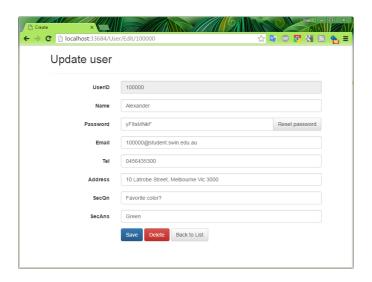
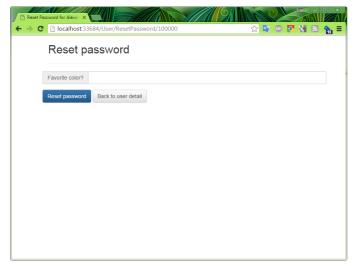
# Pass Task 6.2 Submission form

### Interface template:





## ResetPassword .cshtml

```
<button class="btn btn-primary" ng-click="vm.resetPassword()">Reset
password</button>
                @Html.ActionLink("Back to user detail", "Edit", new { id = Model.UserID }, new {
@class = "btn btn-default" })
            </div>
        </div>
    </div>
</div>
@section Scripts {
    <script src="~/Scripts/angular/angular.min.js"></script>
    <script src="~/Scripts/angular/angular-sanitize.min.js"></script>
    <script>
        angular.module('app', [])
        .controller('accCtrl', ['$http', '$location', function ($http, $location) {
            var vm = this;
            vm.model = {
    UserID: _'@Model.UserID',
                Name: '@Model.Name',
                SecQn: '@Model.SecQn'
                SecAns:
            };
            vm.errorMessage = false;
            vm.resetPassword = function () {
                vm.errorMessage = false;
                // using $http service to create a ajax POST request
                $http.post('/api/UserApi/ResetPassword', vm.model)
                .then(function (success) {
                    // navigate to success page
                    window.location.href = '@Url.Action("ResetPasswordSuccess")';
                }, function (error) {
                    vm.errorMessage = error.data.Message;
                });
            };
        }])
    </script>
```

#### UserController

### UserApiController

```
public class UserApiController : ApiController
   {
       MyUserBOClient _myUserService = null;
       UserAccountQueueClient _myUserAccountQueueService = null;
        public UserApiController()
            _myUserService = new MyUserBOClient("wsHttpBinding_IMyUserBO");
            _myUserAccountQueueService = new UserAccountQueueClient("UserAccountEndpoint");
        [HttpPost]
        public HttpResponseMessage ResetPassword(ResetPasswordRequestViewModel resetPasswordModel)
            var jsonMessage = new JsonResponseMessage();
            if (!ModelState.IsValid)
            {
                jsonMessage.Error("Invalid data", resetPasswordModel);
                return Request.CreateResponse(HttpStatusCode.BadRequest, jsonMessage);
            }
            try
                var resetPasswordDto = new ResetPasswordRequestDto()
                    SecAns = resetPasswordModel.SecAns,
                    UserId = resetPasswordModel.UserID
                };
                var errors = _myUserService.PrecheckForResetPassword(resetPasswordDto);
                if (errors.Length > 0)
                {
                    jsonMessage.Error(string.Join("; ", errors));
                    return Request.CreateResponse(HttpStatusCode.BadRequest, jsonMessage);
                // create a message to do password reset via MSMQ
                MsmqMessage<ResetPasswordRequestDto> resetPwdMsg = new
MsmqMessage<ResetPasswordRequestDto>(resetPasswordDto);
                using (TransactionScope scope = new
TransactionScope(TransactionScopeOption.Required))
                    _myUserAccountQueueService.SendPasswordResetMessage(resetPwdMsg);
                    scope.Complete();
                }
                jsonMessage.Success("Ok", resetPasswordModel);
                return Request.CreateResponse(HttpStatusCode.Accepted, jsonMessage);
            catch (Exception ex)
                jsonMessage.Error("Server error: " + ex.Message);
                return Request.CreateResponse(HttpStatusCode.InternalServerError, jsonMessage);
            }
       }
```

### WCF Configuration in Web.config

#### References:

</system.serviceModel>

(1) BitBucket source code: https://bitbucket.org/werynguyen/swinschool/src/5c746123ab319e06ba450191c9d d63b834a62ce6/?at=PT62

Tutor: Man Lau