**IFB299 Personal Portfolio**

**Group 27**

**Name: SiuLun Wong**

**Student No.: n8502536**

**Artefact 1 – Fetch role from user**

The RBAC was implemented in the David’s Real Estate website that only allows admin (who is David) to have all permissions while other roles(owner, tenant and property manager) have limited access. In the component/UserIdentity.php, in the authenticate function, “$this->setState('roles', $user->roles);” was added. So when Yii::app()->user->getState(roles) was called, the role of the user will be fetched.

**Artefact 2 – Check Access**

The checkAccess would return true if the user role is admin. “WebUser.php” was created under components folder which overloads the checkAccess() method. In the checkAccess() function, the role is fetched from getState(“roles”) line and if the role is admin, It will return true which admin would have access to everything.

**Artefact 3 – Declaring and calling class**After the WebUser class has been declared in the components folder. And it is essential to ensure this class is called and used by Yii. In main.php which is located in protected/config/main.php, ‘user’=>array(‘class’=>’WebUser’) was added so that this class can be used.

**Artefact 4 – Access Rule**In the controller/userController, the accessRule(), admin and index function is only available to admin by adding (‘roles’=>array(‘admin’). If other function is accessible to other roles such as update is accessible to owner and tenant, it simply adds (‘roles’=>array(‘admin’, ’owner’, ‘tenant’) to update function.

**Artefact 5 – Modify database to store Property Manager**To achieve user story 17 which allows admin to assign property managers to property listings so that the property managers can be contacted by customers. The database has to be altered to add a column of propertyManagerID to propertyListings table which is a foreign key to userID in user table. So the database was rebuilt based on this change