UNIT TEST

Table of Unit Test

|  |  |
| --- | --- |
| **Unit Test** | **Test Method** |
| UT-01 | Class AccountController : ActionResult Login(string returnUrl) |
| UT-02 | Class AccountController : ActionResult Login(LoginModel model, string returnUrl) |
| UT-03 | Class AccountController : ActionResult LogOff() |
| UT-04 | Class AccountController : ActionResult Register() |
| UT-05 | Class AccountController : ActionResult Disassociate(string provider, string providerUserId) |
| UT-06 | Class AccountController : ActionResult Manage(ManageMessageId? message) |
| UT-07 | Class AccountController : ActionResult Manage(LocalPasswordModel model) |
| UT-08 | Class AccountController : ActionResult ExternalLogin(string provider, string returnUrl) |
| UT-09 | Class AccountController : ActionResult ExternalLoginCallback(string returnUrl) |
| UT-10 | Class AccountController : ActionResult ExternalLoginConfirmation(RegisterExternalLoginModel model, string returnUrl) |
| UT-11 | Class AccountController : ActionResult ExternalLoginFailure() |
| UT-12 | Class AccountController : ActionResult ExternalLoginsList(string returnUrl) |
| UT-13 | Class AccountController : ActionResult RemoveExternalLogins() |
| UT-14 | Class AdminController : ViewResult IndexManaging() |
| UT-15 | Class AdminController : ViewResult AddWelcomeImage() |
| UT-16 | Class AdminController : ViewResult AddGalleryImage() |
| UT-17 | Class AdminController : ActionResult DeleteWelcomeImage(int ImageID) |
| UT-18 | Class AdminController : ViewResult GalleryManaging() |
| UT-19 | Class AdminController : ViewResult EditBackgroundInfo(int backgroundId) |
| UT-20 | Class AdminController : ViewResult BackgroundManaging() |
| UT-21 | Class AdminController : ViewResult EditActivityClip() |
| UT-22 | Class AdminController : ViewResult ActivityClipManaging() |
| UT-23 | Class AdminController : ActionResult DeleteGalleryImage(int ImageID) |
| UT-24 | Class AdminController : ActionResult DeleteActivityClip(int clipID) |
| UT-25 | Class AdminController : ActionResult IndexManaging(WelcomeImage welcomeImage, HttpPostedFileBase image) |
| UT-26 | Class AdminController : ActionResult Index(GalleryImage galleryImage, HttpPostedFileBase image) |

|  |  |
| --- | --- |
| **Unit Test** | **Test Method** |
| UT-27 | Class AdminController : ActionResult EditBackground(Background background) |
| UT-28 | Class AdminController : ActionResult EditActivityClip(ActivityClip activityClip) |
| UT-29 | Class HomeController : FileContentResult GetWelcomeImage(int imageId) |
| UT-30 | Class HomeController : FileContentResult GetGalleryImage(int imageId) |
| UT-31 | Class HomeController : ActionResult Index() |
| UT-32 | Class HomeController : ActionResult Background() |
| UT-33 | Class HomeController : ActionResult NewsActivities() |
| UT-34 | Class HomeController : ActionResult DhammaContent() |
| UT-35 | Class HomeController : ActionResult DhammaQA() |
| UT-36 | Class HomeController : ActionResult Gallery() |
| UT-37 | Class HomeController : ActionResult ActivityClip() |
| UT-38 | Class HomeController : ActionResult InternetTV() |
| UT-39 | Class HomeController : ActionResult Map() |
| UT-40 | Class HomeController : ActionResult Contact() |

**UNIT TEST**

**UT-01** Class AccountController : ActionResult Login(string returnUrl)

**Test Date** 16 May 2015

**Description** Test for …

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.ReturnUrl = returnUrl  return View(); |  |  |

**UT-02** Class AccountController : ActionResult Login(LoginModel model, string returnUrl)

**Test Date** 16 May 2015

**Description** Test for ...

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | if (ModelState.IsValid && WebSecurity.Login(model.UserName, model.Password, persistCookie: model.RememberMe))  {  return RedirectToLocal(returnUrl);  }  // If we got this far, something failed, redisplay form  ModelState.AddModelError("", "The user name or password provided is incorrect.");  return View(model); |  |  |

**UT-03** Class AccountController : ActionResult LogOff()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | WebSecurity.Logout();  return RedirectToAction("Index", "Home"); |  |  |

**UT-04** Class AccountController : ActionResult Register()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(); |  |  |

**UT-05** Class AccountController : ActionResult Disassociate(string provider, string providerUserId)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | string ownerAccount = OAuthWebSecurity.GetUserName(provider, providerUserId);  ManageMessageId? message = null;  if (ownerAccount == User.Identity.Name){  using (var scope = new TransactionScope(TransactionScopeOption.Required, new TransactionOptions { IsolationLevel = IsolationLevel.Serializable })){  bool hasLocalAccount = OAuthWebSecurity.HasLocalAccount(WebSecurity.GetUserId(User.Identity.Name));  if (hasLocalAccount || OAuthWebSecurity.GetAccountsFromUserName(User.Identity.Name).Count > 1){  OAuthWebSecurity.DeleteAccount(provider, providerUserId);  scope.Complete();  message = ManageMessageId.RemoveLoginSuccess; }}}  return RedirectToAction("Manage", new { Message = message });} |  |  |

**Test Date** 16 May 2015

**Description** Test for…

**UT-06** Class AccountController : ActionResult Manage(ManageMessageId? message)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.StatusMessage =  message == ManageMessageId.ChangePasswordSuccess ? "Your password has been changed."  : message == ManageMessageId.SetPasswordSuccess ? "Your password has been set."  : message == ManageMessageId.RemoveLoginSuccess ? "The external login was removed."  : "";  ViewBag.HasLocalPassword = OAuthWebSecurity.HasLocalAccount(WebSecurity.GetUserId(User.Identity.Name));  ViewBag.ReturnUrl = Url.Action("Manage");  return View(); |  |  |

**UT-07** Class AccountController : ActionResult Manage(LocalPasswordModel model)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | bool hasLocalAccount = OAuthWebSecurity.HasLocalAccount(WebSecurity.GetUserId(User.Identity.Name));  ViewBag.HasLocalPassword = hasLocalAccount;  ViewBag.ReturnUrl = Url.Action("Manage");  if (hasLocalAccount){  if (ModelState.IsValid){  bool changePasswordSucceeded;  try{ changePasswordSucceeded = WebSecurity.ChangePassword(User.Identity.Name, model.OldPassword, model.NewPassword);  } catch (Exception e) { changePasswordSucceeded = false; }  if (changePasswordSucceeded){  return RedirectToAction("Manage", new { Message = ManageMessageId.ChangePasswordSuccess });} else{  ModelState.AddModelError("", "The current password is incorrect or the new password is invalid.”);}}} else{  ModelState state = ModelState[“OldPassword"];  if (state != null) {state.Errors.Clear(); }  if (ModelState.IsValid) { try { WebSecurity.CreateAccount(User.Identity.Name, model.NewPassword);  return RedirectToAction("Manage", new { Message = ManageMessageId.SetPasswordSuccess });} catch (Exception e){ModelState.AddModelError("", e);  return View(model); |  |  |

**UT-08** Class AccountController : ActionResult ExternalLogin(string provider, string returnUrl)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return new ExternalLoginResult(provider, Url.Action("ExternalLoginCallback", new { ReturnUrl = returnUrl })); |  |  |

**UT-09** Class AccountController : ActionResult ExternalLoginCallback(string returnUrl)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | AuthenticationResult result = OAuthWebSecurity.VerifyAuthentication(Url.Action("ExternalLoginCallback", new { ReturnUrl = returnUrl }));  if (!result.IsSuccessful){  return RedirectToAction("ExternalLoginFailure");}if (OAuthWebSecurity.Login(result.Provider, result.ProviderUserId, createPersistentCookie: false)){  return RedirectToLocal(returnUrl);}  if (User.Identity.IsAuthenticated){ OAuthWebSecurity.CreateOrUpdateAccount(result.Provider, result.ProviderUserId, User.Identity.Name);  return RedirectToLocal(returnUrl);} else{  string loginData = OAuthWebSecurity.SerializeProviderUserId(result.Provider, result.ProviderUserId);  ViewBag.ProviderDisplayName = OAuthWebSecurity.GetOAuthClientData(result.Provider).DisplayName;  ViewBag.ReturnUrl = returnUrl;  return View("ExternalLoginConfirmation", new RegisterExternalLoginModel { UserName = result.UserName, ExternalLoginData = loginData }); |  |  |

**UT-10** Class AccountController : ActionResult ExternalLoginConfirmation(RegisterExternalLoginModel model, string returnUrl)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | string provider = null; string providerUserId = null;  if (User.Identity.IsAuthenticated || !OAuthWebSecurity.TryDeserializeProviderUserId(model.ExternalLoginData, out provider, out providerUserId)) {  return RedirectToAction("Manage");}  if (ModelState.IsValid) {  using (UsersContext db = new UsersContext()) {  UserProfile user = db.UserProfiles.FirstOrDefault(u => u.UserName.ToLower() == model.UserName.ToLower());  if (user == null) {  db.UserProfiles.Add(new UserProfile { UserName = model.UserName });  db.SaveChanges();  OAuthWebSecurity.CreateOrUpdateAccount(provider, providerUserId, model.UserName); OAuthWebSecurity.Login(provider, providerUserId, createPersistentCookie: false);  return RedirectToLocal(returnUrl); } else{ ModelState.AddModelError("UserName", "User name already exists. Please enter a different user name.”); }}}  ViewBag.ProviderDisplayName = OAuthWebSecurity.GetOAuthClientData(provider).DisplayName;  ViewBag.ReturnUrl = returnUrl;  return View(model); |  |  |

**UT-11** Class AccountController : ActionResult ExternalLoginFailure()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(); |  |  |

**UT-12** Class AccountController : ActionResult ExternalLoginsList(string returnUrl)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.ReturnUrl = returnUrl;  return PartialView("\_ExternalLoginsListPartial", OAuthWebSecurity.RegisteredClientData); |  |  |

**UT-13** Class AccountController : ActionResult RemoveExternalLogins()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ICollection<OAuthAccount> accounts = OAuthWebSecurity.GetAccountsFromUserName(User.Identity.Name);  List<ExternalLogin> externalLogins = new List<ExternalLogin>();  foreach (OAuthAccount account in accounts){  AuthenticationClientData clientData = OAuthWebSecurity.GetOAuthClientData(account.Provider);  externalLogins.Add(new ExternalLogin {  Provider = account.Provider, ProviderDisplayName = clientData.DisplayName, ProviderUserId = account.ProviderUserId,}); }  ViewBag.ShowRemoveButton = externalLogins.Count > 1 || OAuthWebSecurity.HasLocalAccount(WebSecurity.GetUserId(User.Identity.Name));  return PartialView("\_RemoveExternalLoginsPartial", externalLogins); |  |  |

**UT-14** Class AdminController : ViewResult IndexManaging()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(repository.WelcomeImages); |  |  |

**UT-15** Class AdminController : ViewResult AddWelcomeImage()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(“AddWelcomeImage”, new WelcomeImage); |  |  |

**UT-16** Class AdminController : ViewResult AddGalleryImage()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(“AddGalleryImage”, new GalleryImage); |  |  |

**UT-17** Class AdminController : ActionResult DeleteWelcomeImage(int ImageID)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | WelcomeImage deletedWelcomeImage = repository.DeleteWelcomeImage(ImageID)  if (deletedWelcomeImage != null{  TempData["message"] = "Delete success”; }  return RedirectToAction("IndexManaging"); |  |  |

**UT-18** Class AdminController : ViewResult GalleryManaging()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(repository2.GalleryImages); |  |  |

**UT-19** Class AdminController : ViewResult EditBackgroundInfo(int backgroundId)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Background background = repository3.Backgrounds.FirstOrDefault(p => p.BackgroundID == backgroundId);  return View(background); |  |  |

**UT-20** Class AdminController : ViewResult BackgroundManaging()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(repository3.Backgrounds); |  |  |

**UT-21** Class AdminController : ViewResult EditActivityClip()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(“EditActivityClip”, new ActivityClip()); |  |  |

**UT-22** Class AdminController : ViewResult ActivityClipManaging()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | return View(repository4.ActivityClips); |  |  |

**UT-23** Class AdminController : ActionResult DeleteGalleryImage(int ImageID)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | GalleryImage deletedGalleryImage = repository2.DeleteGalleryImage(ImageID);  if (deletedGalleryImage != null){ TempData["message"] = "Delete success”; }  return RedirectToAction(“Index"); |  |  |

**UT-24** Class AdminController : ActionResult DeleteActivityClip(int clipID)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ActivityClip deletedActivityClip = repository4.DeleteActivityClip(clipID);  if (deletedActivityClip != null) { TempData["message"] = "Delete success”;}  return RedirectToAction("ActivityClipManaging"); |  |  |

**UT-25** Class AdminController : ActionResult IndexManaging(WelcomeImage welcomeImage, HttpPostedFileBase image)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | if (ModelState.IsValid{  if (image != null) {  welcomeImage.ImageMimeType = image.ContentType;  welcomeImage.ImageData = new byte[image.ContentLength];  image.InputStream.Read(welcomeImage.ImageData, 0, image.ContentLength); }  repository.SaveWelcomeImage(welcomeImage);  TempData["message"] = string.Format("{0} has been saved", welcomeImage.ImageID);  return RedirectToAction(“IndexManaging”); } else {  return View(welcomeImage); |  |  |

**UT-26** Class AdminController : ActionResult Index(GalleryImage galleryImage, HttpPostedFileBase image)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | if (ModelState.IsValid){  if (image != null){  galleryImage.ImageMimeType = image.ContentType; galleryImage.ImageData = new byte[image.ContentLength]; image.InputStream.Read(galleryImage.ImageData, 0, image.ContentLength); }  repository2.SaveGalleryImage(galleryImage); TempData["message"] = string.Format("{0} has been saved", galleryImage.ImageID);  return RedirectToAction(“Index"); } else {  return View(galleryImage); |  |  |

**UT-27** Class AdminController : ActionResult EditBackground(Background background)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | if (ModelState.IsValid) {  repository3.SaveBackgroundInfo(background);  return RedirectToAction(“BackgroundManaging"); } else {  return View(background); |  |  |

**UT-28** Class AdminController : ActionResult EditActivityClip(ActivityClip activityClip)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | if (ModelState.IsValid) {  repository4.SaveActivityClip(activityClip);  return RedirectToAction(“ActivityClipManaging”); } else {  return View(activityClip); |  |  |

**UT-29** Class HomeController : FileContentResult GetWelcomeImage(int imageId)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | WelcomeImage welimg = repository.WelcomeImages.FirstOrDefault(p => p.ImageID == imageId);  if (welimg != null) {  return File(welimg.ImageData, welimg.ImageMimeType);  } else {  return null;}} |  |  |

**UT-30** Class HomeController : FileContentResult GetGalleryImage(int imageId)

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | GalleryImage galimg = repository2.GalleryImages.FirstOrDefault(p => p.ImageID == imageId);  if (galimg != null) {  return File(galimg.ImageData, galimg.ImageMimeType);  } else {  return null; } } |  |  |

**UT-31** Class HomeController : ActionResult Index()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “หน้าแรก”;  return View(repository.WelcomeImages); |  |  |

**UT-32** Class HomeController : ActionResult Background()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “ประวัติวัด”;  return View(repository3.Backgrounds); |  |  |

**UT-33** Class HomeController : ActionResult NewsActivities()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “ข่าวสาร-กิจกรรม”;  return View(); |  |  |

**UT-34** Class HomeController : ActionResult DhammaContent()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “สาระธรรม”;  return View(); |  |  |

**UT-35** Class HomeController : ActionResult DhammaQA()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “ถาม-ตอบธรรม”;  return View(); |  |  |

**UT-36** Class HomeController : ActionResult Gallery()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “แกลอรี่";  return View(repository2.GalleryImages); |  |  |

**UT-37** Class HomeController : ActionResult ActivityClip()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “วิดีโอกิจกรรม";  return View(repository4.ActivityClips); |  |  |

**UT-38** Class HomeController : ActionResult InternetTV()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “อินเตอร์เน็ตทีวี";  return View(); |  |  |

**UT-39** Class HomeController : ActionResult Map()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “แผนที่วัด”;  return View(); |  |  |

**UT-40** Class HomeController : ActionResult Contact()

**Test Date** 16 May 2015

**Description** Test for…

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | ViewBag.Message = “ติดต่อวัด”;  return View(); |  |  |