Steps: Service Provider goes to the website Accept User is propmted to log in or register ASSUME: Registered This will keep track of each service a user has After user is User eneters username and password Data Store: SevicesDB successfully logged in **EntryPoint** Manager: Service: Data Access: user is successfully logged in ServiceRequest/UserService Class/Object Class/Object Class/Object Class/Object and viewing service ServiceController ServiceManager RequestManagement RequestDAO user sees all the requests made Homepage (table) Result User Service Request UserService request User is able/sees accept and decline options User selects accept: User clicks accept public Task<List<ServiceRequest>> the service is added to the users profile/services ServiceManager serviceManager = AcceptRequest() the Property manager is notified new ServiceManager() he user is notified that the service was successfully accepted User Selects Decline: Ctor() return ServiceManager the user is notified that the service was successfully declined var requst = new ServiceRequest(); Ctor() return User instance public Task<List<SeviceRequest>> AddServiceRequests(request) RequestManagement reqMan = new RequestManagement(); return RequestManagement instance var result = new Result(); SmtpClient Ctor() Result result = return Result instance reqMan.AddUserService(request) : Result var result = new Result(); Ctor() return Result instance RequestDAO _requestDao = new RequestDAO(); Ctor() return RequestDAO instance var service = new UserService(); Ctor() return UserService instance List<ServiceRequest> serviceRequest = _requestDAO.GetUserRequest Procedure: AcceptServiceRequest (request, status, rating) var command = new SqlCommand(sqlQuery, connection) CREATE procedure AcceptServiceRequest BEGIN SqlDataReader reader = INSERT INTO dbo.UserServices(UserServiceID, ServiceName, command.ExecuteReader() SqlCommand exec = CreateCommand() ServiceType, ServiceDescription, ServiceFrequency, exec.CommandText = "AcceptServiceRequest" all rows that matches ServiceProviderName, ServiceProviderEmail, _exec.CommandType = Service Provider ID PropertyManagerName, PropertyManagerEmail) System.Data.CommandType.StoredProcedure; VALUES(user_service_ld, service_name, service_type, Note: Notify service_description, service_frequency, service_provider_name, return result service_provider_email, property_manager_name, user via return result System.Net.Mail property_manager_email) return result.Pavload return List<ServiceRequest> DELETE FROM dbo.ServiceRequest public async Task AutomaticEmail WHERE ServiceRequestID = service_request_id var smtpClient = new SmtpClient (string userEmail, string emailBody) : boo ("smtp.gmail.com", 587); SELECT * FROM dbo.ServiceRequest WHERE ServiceProviderID = service_provider_Id system sends email return smtp client END return true return result.Pavload return List<ServiceRequest> display list of service request Success Cases for updating a service request: Decline Data Store: SevicesDB EntryPoint Service: Data Access: Manager: ServiceRequest Class/Object Class/Object Class/Object Homepage ServiceController ServiceManager RequestManagement RequestDAO (table) Result Service Request [HttpGet] User clicks decline public Task<List<ServiceRequest>> ServiceManager serviceManager = DeclineRequest() new ServiceManager() Ctor() var requst = new ServiceRequest(); Ctor() return ServiceRequest instance public Task<List<SeviceRequest>> RequestManagement reqMan = RemoveServiceRequests(request) new RequestManagement(); return RequestManagement instance var result = new Result(); Result result = return Result instance reqMan.RemoveRequest(request) : Result var result = new Result(); Ctor() return Result instance RequestDAO _requestDao = new RequestDAO(); return RequestDAO instance var serviceRequest = new ServiceRequest(); Ctor() return serviceRequest instance List<ServiceRequest> serviceRequest = _requestDAO.GetUserRequest Note: should have (request, status, rating) Procedure: DeclineServiceRequest var command = new SqlCommand(sqlQuery, connection) decline/accept CREATE procedure DeclineServiceRequest SqlDataReader reader = DELETE FROM dbo.ServiceRequest command.ExecuteReader() SqlCommand exec = CreateCommand() WHERE ServiceRequestID = service_request_id all rows that matches exec.CommandText = "AcceptServiceRequest" Service Provider ID exec.CommandType = SELECT * FROM dbo.ServiceRequest System.Data.CommandType.StoredProcedure; WHERE ServiceProviderID = service_provider_Id return result END return result.Pavload return List<ServiceRequest> return List<ServiceRequest> return result.Payload ask for the request for a display list of service request specific user/ SP

Success Cases for updating a service request: