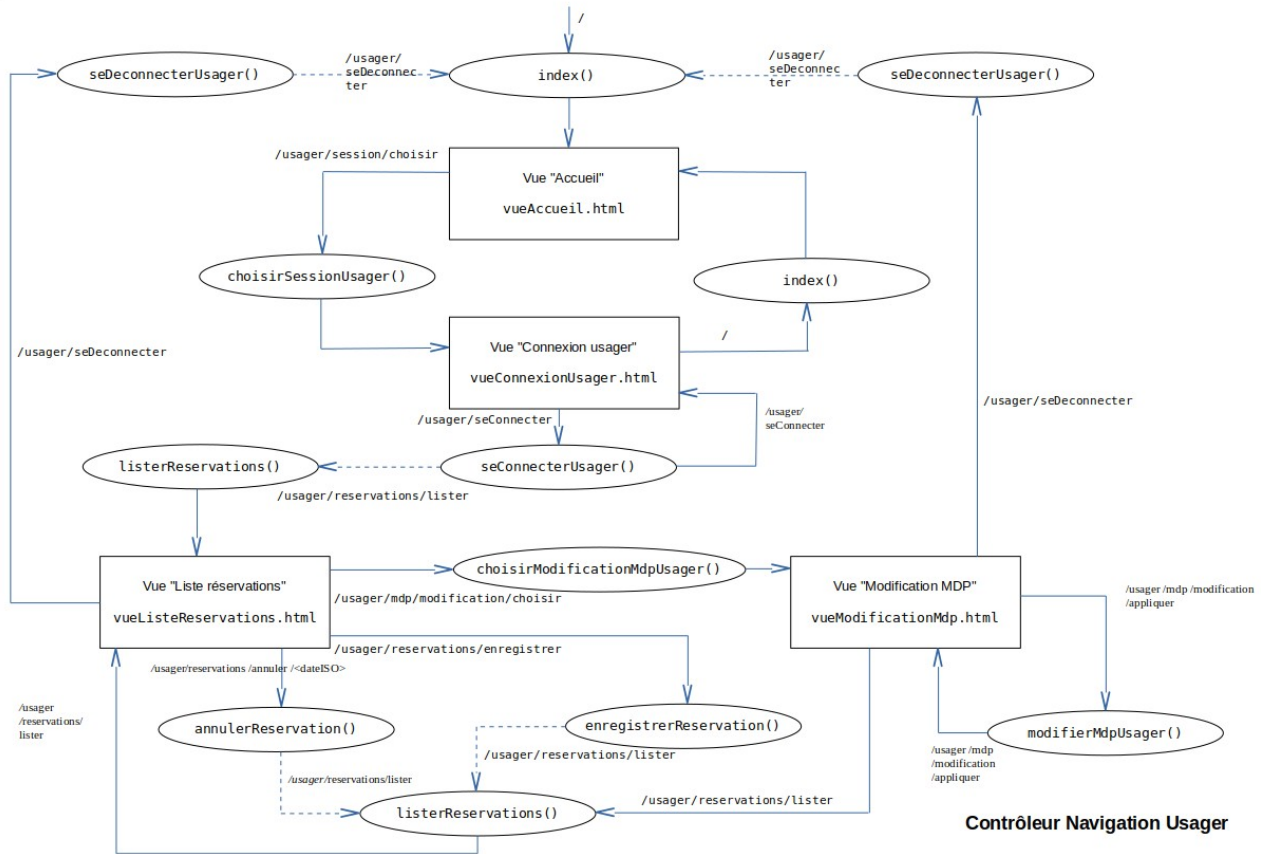


Compte rendu AP2-Mission4-Prototype – Analyse de l'existant :

1,3,4)Voici le contrôleur navigation usager complété :



2)

Car dans la fonctions `seConnecterUsager()`, la sortie est différente si une condition est vraie ou fausse :

```
@app.route( '/usager/seConnecter' , methods = [ 'POST' ] )
def seConnecterUsager() :
    numeroCarte = request.form[ 'numeroCarte' ]
    mdp = request.form[ 'mdp' ]

    if numeroCarte != '' and mdp != '' :
        usager = modeleResanet.seConnecterUsager( numeroCarte , mdp )
        if len(usager) != 0 :
            if usager[ 'activee' ] == True :
                session[ 'numeroCarte' ] = usager[ 'numeroCarte' ]
                session[ 'nom' ] = usager[ 'nom' ]
                session[ 'prenom' ] = usager[ 'prenom' ]
                session[ 'mdp' ] = mdp

                return redirect( '/usager/reservations/list' )

            else :
                return render_template( 'vueConnexionUsager.html', carteBlokuee
        else :
            return render_template( 'vueConnexionUsager.html', carteBlokuee = Fa
    else :
        return render_template( 'vueConnexionUsager.html', carteBlokuee = False
```

En effet, lorsque tous les identifiants sont correctes, la fonction redirige vers une autre fonction de route « /usager /reservations /lister » . Cette fonction est la fonction « listerReservations() » qui elle renvoie la vue « vueListeReservations.html » .Mais lorsque la connexion échoue, la fonction renvoie la vue « vueConnexionUsager.html »

Elle propose donc une redirection (vers une autre fonction) ainsi qu'une génération de vue (vueConnexionUsager.html).