



System

Name ----- ID

S M IFTAKHAIRUL-----1421447042

MALIHA ALAM----- 1421399042

RAKIBUL ISLAM -----1511040642

MD. TANVIR ISLAM-----1421463042

Introduction

The name of our project is Visitor Management System.

Visitor Management System is a web application. The aim of our web application is to allow visitors to request for appointments and visits.

Admin will manage visitors and whether the visiting requests of the visitors will be approved or not, this will depend on admin.

Host will manage the appointment requests made by visitors.

After going to this web page every single new visitor have to sign up for creating account. Only after doing that, visitors will be able to request for appointments and visits. They can see the status of their requests (whether approved or not) and the visiting or appointment time table via logging into their accounts.

If the requests of visitors are approved by admin, admin will give each and every visitor an access code. At the day of the visitors visiting's or appointments, admin will check whether they are real users or not through those access codes.

Visitors can also update their profiles whenever they want to do that by logging into their accounts.

Overview

Our team designed this visitor management system web application. During each iteration, we completed different use cases to get the application done.

In our project, there are three types of users-Admin, Host, and Visitor.

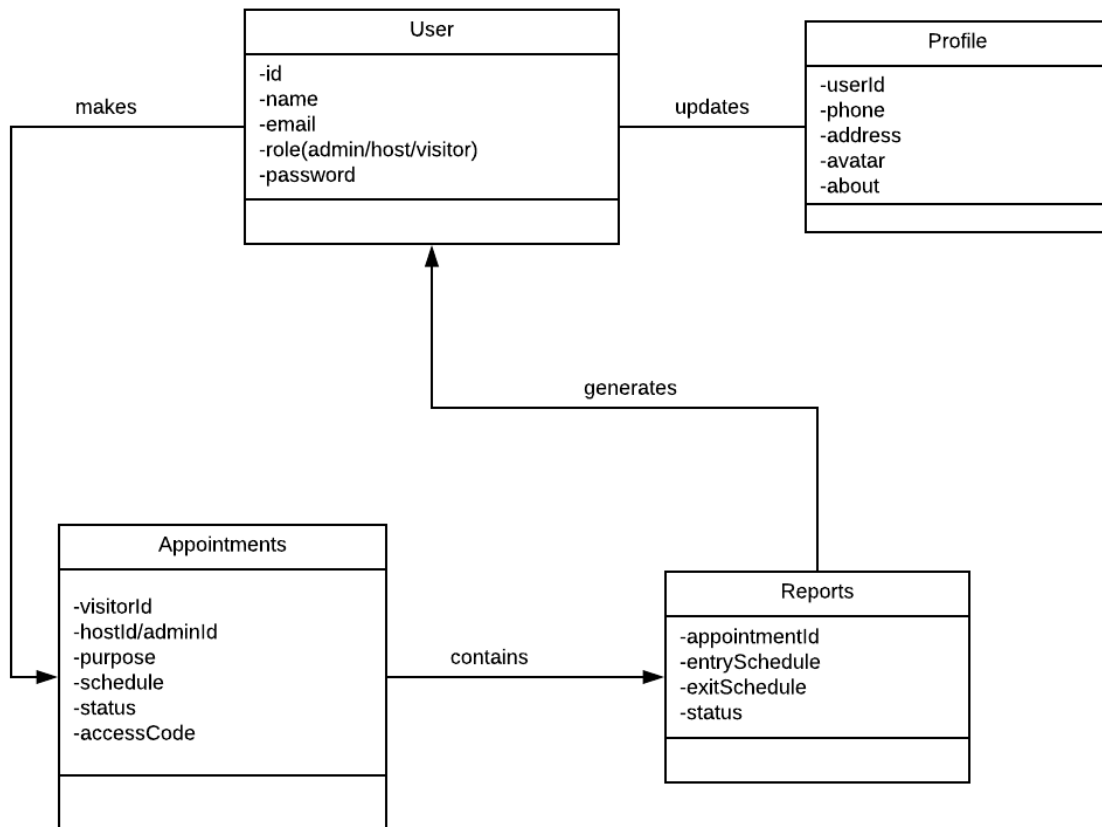
The use cases which we have designed for Visitor type user's are-login, signup, request for appointment, and request for visit

Use cases designed for Admin type user are-Manage visitors, manage visit request, Add Designation, Manage Designation, Generate Reports, Add Host, Manage Host

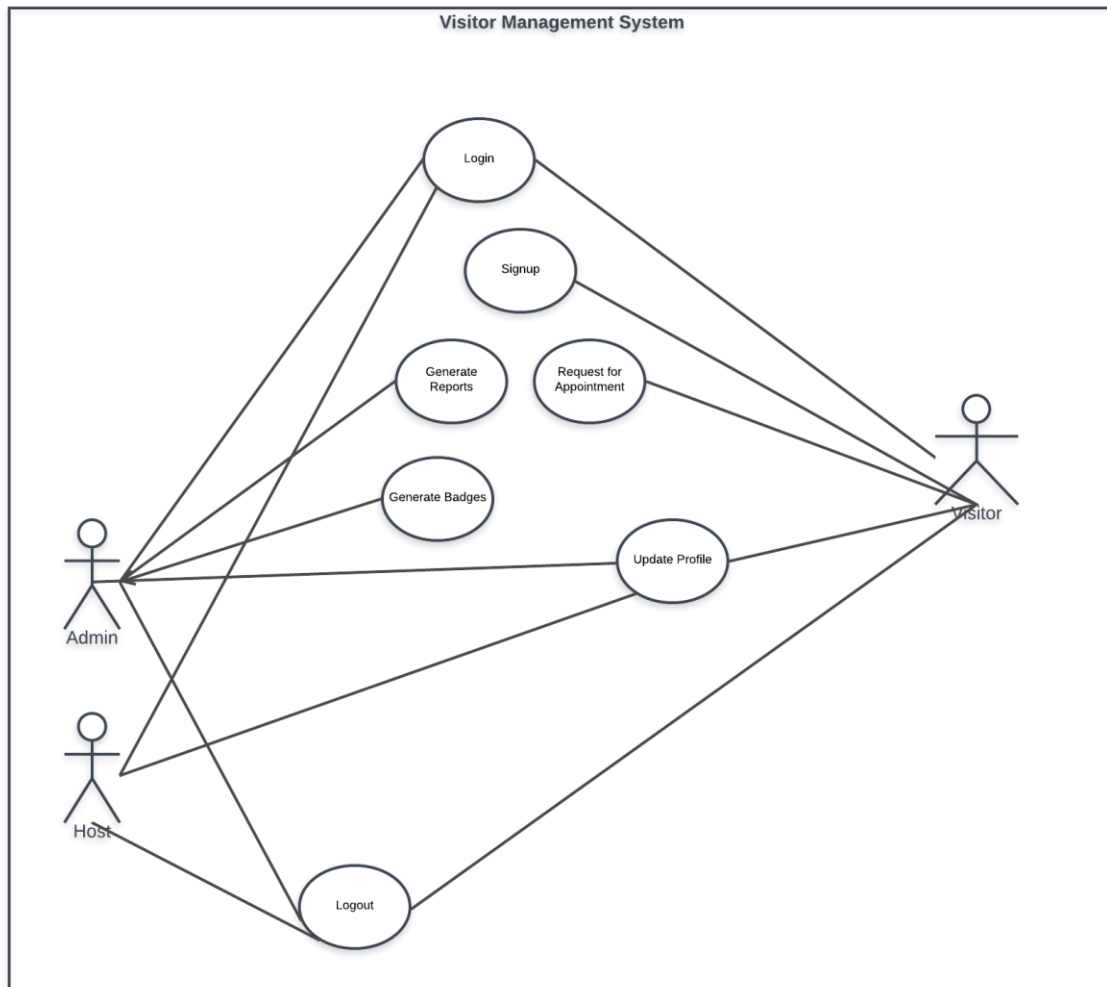
Use cases designed for Host type user is-Manage Appointment request.

Common Use Cases for all type users are-Login, Inbox, Profile, Logout

Domain Model Diagram



Use Case Diagram



Use Case Description

Use Case-1[SignUpUser]:

ID:	SignupForUser
Name:	SignUp User
Level:	User Goal
Description:	Only visitor can register by himself/herself. Admin and host are not allowed to register by self. Visitor will confirm email to activate account.
Primary actor:	User
Precondition:	
Post condition	Visitor signs up a new account.
Main success scenario:	<ol style="list-style-type: none">1. System provides signup interface.2. User accesses the signup interface.3. User fills the required section for signup.4. User adds to system.5. User receives email confirmation message.6. User redirects back to login interface.
Extension:	<ol style="list-style-type: none">3a. System not accepting user credentials.<ol style="list-style-type: none">3a1. User reenters credentials.3b. System shows invalid email address message.<ol style="list-style-type: none">3b1. User reenters email address in correct format.3c. System shows password not matched message.<ol style="list-style-type: none">3c1. User reenters same password correctly.5a. System fails to send confirmation email & shows message.<ol style="list-style-type: none">5a1. User requests system to resend email.
Frequency of use:	The first time of a new visitor.

Use Case-2[LoginUser]:

ID:	LoginForUser
Name:	Login User
Level:	User Goal
Description:	All users can login. Email and password is required to log in the system.
Primary actor:	User
Precondition:	User must be registered before.
Post condition	User enters home page.
Main success scenario:	<ol style="list-style-type: none">1. System provides login interface.2. User accesses the login interface.3. User fills the required section for login.4. User redirects to dashboard.
Extension:	<ol style="list-style-type: none">3a. System not accepting user credentials.<ol style="list-style-type: none">3a1. User reenters credentials.3b. System shows invalid email address message.<ol style="list-style-type: none">3b1. User reenters email address in correct format.3c. System shows email and password not matched message.<ol style="list-style-type: none">3c1. User reenters correct password of his/her account.
Frequency of use:	Seldom

Use Case-3[UpdateUserProfiles]:

ID:	UpdateUserProfiles
Name:	Update User Profiles
Level:	User Goal
Description:	User updates his/her profiles.
Primary actor:	User
Precondition:	User must have account.
Post condition	
Main success scenario:	<ol style="list-style-type: none"> 1. System provides profile update interface. 2. User accesses the profile update section. 3. Updated user credentials. 4. User performs profile update by confirming password. 5. User redirects back to profile section with updated credentials.
Extension:	<ol style="list-style-type: none"> 3a. System not accepting user credentials. <ol style="list-style-type: none"> 3a1. User reenters credentials. 3b. System shows all required field was not filled up. <ol style="list-style-type: none"> 3b1. User fills the missing field. 3c. System not accepting file format for user's profile picture. <ol style="list-style-type: none"> 3c1. User's enter valid file format (only image file). 3d. System shows invalid characters errors. <ol style="list-style-type: none"> 3d1. User fills the section correctly.
Frequency of use:	Seldom

Use Case-4[RequestForAppointment]:

ID:	AppointmentRequest
Name:	Request For Appointment
Level:	User Goal
Description:	User requests for appointment. Admin/Host handles the requests and user gets updated information.
Primary actor:	User(visitor)
Precondition:	User must have an account with updated profile.
Post condition	User waits for his/her request approval by admin/host.
Main success scenario:	<ol style="list-style-type: none"> 1. User accesses appointment interface provided by system. 2. User enters appointment request section. 3. User fulfills all section required. 4. User selects admin for visit request or particular host for appointment. 5. User's request sends to admin/host. 6. Email sends to admin/host about appointment request. 7. User redirects to appointment interface.
Extension:	<ol style="list-style-type: none"> 3a. System not accepting user's appointment request. <ol style="list-style-type: none"> 3a1. User again fills the appointment request form correctly. 3b. System shows all required field was not filled up. <ol style="list-style-type: none"> 3b1. User fills the missing field. 3c. System shows appointment purpose was invalid. <ol style="list-style-type: none"> 3c1. User selects the valid reason for appointment. 3d. System shows invalid appointment schedule (date, time) message. <ol style="list-style-type: none"> 3d1. User selects valid appointment schedule. 4a. System shows host will be busy at requested schedule message. <ol style="list-style-type: none"> 4a1. User selects schedule at which host won't be busy. 4b. System shows host will be unavailable at requested schedule

	<p>message.</p> <p>4b1. User selects schedule at which host won't be unavailable.</p> <p>4c. System shows time clashes error message.</p> <p>4c1. User selects valid schedule.</p> <p>6a. System fails to send email & shows message.</p> <p>6a1. User requests system to resend email.</p>
Frequency of use:	Often

Use Case-5[GenerateBadges]:

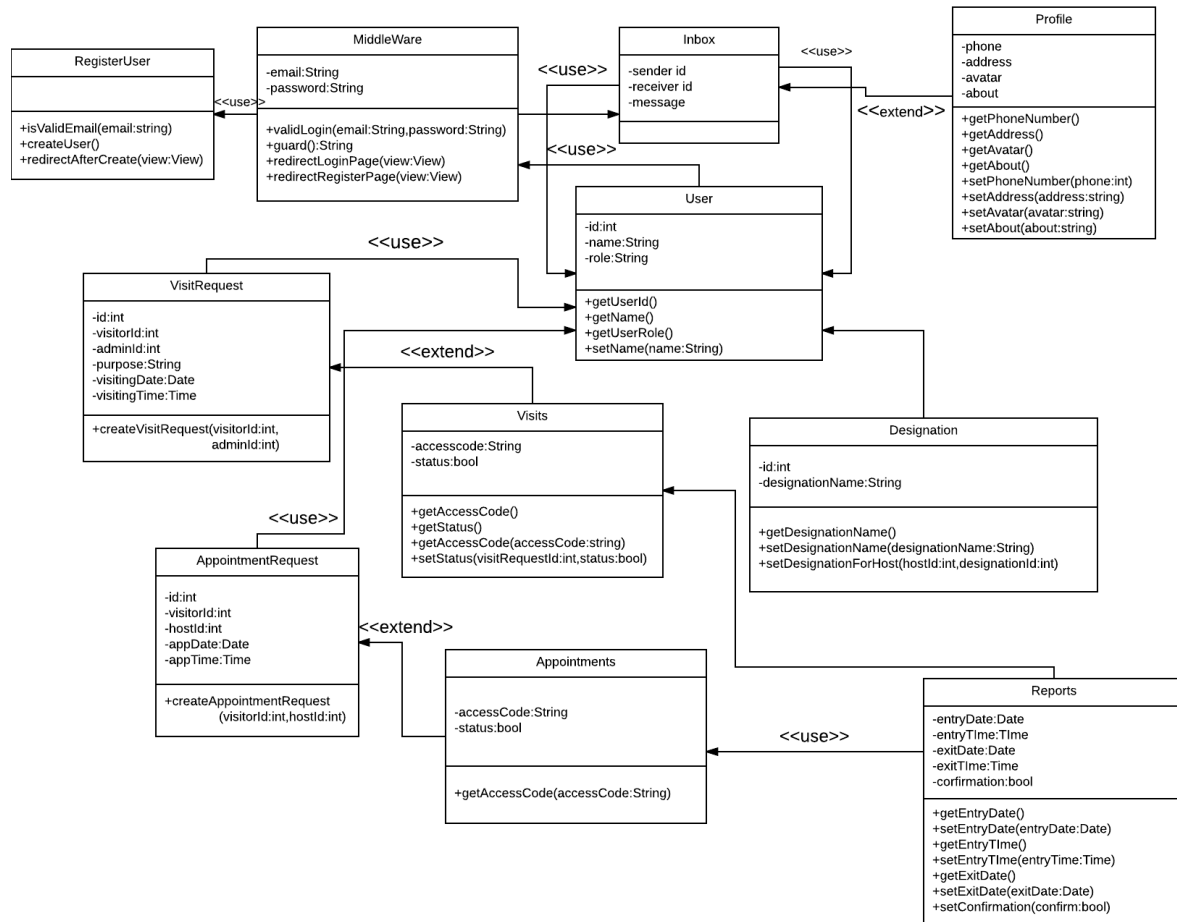
ID:	GenerateBadges
Name:	GenerateBadges
Level:	User Goal
Description:	A detail badge generates for after appointment request approval. Badge contains a unique badge code by which user can enter the organization.
Primary actor:	User
Precondition:	User's appointment request must be accepted by admin/host.
Post condition	
Main success scenario:	<ol style="list-style-type: none"> 1. Admin/host accesses to Appoint Request interface. 2. Admin/host accepts user's request. 3. Mail sends to user's email about appointment approval. 4. A report with appointment schedule, unique badge code & user's basic information generates by system. 5. User's access to Approved Appointment interface. 6. User's downloads the badge report.
Extension:	<ol style="list-style-type: none"> 2a. System shows admin/host will be busy message. <ol style="list-style-type: none"> 2a1. User requests to admin/host again at different schedule. 2b. System shows inappropriate appointment purpose message. <ol style="list-style-type: none"> 2b1. User requests to admin/host again with valid purpose. 2c. System shows user's profile not updated message. <ol style="list-style-type: none"> 2c1. User updates his/her profile and request again. 2d. System shows organization will be closed at requested schedule message. <ol style="list-style-type: none"> 2d1. User requests again at which organization won't be closed. 3a. System fails to send email & shows message. <ol style="list-style-type: none"> 3a1. User requests system to resend email. 4a. System shows badge generate failed message. <ol style="list-style-type: none"> 4a. User requests system to fix this and regenerate badge. 6a. User finds downloadable link disabled. <ol style="list-style-type: none"> 6a. User requests to system to fix this.
Frequency of use:	Often

Use case-6 [GenerateReports]:

ID:	GenerateRports
Name:	GenerateReports
Level:	Admin Goal
Description:	A details report generate after user's entry to the organization
Primary actor:	Admin
Precondition:	User's appointment request must be accepted and he/she must have

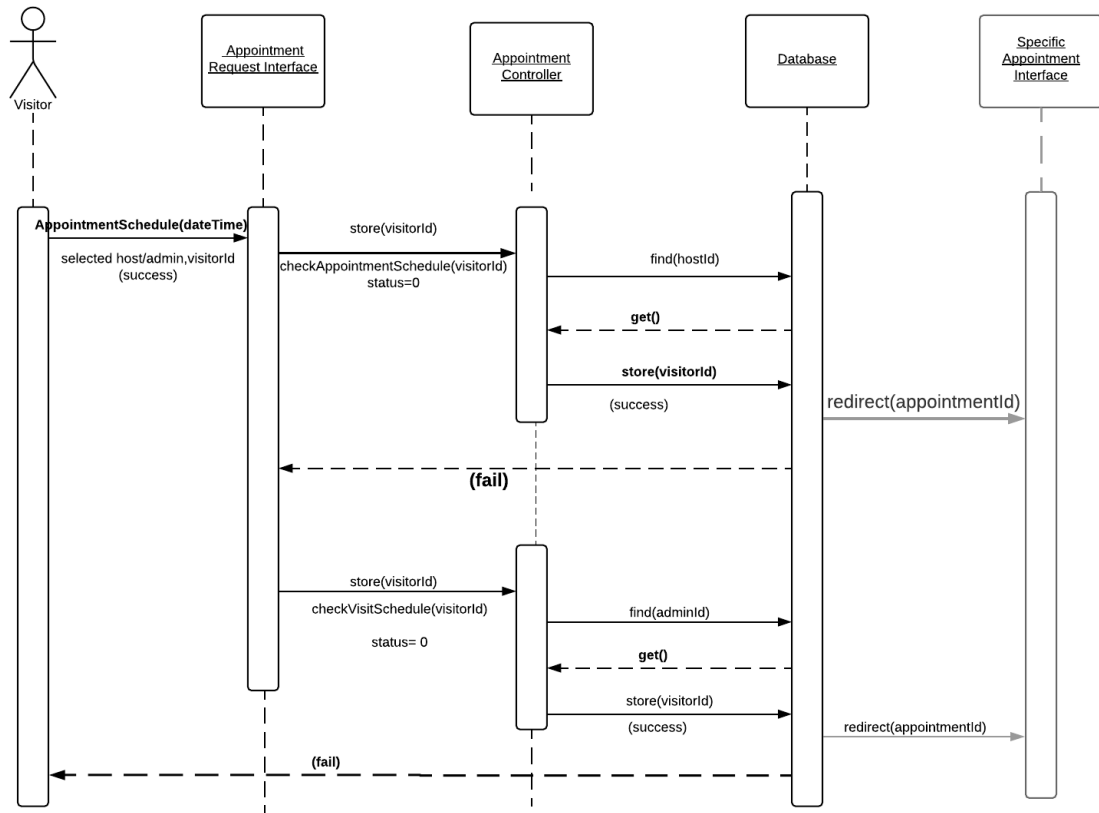
	badge report.
Post condition	
Main success scenario:	<ol style="list-style-type: none"> 1. User brings the badge report. 2. Badge code is scanned by reception. 3. Admin allows user to enter. 4. System updates user's entry and exit schedule to report. 5. User's appointment status marks as completed.
Extension:	<ol style="list-style-type: none"> 2a. Admin finds badge code was previously used. <ol style="list-style-type: none"> 2a1. Admin closes the task completely. 2b. Admin finds fake user of the appointment request. <ol style="list-style-type: none"> 2b1. Admin takes immediate action of that user. 3a. Admin doesn't allow user to enter. <ol style="list-style-type: none"> 3a1. Admin warns user. Admin tells user to make valid appointment request. 4a. Admin sees report failed message. <ol style="list-style-type: none"> 4a1. Admin checks the system and fixes it. 5a. Admin sees appointment status update failed message. <ol style="list-style-type: none"> 5a1. Admin checks the system and fixes it.
Frequency of use:	Often

Class Diagram

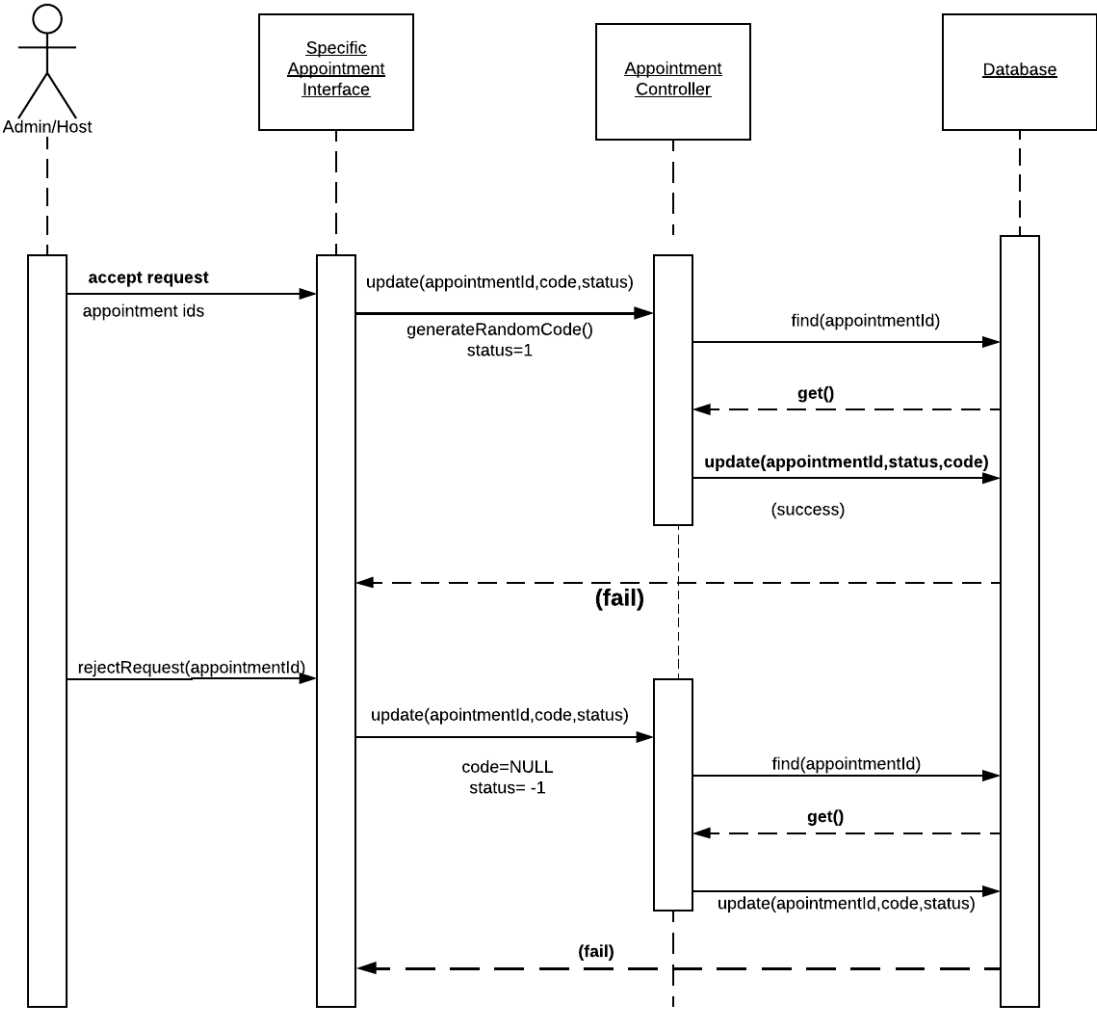


Sequence Diagrams

Sequence Diagram 1



Sequence Diagram 2



Sequence Diagram 3

