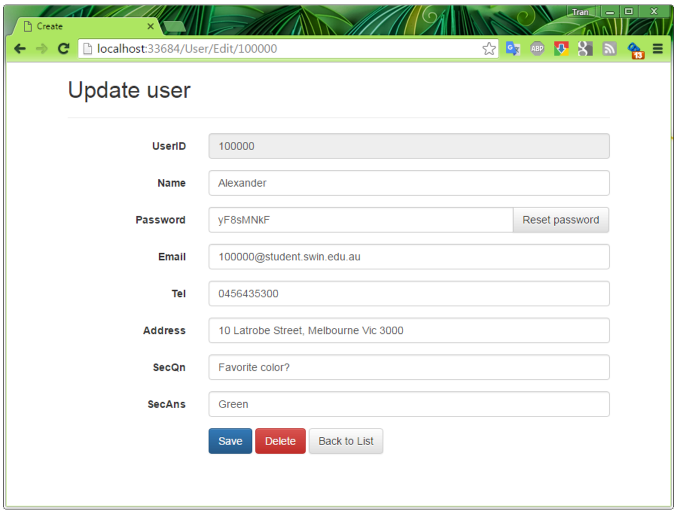
Pass Task 6.2 Submission form

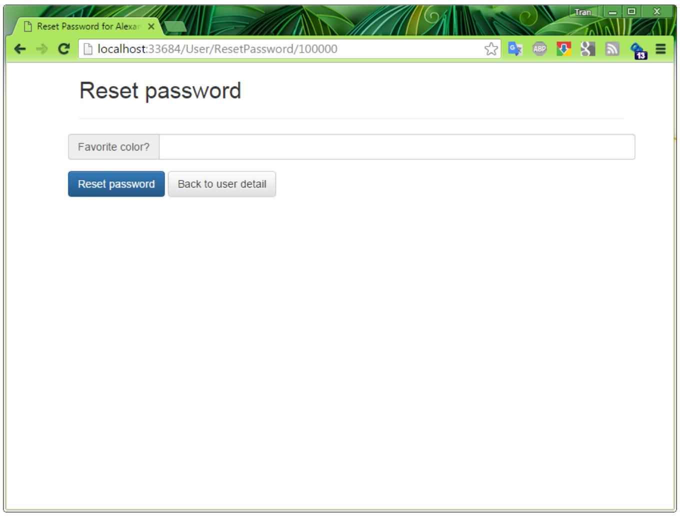
Interface template:



localhost:33684/User/Edit/100000

Update user

UserID	100000
Name	Alexander
Password	yF8sMNkF
Email	100000@student.swin.edu.au
Tel	0456435300
Address	10 Latrobe Street, Melbourne Vic 3000
SecQn	Favorite color?
SecAns	Green



localhost:33684/User/ResetPassword/100000

Reset password

Favorite color?

ResetPassword .cshtml

```
@model SwinSchool.WebUI.Models.ResetPasswordRequestViewModel
@{
    ViewBag.Title = "Reset Password for "+Model.Name;
}
<div class="container" ng-app="app">
    <h2>Reset password</h2>
    <hr />

    <div ng-controller="accCtrl as vm">
        <div class="form form-horizontal">
            <div class="form-group">
                <div class="input-group">
                    <span class="input-group-addon" ng-bind="vm.model.SecQn"></span>
                    <input type="text" class="form-control" ng-model="vm.model.SecAns" />
                </div>

                <div class="error" ng-bind="vm.errorMessage" ng-show="vm.errorMessage"></div>
            </div>
            <div class="form-group">
```

```
                <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

```
public class UserController : Controller
    {
        MyUserBOClient _myUserService = new MyUserBOClient("wsHttpBinding_IMyUserBO");

        ……

        public ActionResult ResetPassword(string id)
        {
            var user = _myUserService.GetUserById(id);

            ResetPasswordRequestViewModel vm = new ResetPasswordRequestViewModel()
            {
                Name = user.Name,
                SecQn = user.SecQn,
                UserID = user.UserID
            };

            return View(vm);
        }

        public ActionResult ResetPasswordSuccess()
        {
            return View();
        }
        ………
    }
```

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

```
<system.serviceModel>
    <bindings>
      <wsHttpBinding>
        <binding name="BindingDefault">
          <reliableSession enabled="true"/>
        </binding>
      </wsHttpBinding>
      <msmqIntegrationBinding>
        <binding name="UserAccountQueueBinding">
          <security mode="None"/>
        </binding>
      </msmqIntegrationBinding>
    </bindings>
    <client>
      <endpoint address="http://localhost:17489/MyUserBO.svc" binding="wsHttpBinding"
          bindingConfiguration="BindingDefault" contract="IMyUserBO"
          name="wsHttpBinding_IMyUserBO" />
      <endpoint address="http://localhost:17489/ShopCartBO.svc" binding="wsHttpBinding"
          bindingConfiguration="BindingDefault" contract="IShopCartBO"
          name="wsHttpBinding_IShopCartBO" />
      <endpoint address="msmq.formatname:DIRECT=OS:.\private$\UserAccountQueue"
binding="msmqIntegrationBinding"
          bindingConfiguration="UserAccountQueueBinding"
contract="SwinSchool.WebUI.Service.IUserAccountQueue"
          name="UserAccountEndpoint" >
      </endpoint>
    </client>
  </system.serviceModel>
```

References:

(1) BitBucket source code: https://bitbucket.org/werynguyen/swinschool/src/a409d44cd5ab81273ffbf83ada1ec4947d109bbd/?at=PT62