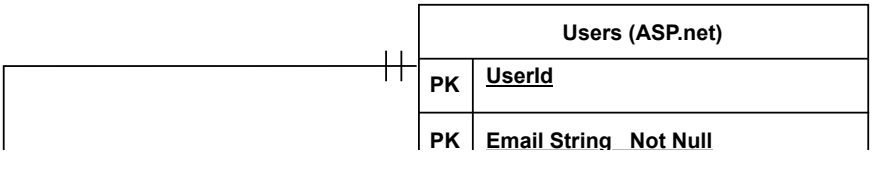
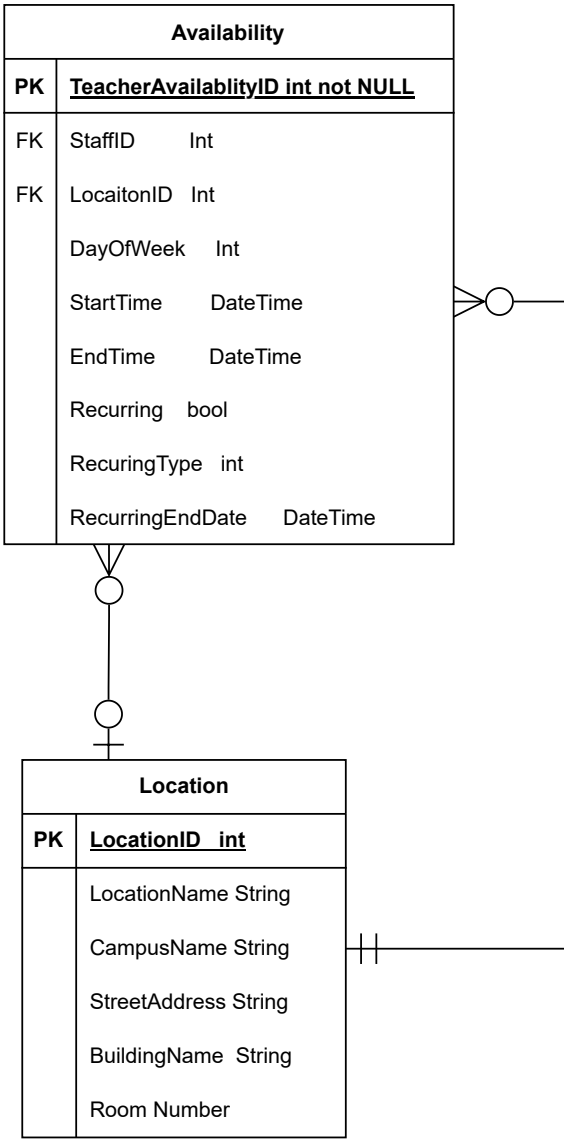
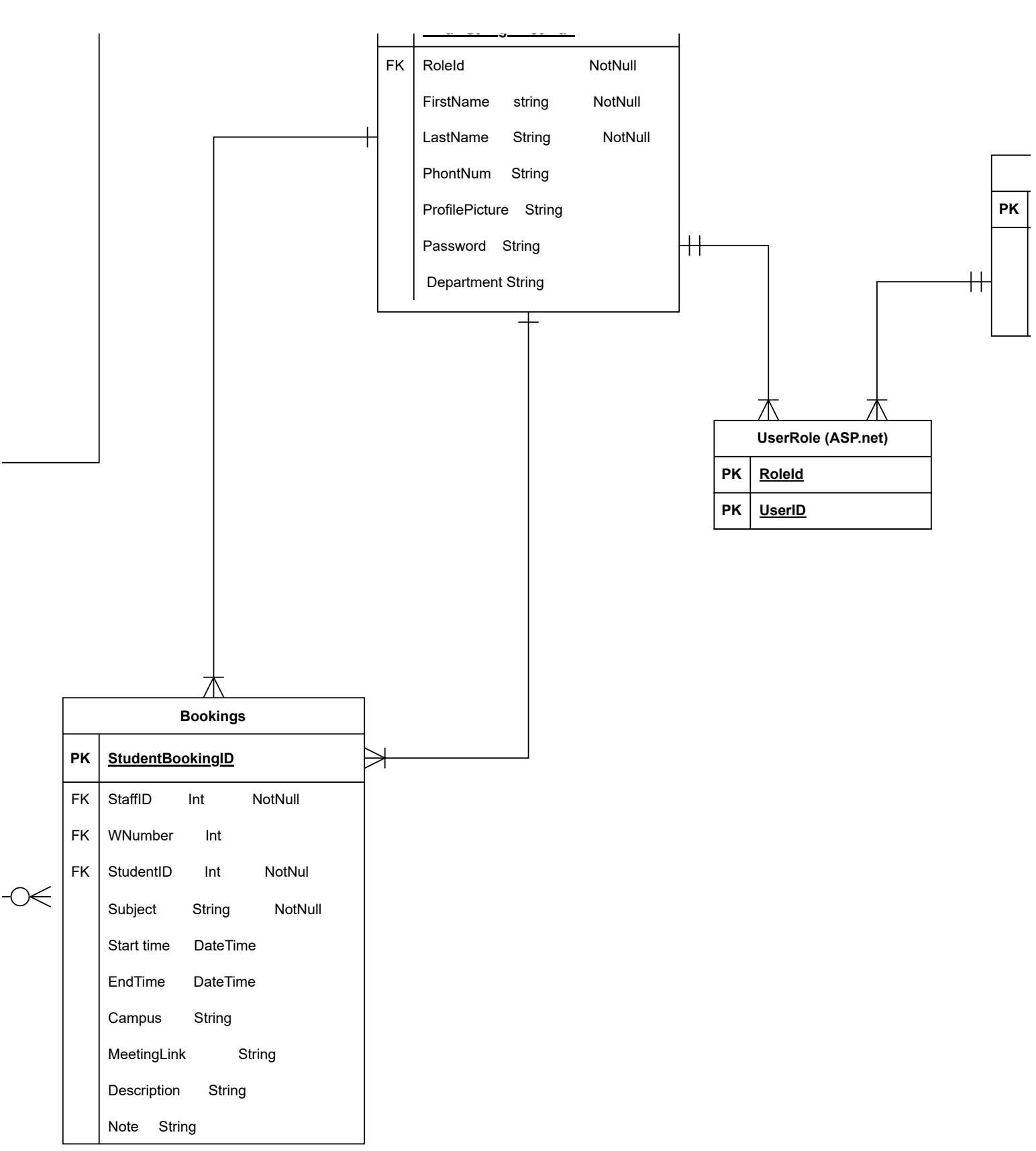
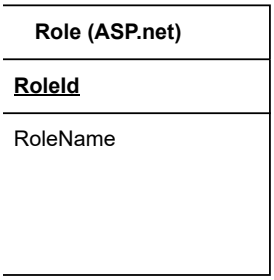
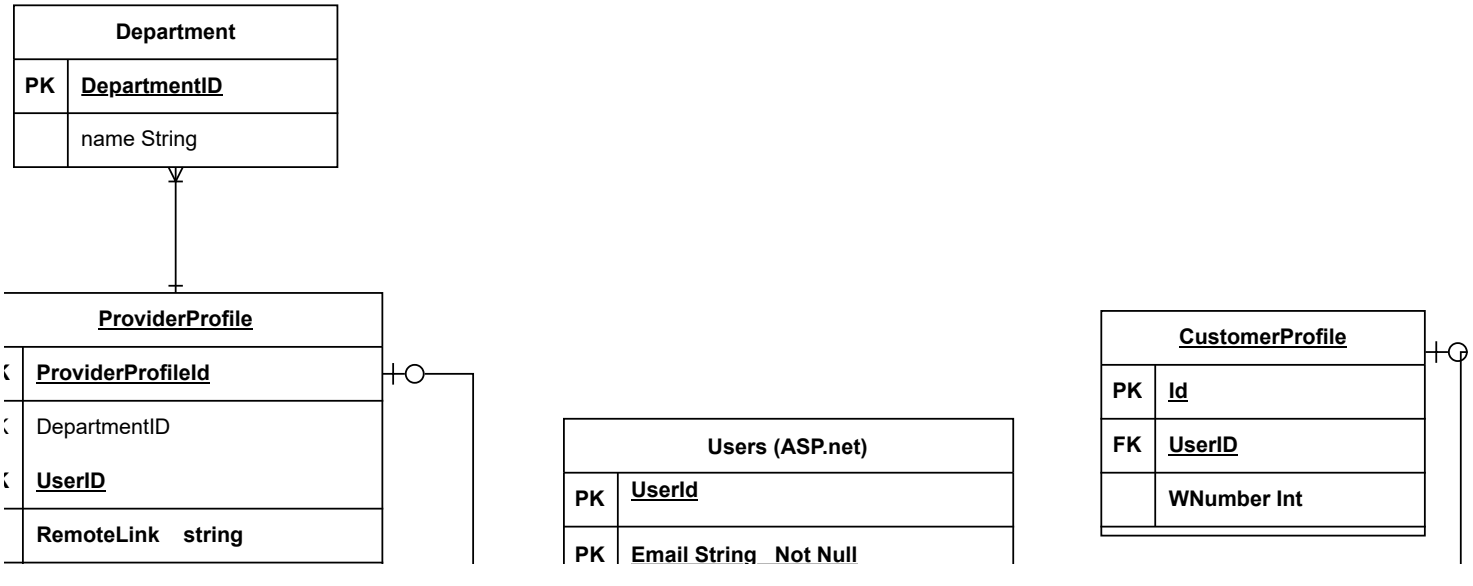

Users (ASP.net)
PK
UserId
PK
Email String Not Null

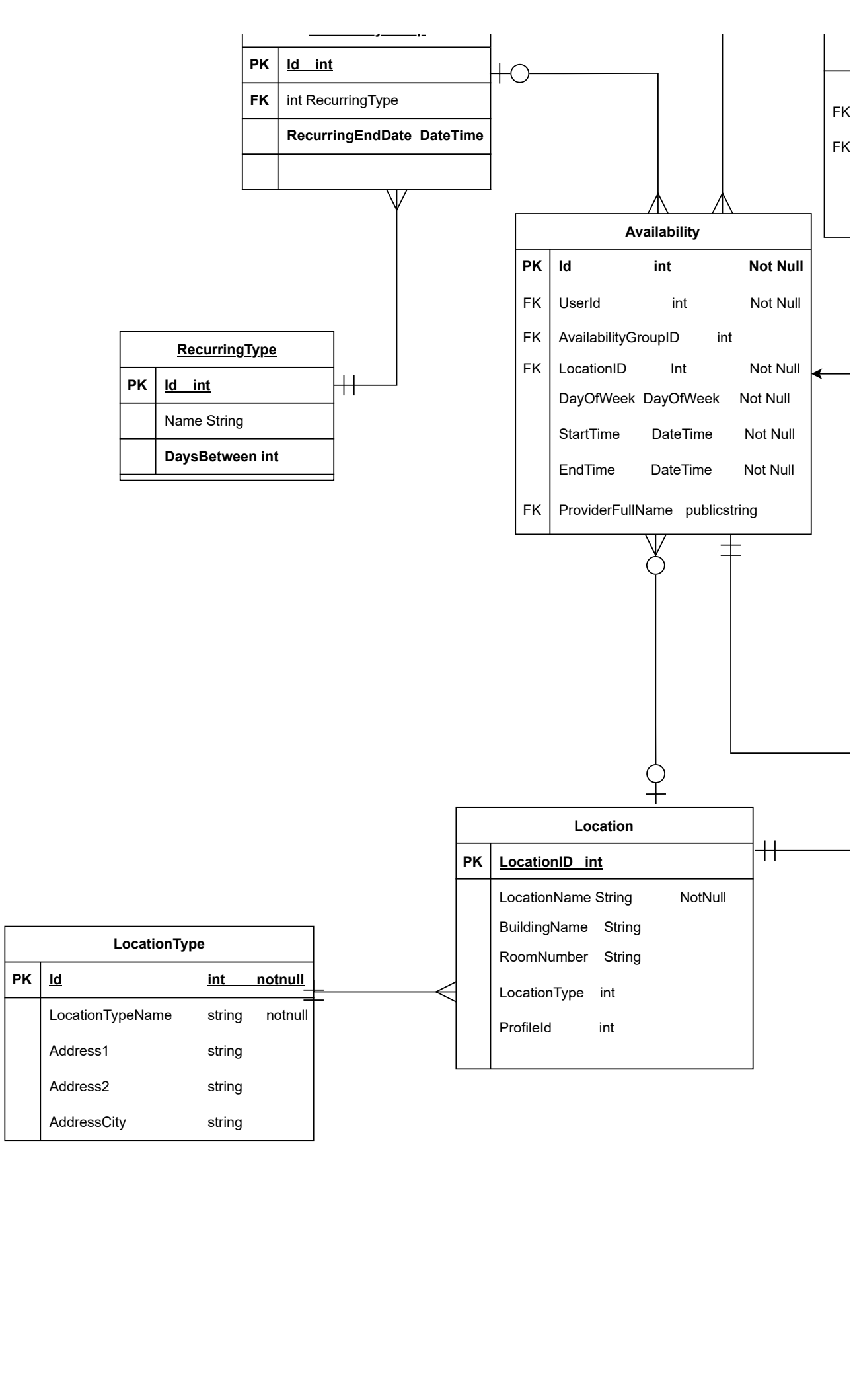

Availability
PK
TeacherAvailablityID int not NULL
FK
StaffID Int
FK
LocaitonID Int
DayOfWeek Int
StartTime DateTime
EndTime DateTime
Recurring bool
RecuringType int
RecurringEndDate DateTime
Location
PK
LocationID int
LocationName String
CampusName String
StreetAddress String
BuildingName String
Room Number

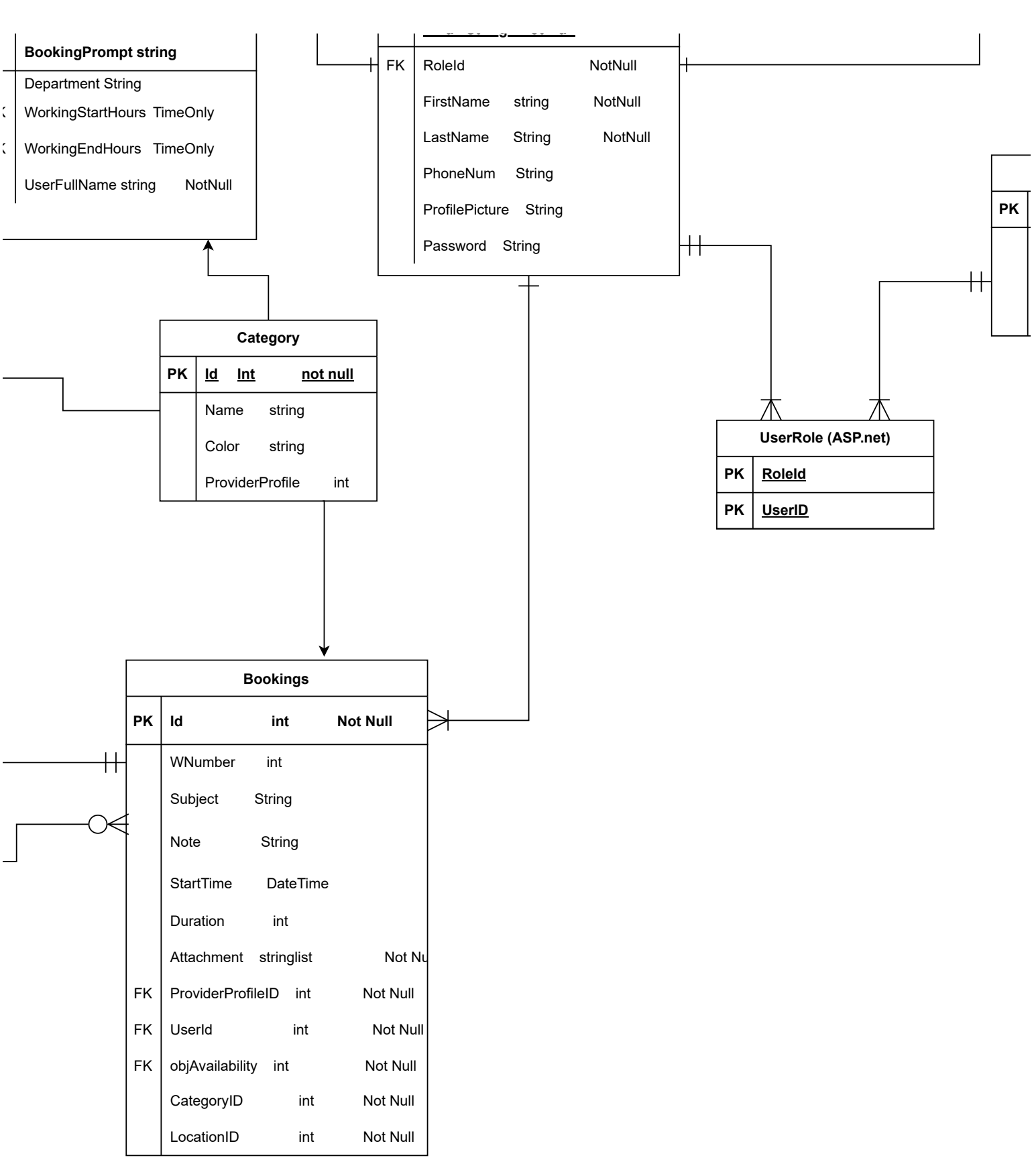

FK RoleId NotNull
FirstName string NotNull
LastName String NotNull
PhontNum String
ProfilePicture String
Password String
Department String
PK
UserRole (ASP.net)
PK RoleId
PK UserID
Bookings
PK StudentBookingID
FK StaffID Int NotNull
FK WNumber Int
FK StudentID Int NotNul
Subject String NotNull
Start time DateTime
EndTime DateTime
Campus String
MeetingLink String
Description String
Note String

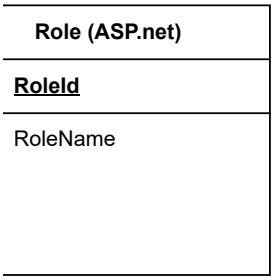

Role (ASP.net)
RoleId
RoleName

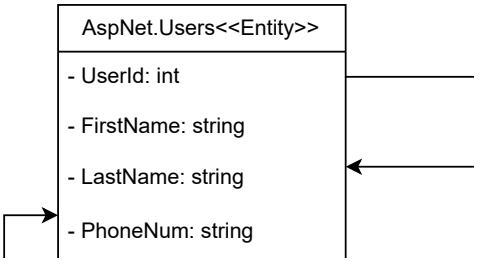
AvailabilityGroup
PK
FK
FK

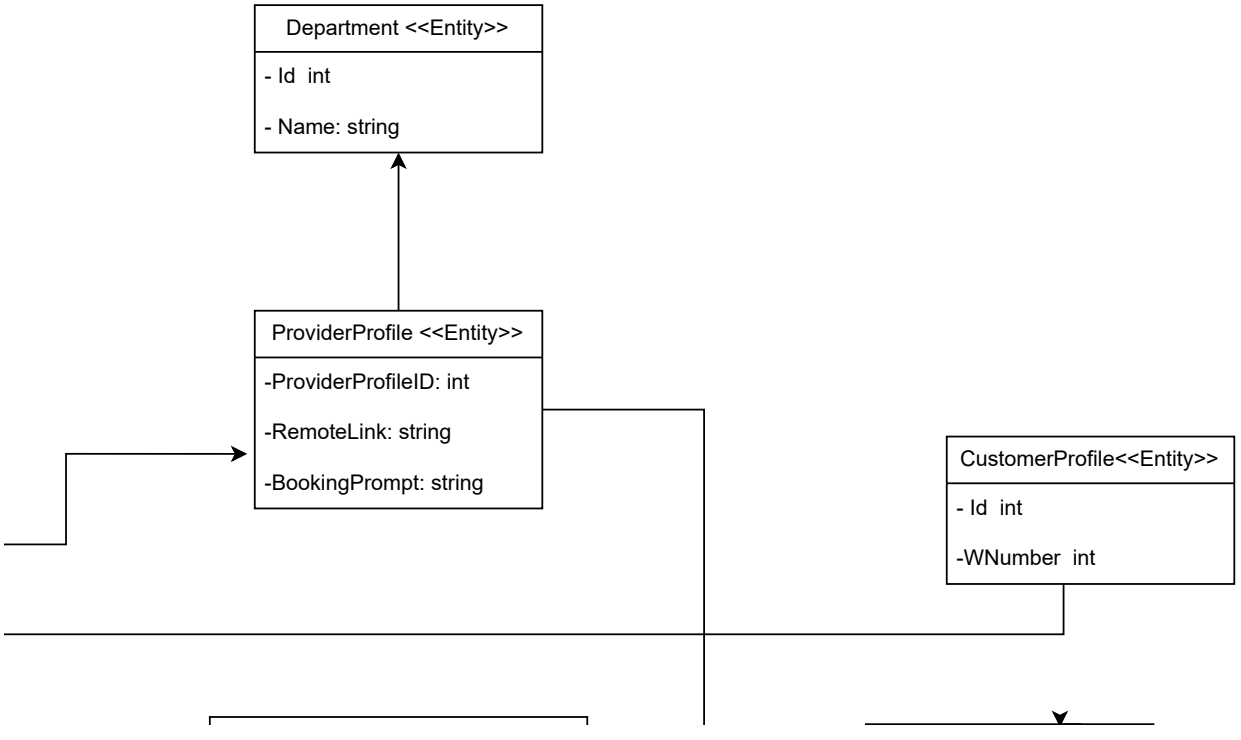

Department
PK
DepartmentID
name String
ProviderProfile
ProviderProfileId
DepartmentID
UserID
RemoteLink string
Users (ASP.net)
PK
UserId
PK
Email String Not Null
CustomerProfile
PK
Id
FK
UserID
WNumber Int

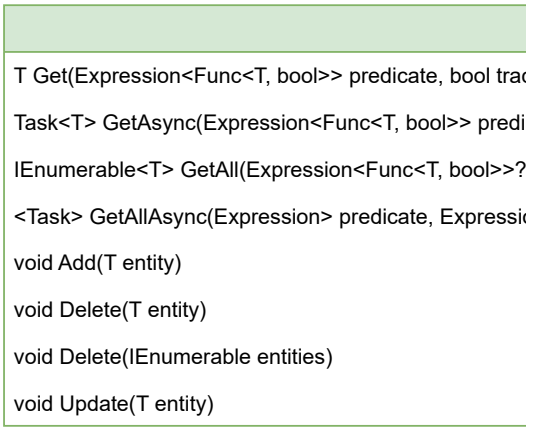

PK Id int
FK int RecurringType
RecurringEndDate DateTime
FK
FK
Availability
PK Id int Not Null
FK UserId int Not Null
FK AvailabilityGroupID int
FK LocationID Int Not Null
DayOfWeek DayOfWeek Not Null
StartTime DateTime Not Null
EndTime DateTime Not Null
FK ProviderFullName publicstring
RecurringType
PK Id int
Name String
DaysBetween int
Location
PK LocationID int
LocationName String NotNull
BuildingName String
RoomNumber String
LocationType int
ProfileId int
LocationType
PK Id int notnull
LocationTypeName string notnull
Address1 string
Address2 string
AddressCity string

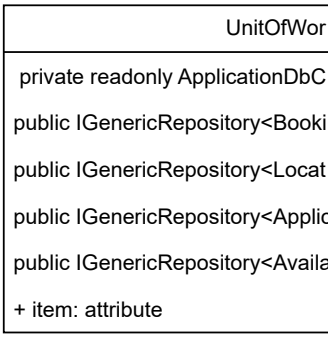

BookingPrompt string
Department String
WorkingStartHours TimeOnly
WorkingEndHours TimeOnly
UserFullName string NotNull
FK RoleId NotNull
FirstName string NotNull
LastName String NotNull
PhoneNum String
ProfilePicture String
Password String
PK
Category
PK Id Int not null
Name string
Color string
ProviderProfile int
UserRole (ASP.net)
PK RoleId
PK UserID
Bookings
PK Id int Not Null
WNumber int
Subject String
Note String
StartTime DateTime
Duration int
Attachment stringlist Not Nu
FK ProviderProfileID int Not Null
FK UserId int Not Null
FK objAvailability int Not Null
CategoryID int Not Null
LocationID int Not Null

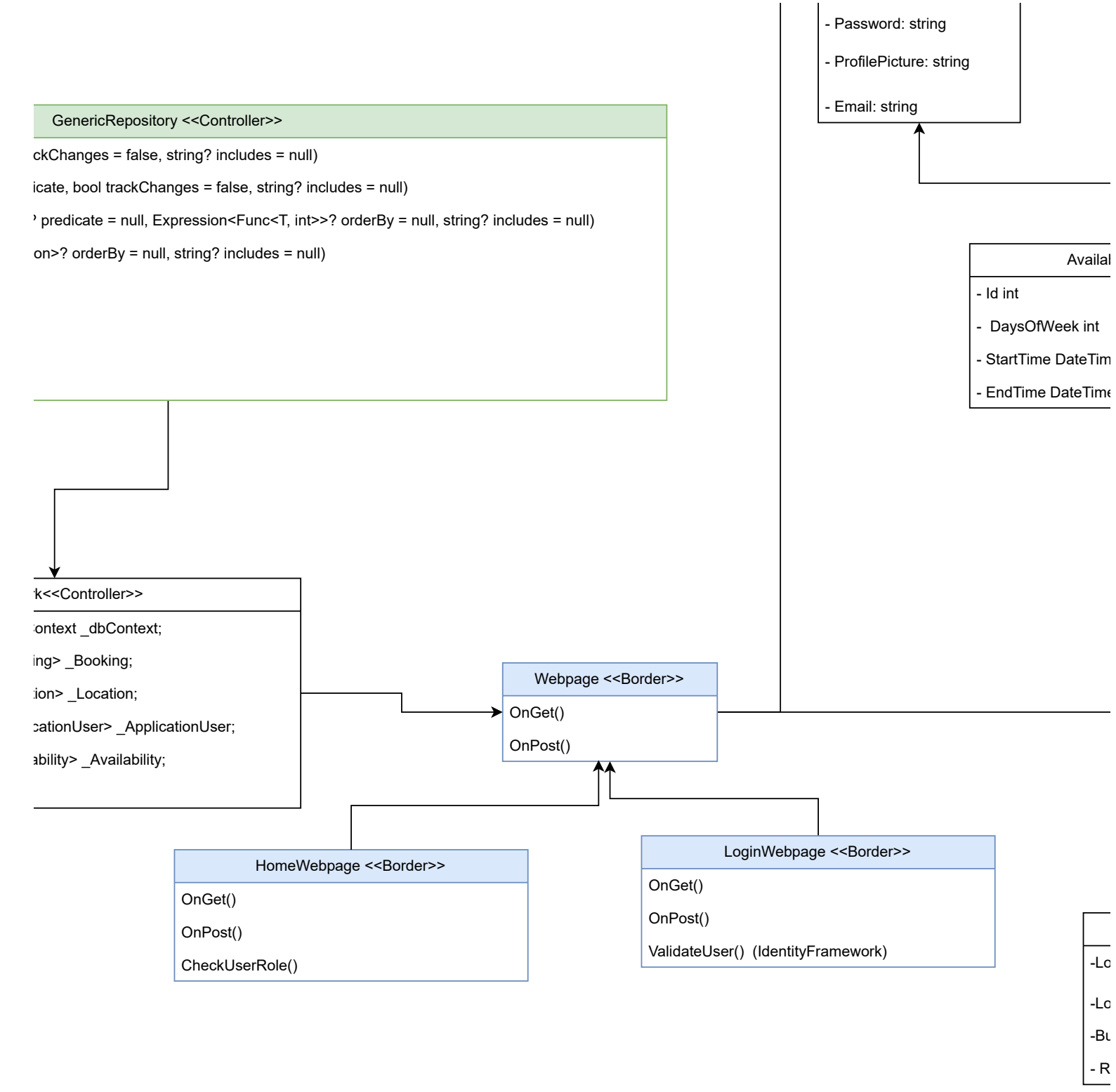

Role (ASP.net)
RoleId
RoleName

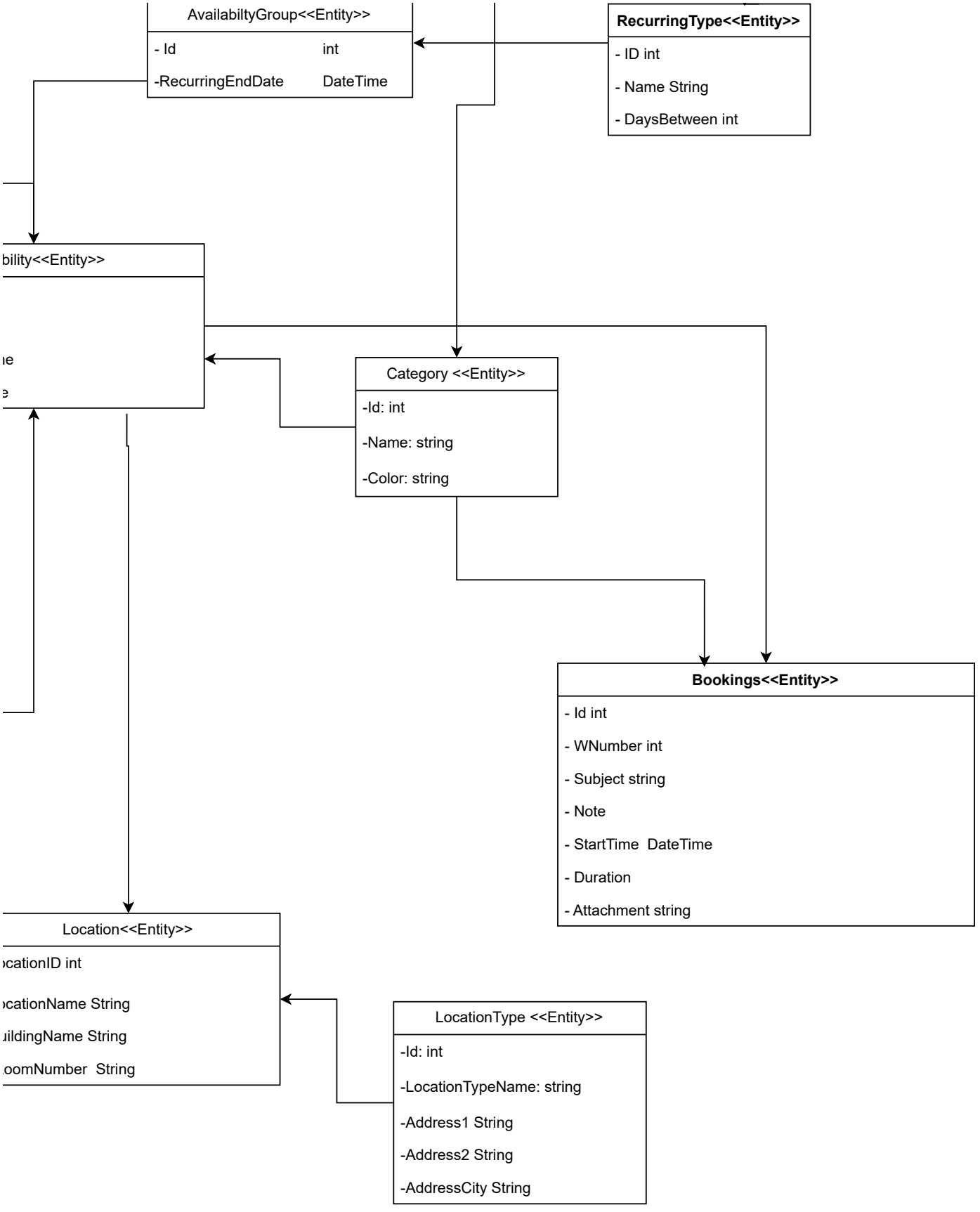

AspNet.Users<<Entity>>
- UserId: int
- FirstName: string
- LastName: string
- PhoneNum: string

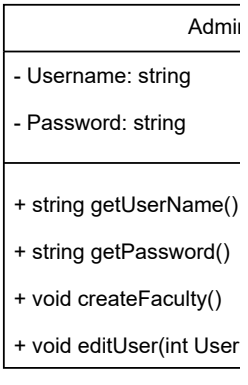

Department <<Entity>>
- Id int
- Name: string
ProviderProfile <<Entity>>
-ProviderProfileID: int
-RemoteLink: string
-BookingPrompt: string
CustomerProfile<<Entity>>
- Id int
-WNumber int

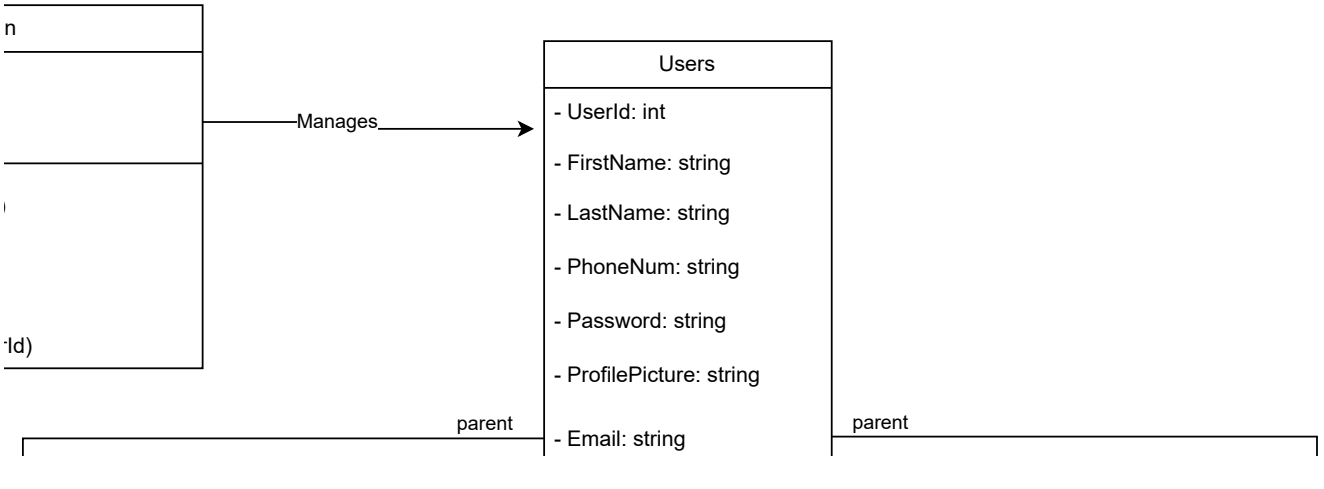

T Get(Expression<Func<T, bool>> predicate, bool trac
Task<T> GetAsync(Expression<Func<T, bool>> predi
IEnumerable<T> GetAll(Expression<Func<T, bool>>?
<Task> GetAllAsync(Expression> predicate, Expressi
void Add(T entity)
void Delete(T entity)
void Delete(IEnumerable entities)
void Update(T entity)

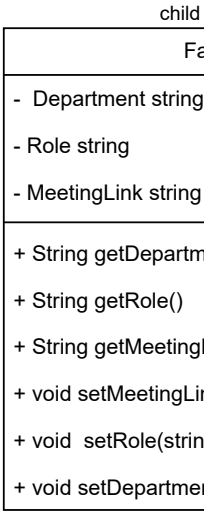

UnitOfWor
private readonly ApplicationDbC
public IGenericRepository<Booki
public IGenericRepository<Locat
public IGenericRepository<Applic
public IGenericRepository<Availa
+ item: attribute

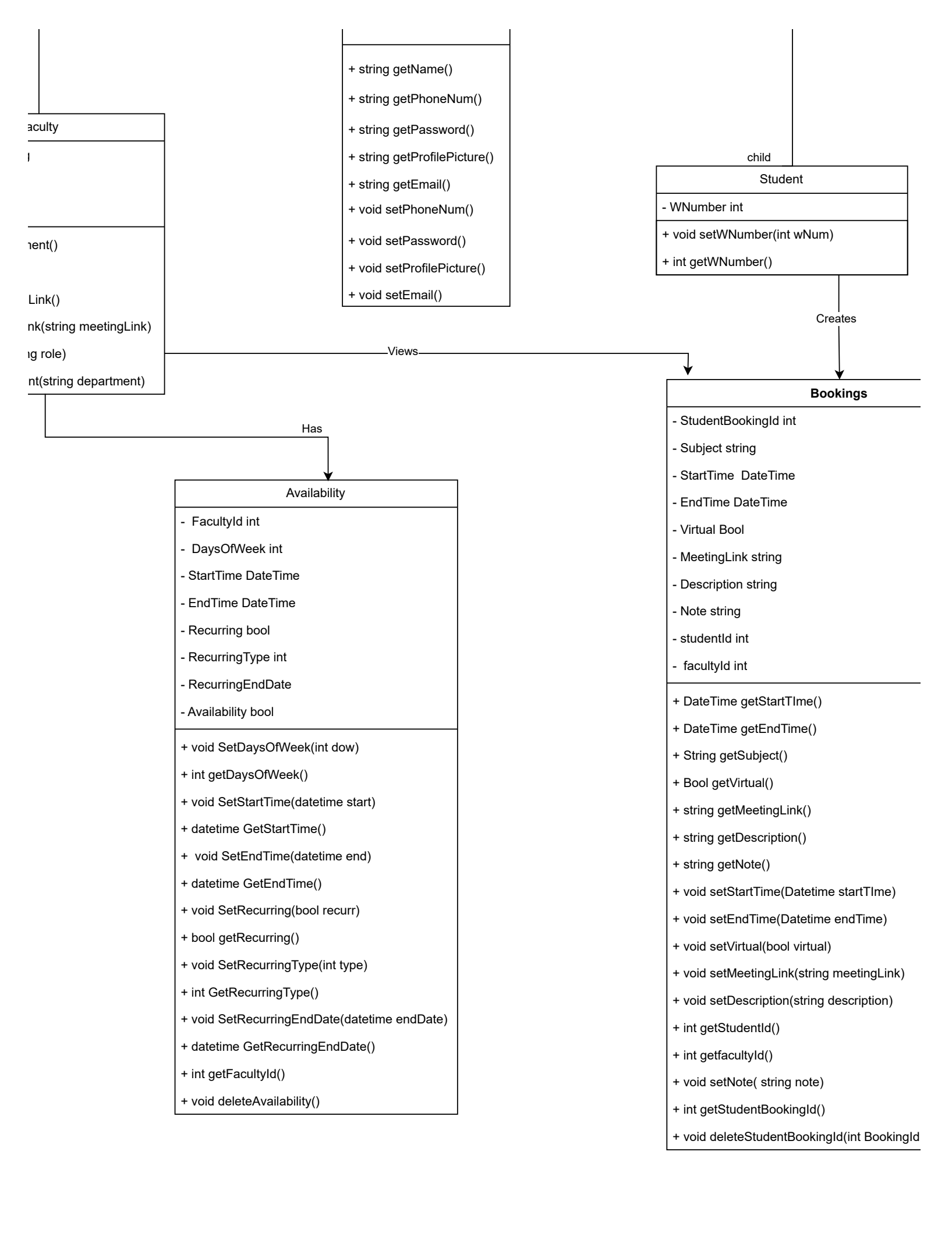

- Password: string
- ProfilePicture: string
- Email: string
GenericRepository <<Controller>>
ckChanges = false, string? includes = null)
icate, bool trackChanges = false, string? includes = null)
predicate = null, Expression<Func<T, int>>? orderBy = null, string? includes = null)
on>? orderBy = null, string? includes = null)
Availal
- Id int
- DaysOfWeek int
- StartTime DateTim
- EndTime DateTime
k<<Controller>>
ontext _dbContext;
ing> _Booking;
ion> _Location;
cationUser> _ApplicationUser;
ability> _Availability;
Webpage <<Border>>
OnGet()
OnPost()
HomeWebpage <<Border>>
OnGet()
OnPost()
CheckUserRole()
LoginWebpage <<Border>>
OnGet()
OnPost()
ValidateUser() (IdentityFramework)
-Lo
-Lo
-Bu
- R

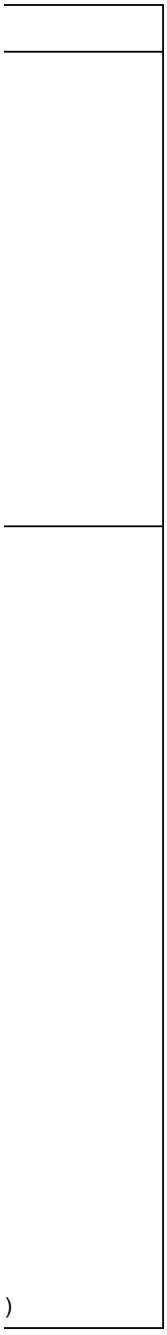

AvailabiltyGroup<<Entity>>
- Id int
-RecurringEndDate DateTime
RecurringType<<Entity>>
- ID int
- Name String
- DaysBetween int
bility<<Entity>>
Category <<Entity>>
-Id: int
-Name: string
-Color: string
Bookings<<Entity>>
- Id int
- WNumber int
- Subject string
- Note
- StartTime DateTime
- Duration
- Attachment string
Location<<Entity>>
cationID int
cationName String
uildingName String
oomNumber String
LocationType <<Entity>>
-Id: int
-LocationTypeName: string
-Address1 String
-Address2 String
-AddressCity String


Admi
- Username: string
- Password: string
+ string getUserName()
+ string getPassword()
+ void createFaculty()
+ void editUser(int User

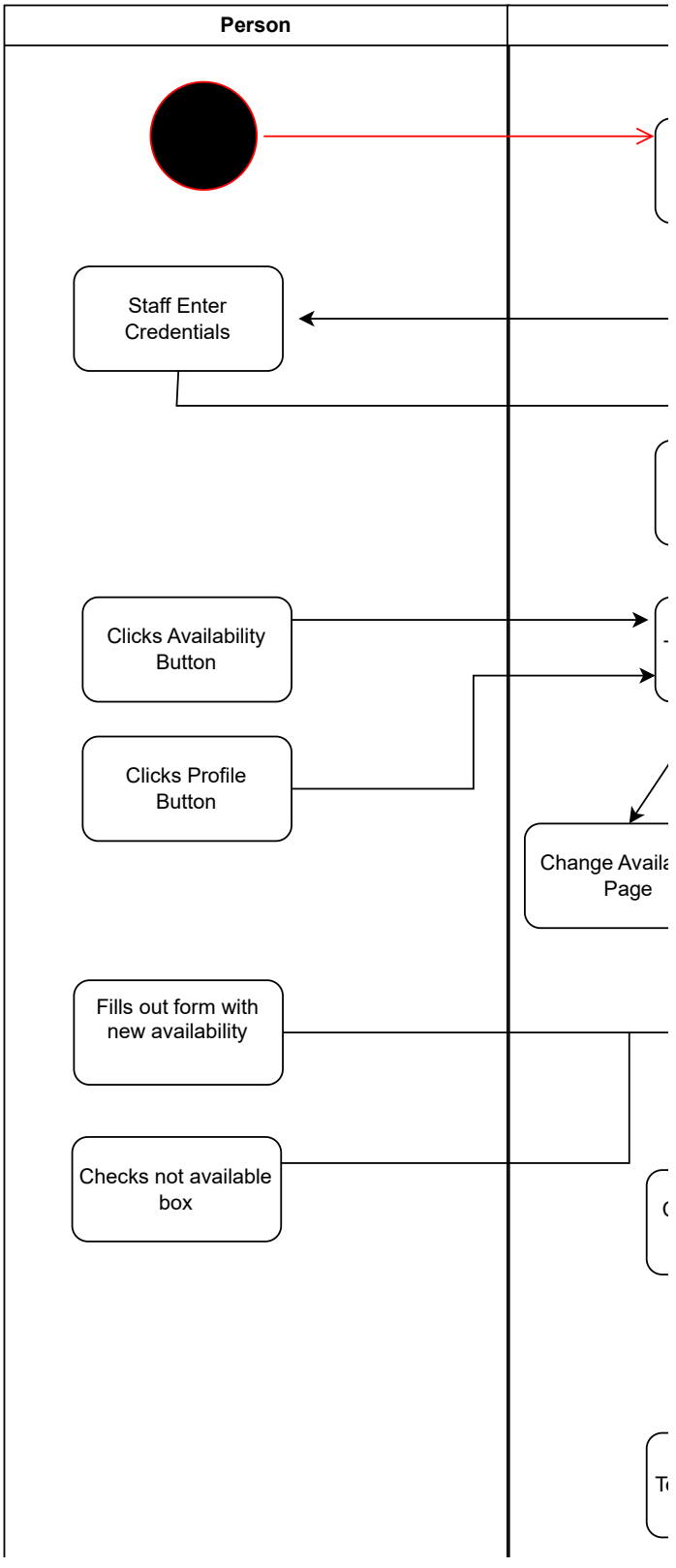

n
Manages
Users
- UserId: int
- FirstName: string
- LastName: string
- PhoneNum: string
- Password: string
- ProfilePicture: string
- Email: string
parent
parent


child
Fa
- Department string
- Role string
- MeetingLink string
+ String getDepartm
+ String getRole()
+ String getMeetingl
+ void setMeetingLi
+ void setRole(strin
+ void setDepartme

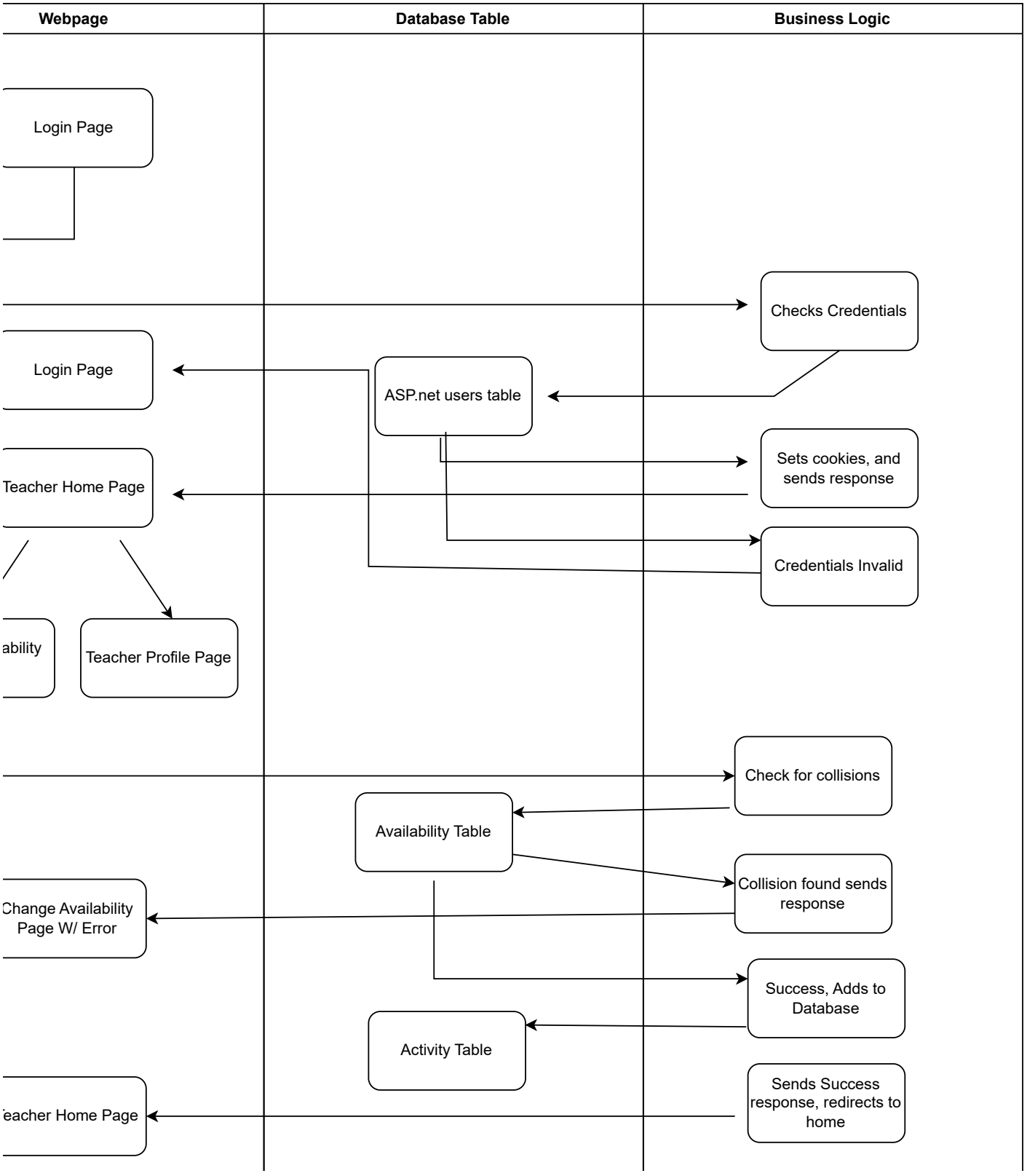

+ string getName()
+ string getPhoneNum()
+ string getPassword()
+ string getProfilePicture()
+ string getEmail()
+ void setPhoneNum()
+ void setPassword()
+ void setProfilePicture()
+ void setEmail()
child
Student
- WNumber int
+ void setWNumber(int wNum)
+ int getWNumber()
Creates
Views
Has
Availability
- FacultyId int
- DaysOfWeek int
- StartTime DateTime
- EndTime DateTime
- Recurring bool
- RecurringType int
- RecurringEndDate
- Availability bool
+ void SetDaysOfWeek(int dow)
+ int getDaysOfWeek()
+ void SetStartTime(datetime start)
+ datetime GetStartTime()
+ void SetEndTime(datetime end)
+ datetime GetEndTime()
+ void SetRecurring(bool recurr)
+ bool getRecurring()
+ void SetRecurringType(int type)
+ int GetRecurringType()
+ void SetRecurringEndDate(datetime endDate)
+ datetime GetRecurringEndDate()
+ int getFacultyId()
+ void deleteAvailability()
Bookings
- StudentBookingId int
- Subject string
- StartTime DateTime
- EndTime DateTime
- Virtual Bool
- MeetingLink string
- Description string
- Note string
- studentId int
- facultyId int
+ DateTime getStartTIme()
+ DateTime getEndTime()
+ String getSubject()
+ Bool getVirtual()
+ string getMeetingLink()
+ string getDescription()
+ string getNote()
+ void setStartTime(Datetime startTIme)
+ void setEndTime(Datetime endTime)
+ void setVirtual(bool virtual)
+ void setMeetingLink(string meetingLink)
+ void setDescription(string description)
+ int getStudentId()
+ int getfacultyId()
+ void setNote(string note)
+ int getStudentBookingId()
+ void deleteStudentBookingId(int BookingId



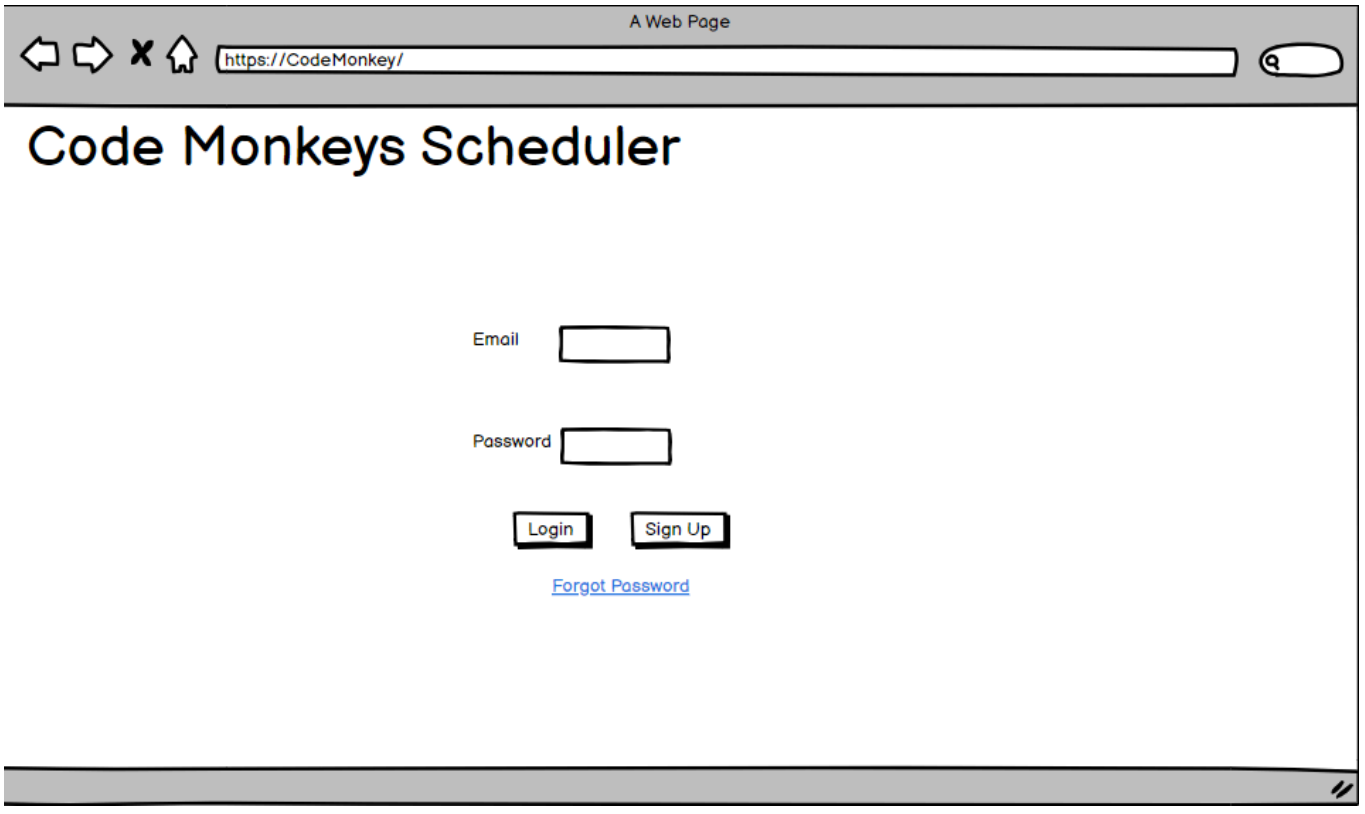
As staff, I
times on t
book app



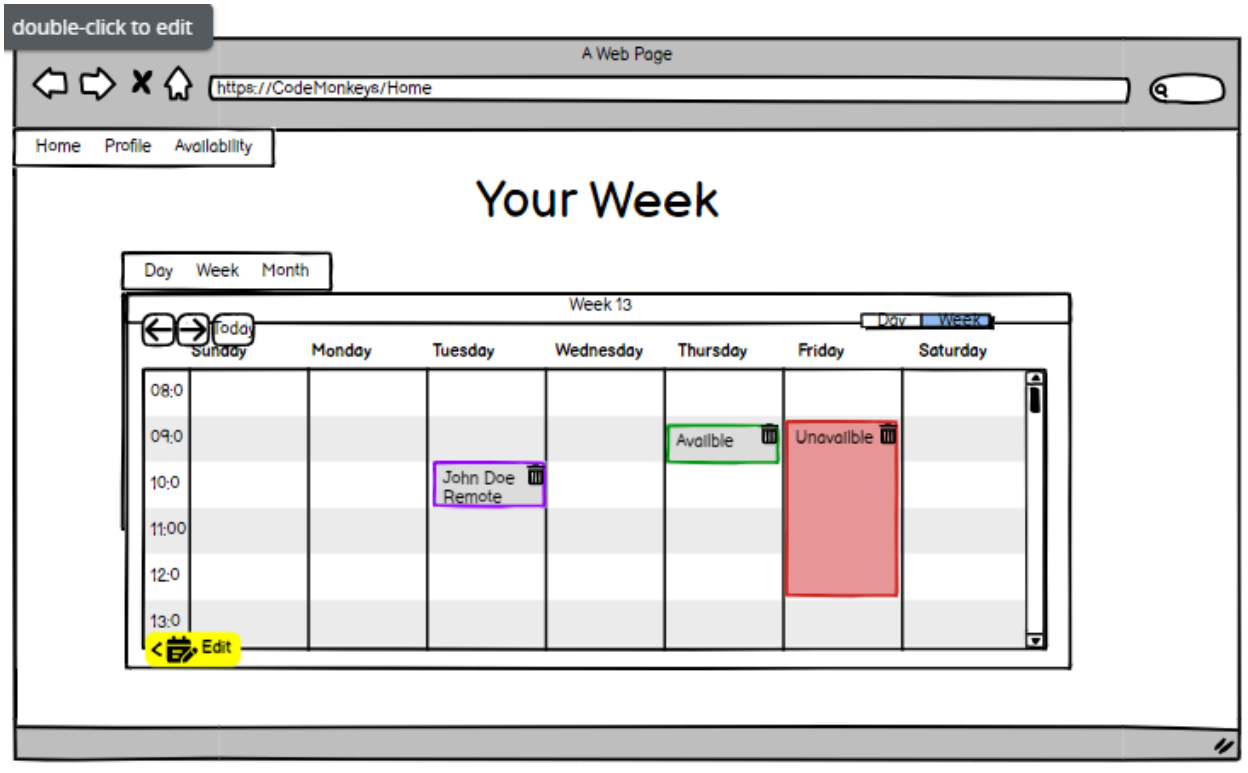
I need to set my availably and non-available the scheduler enabling me to dynamically ointments around my schedule

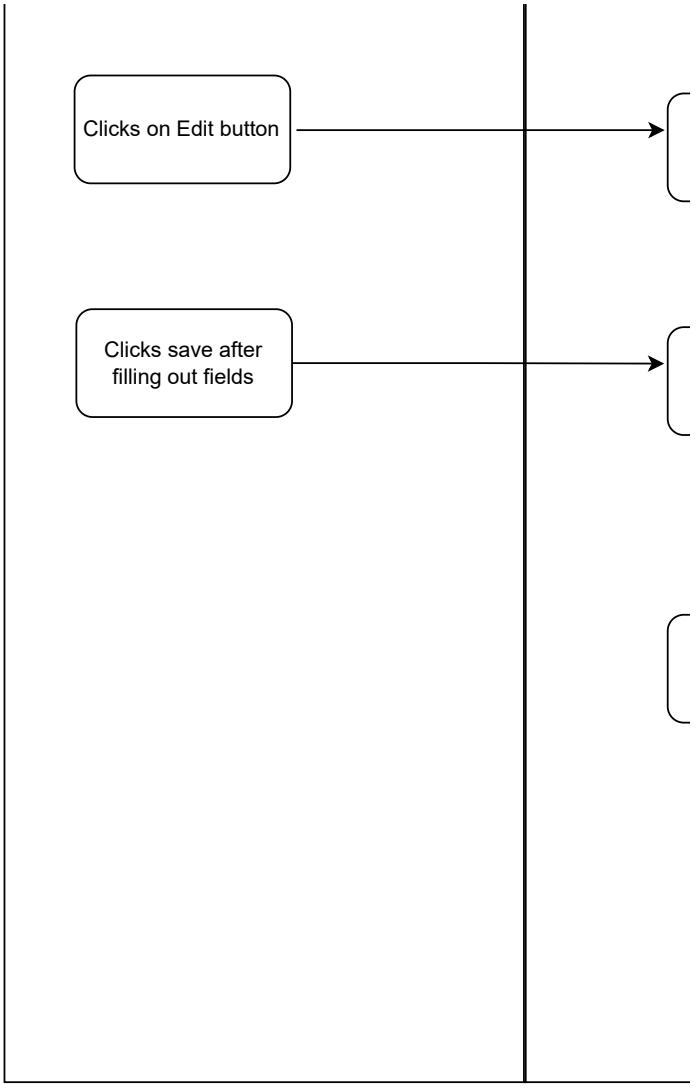


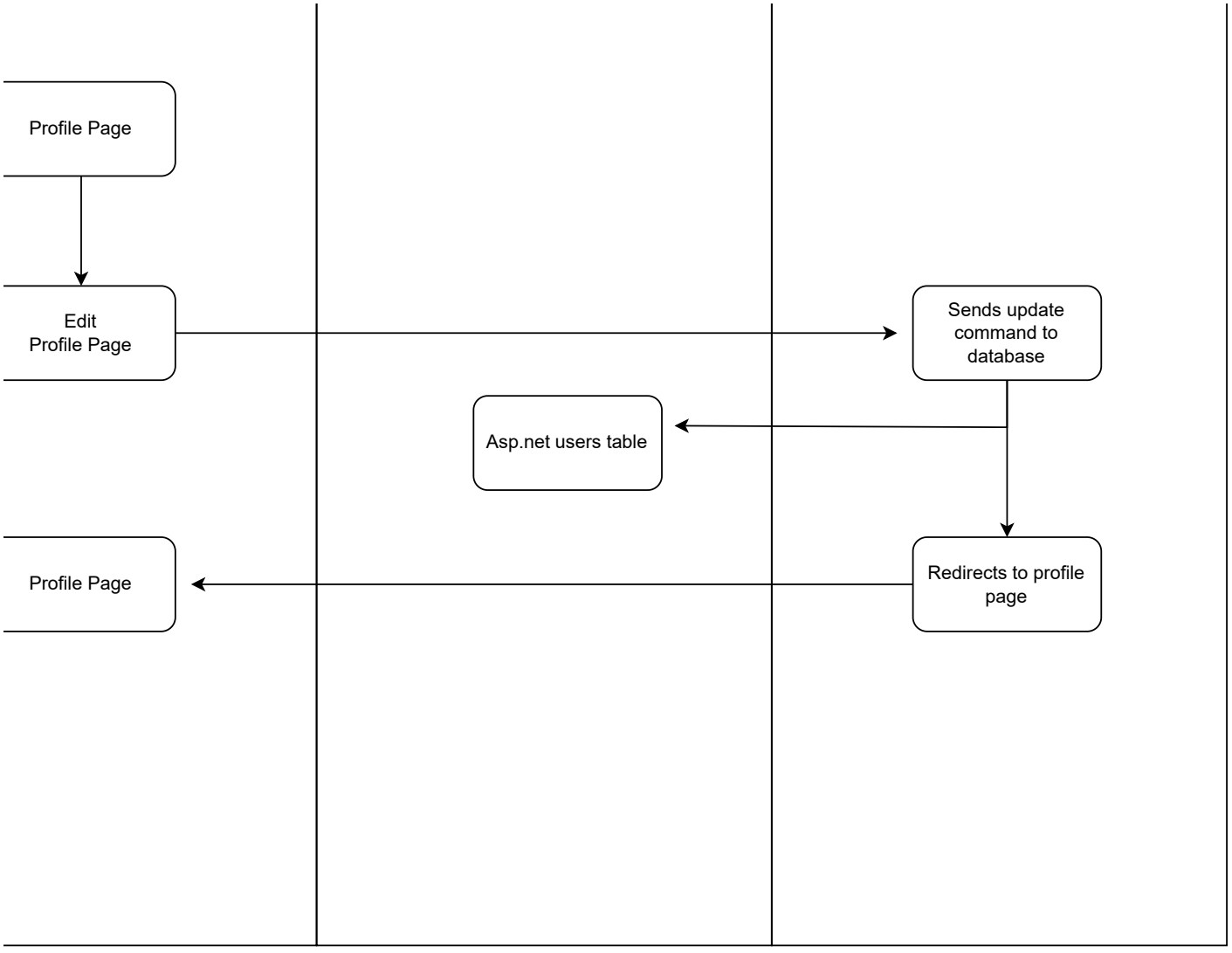
Log In



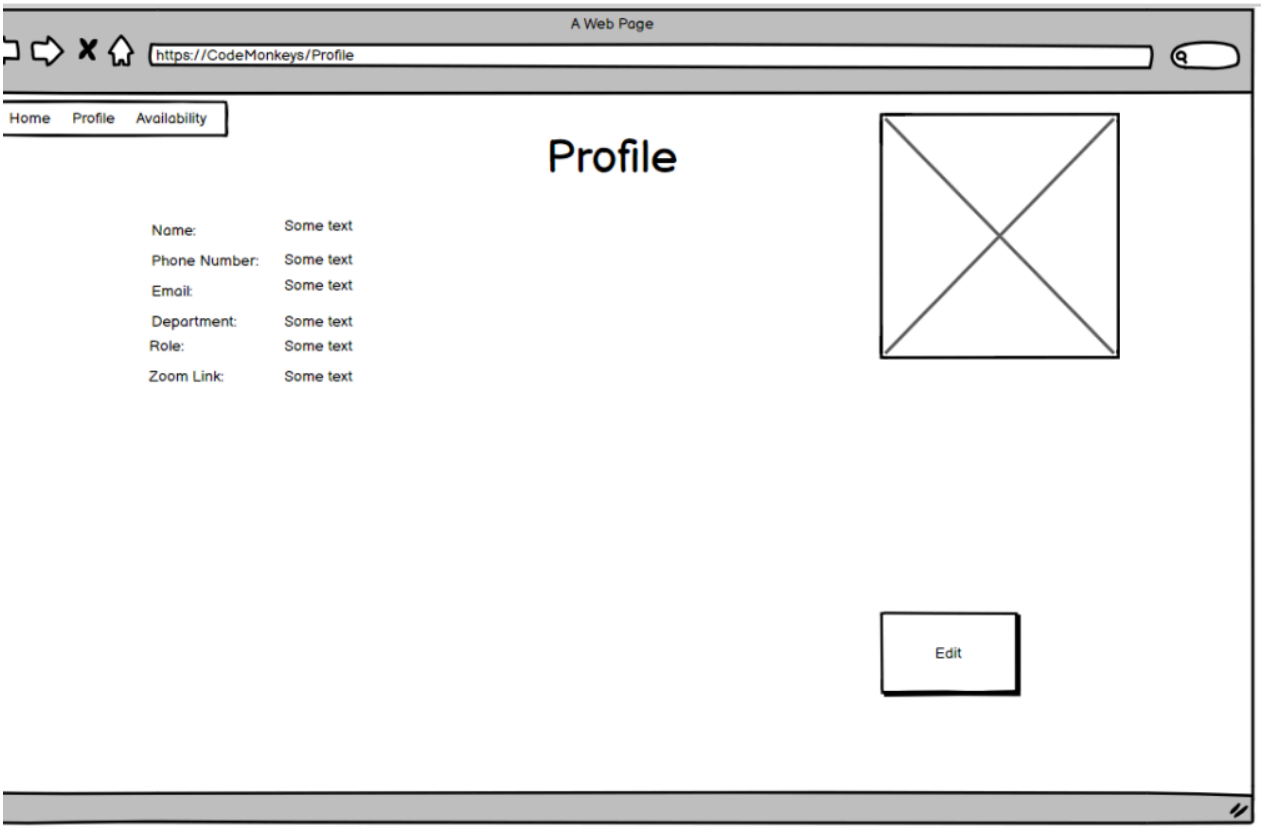
Availability



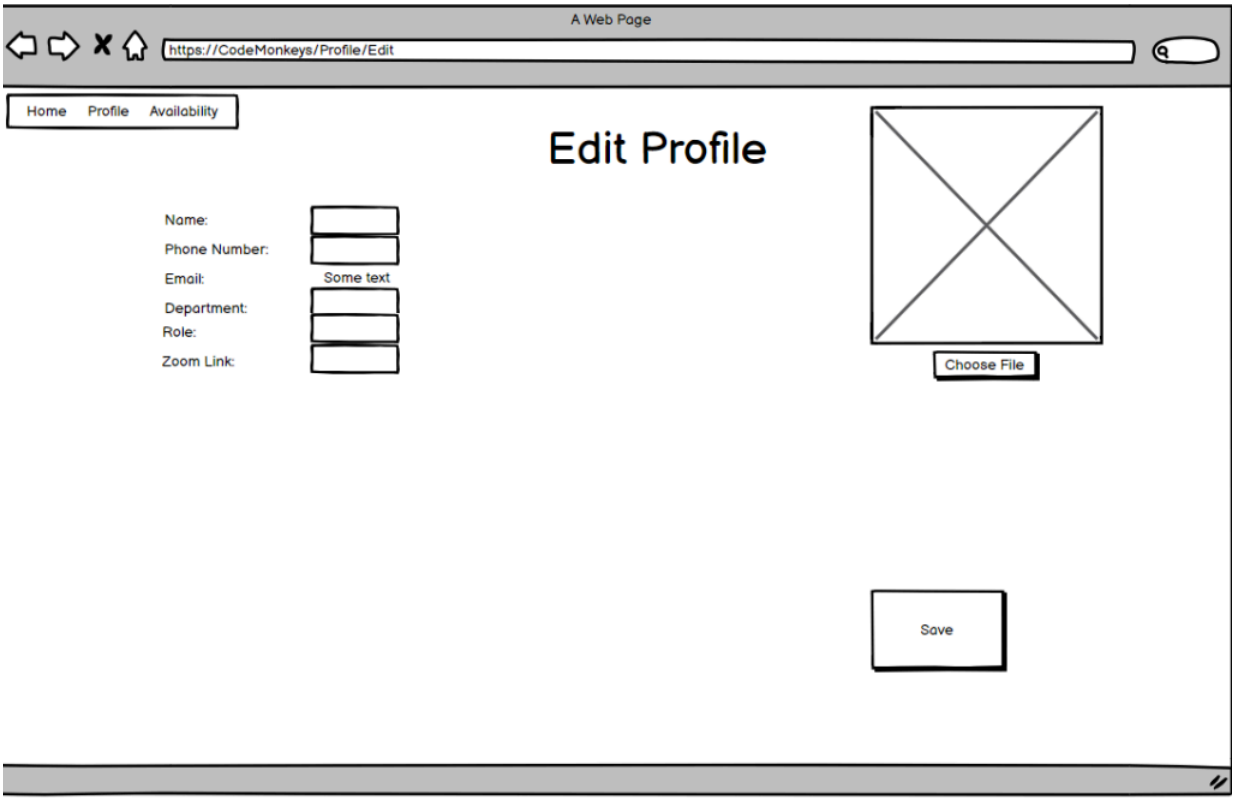

Clicks on Edit button
Clicks save after filling out fields


Profile Page
Edit
Profile Page
Sends update
command to
database
Asp.net users table
Profile Page
Redirects to profile
page

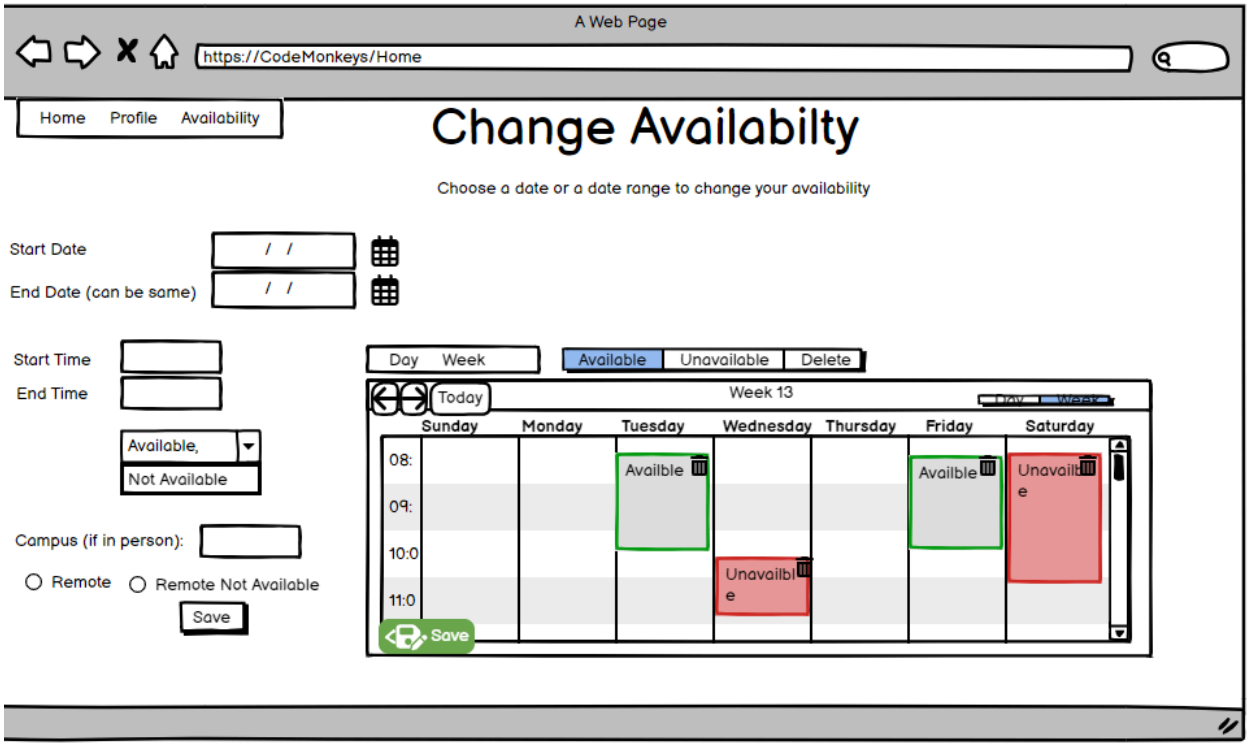
Profile

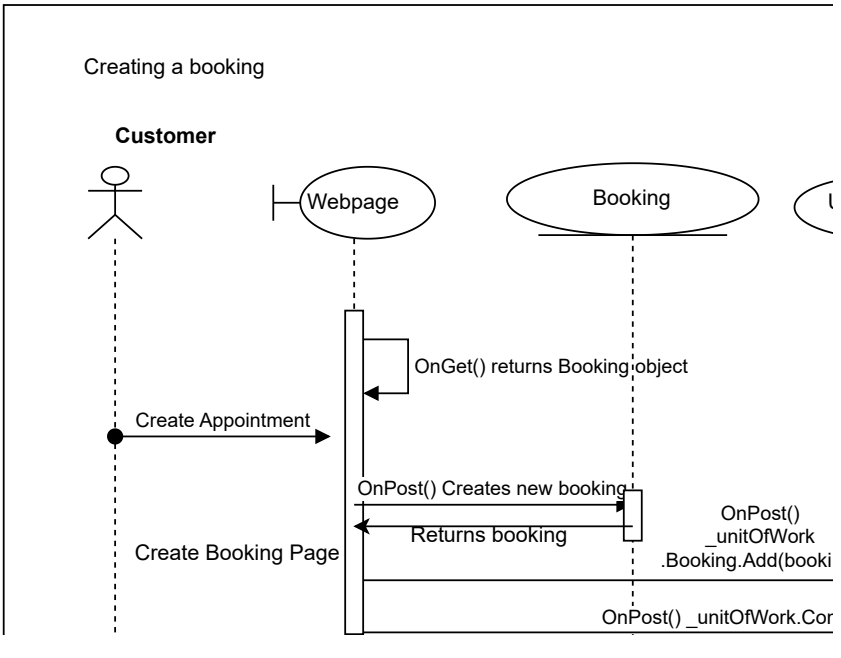


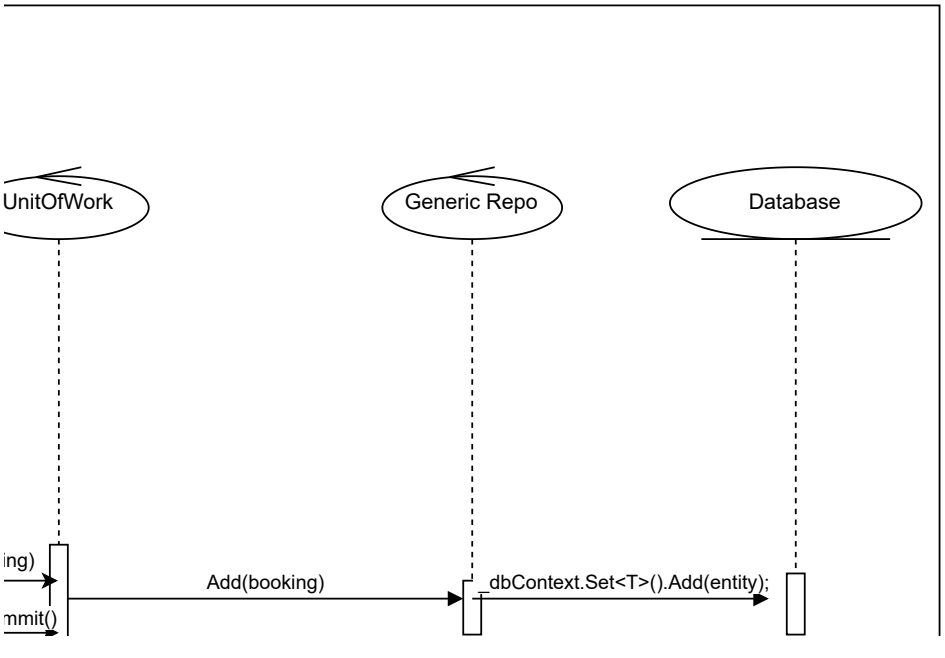
Edit Profile

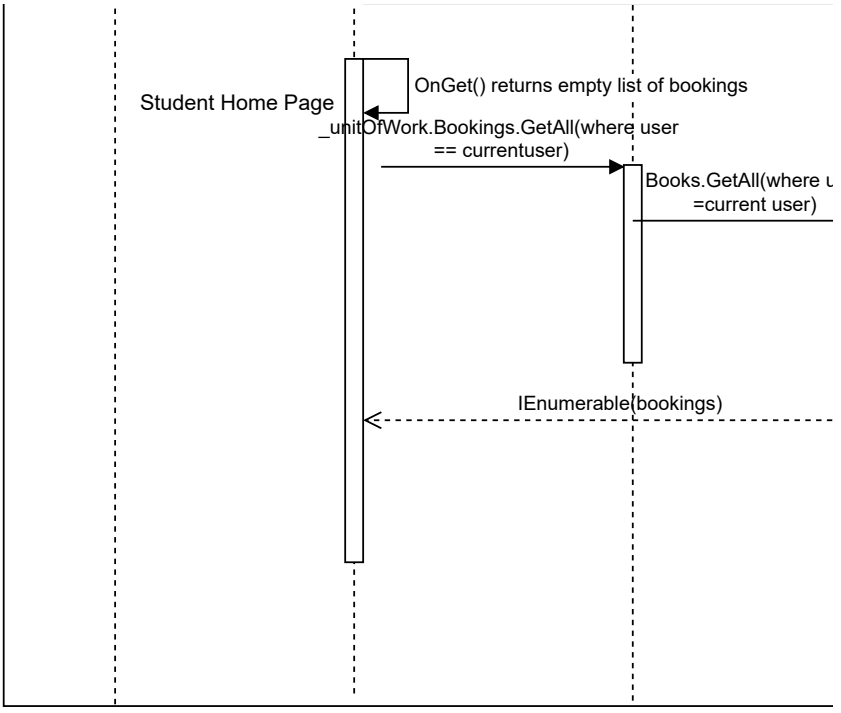


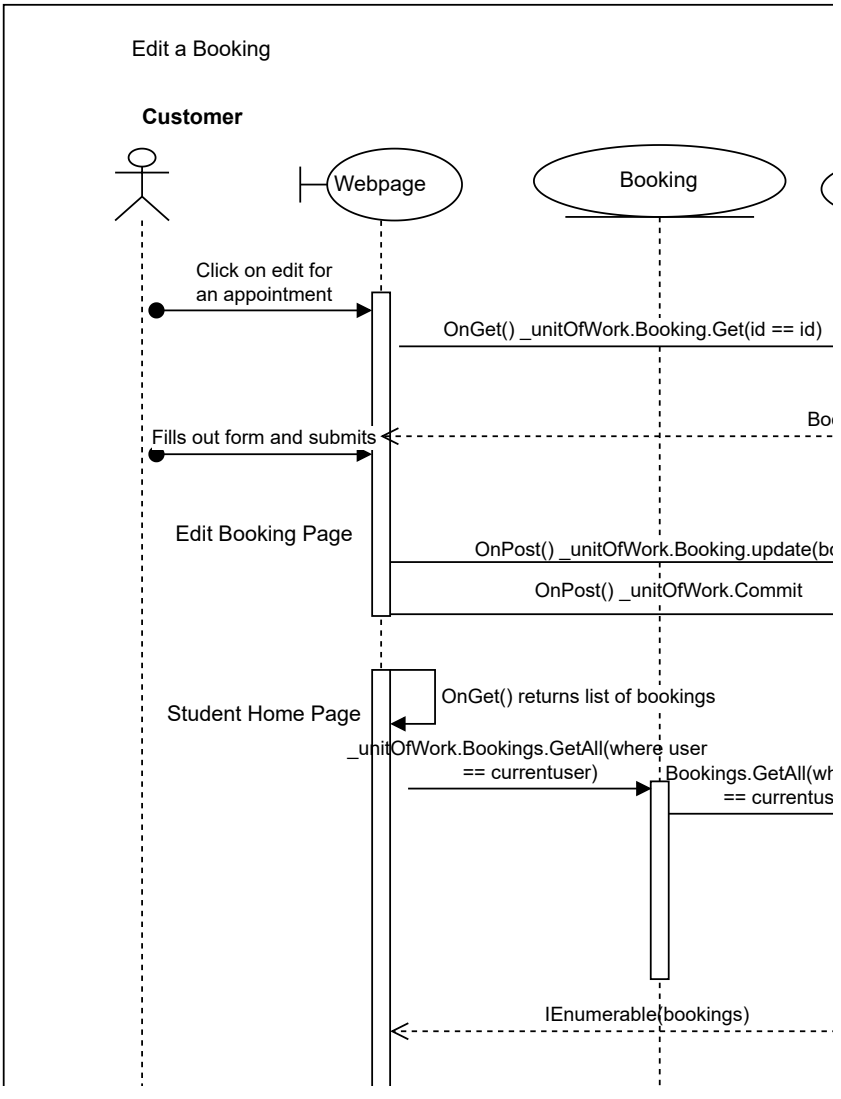
Availability

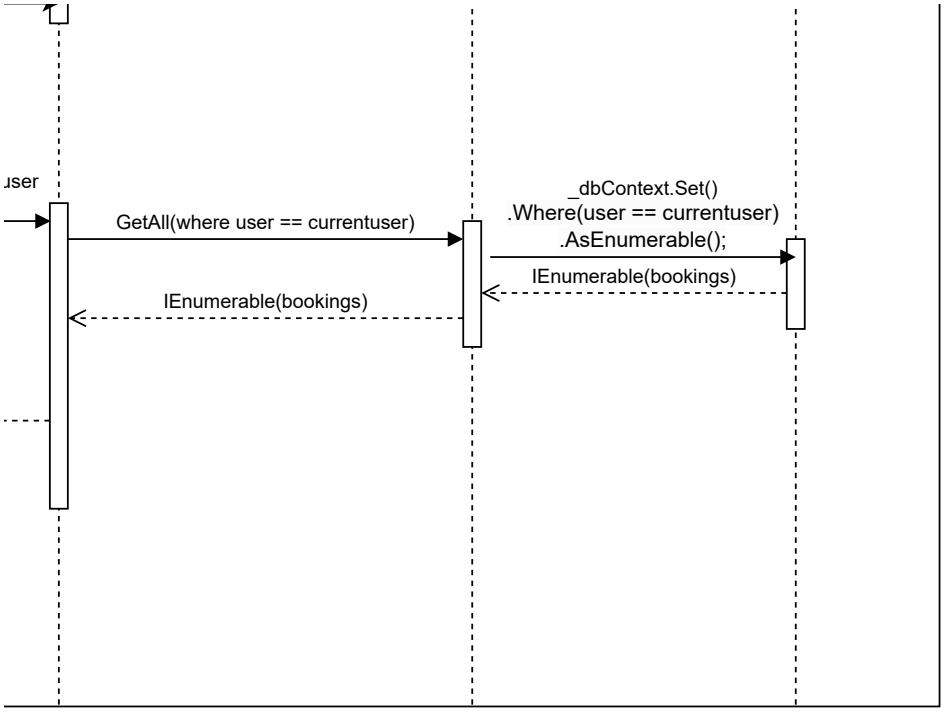


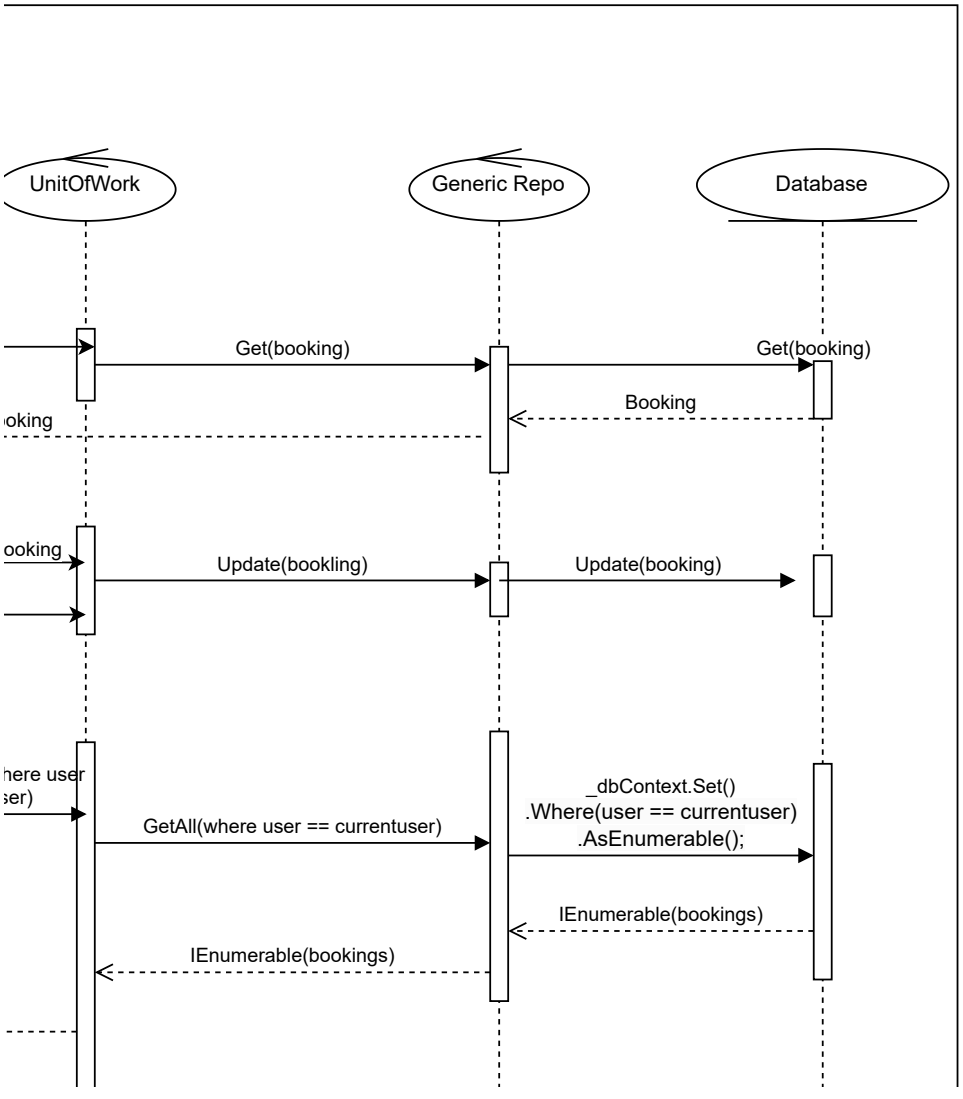

Creating a booking
Customer
Webpage
Booking
OnGet() returns Booking object
Create Appointment
OnPost() Creates new booking
Returns booking
Create Booking Page
OnPost()
_unitOfWork
.Booking.Add(booki
OnPost() _unitOfWork.Cor

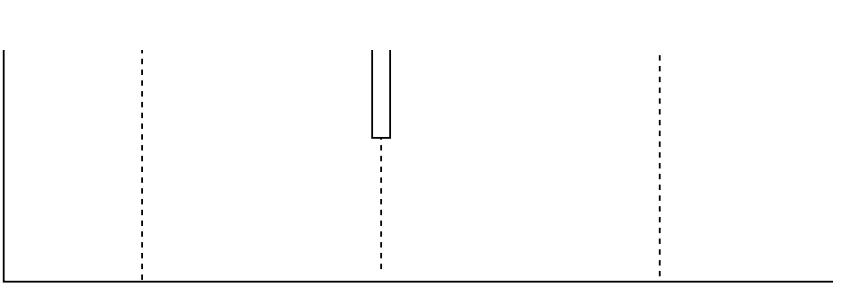

UnitOfWork
Generic Repo
Database
ing)
mmit()
Add(booking)
_dbContext.Set<T>().Add(entity);

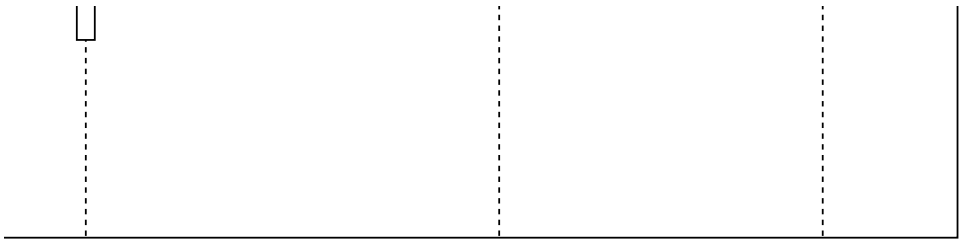

Student Home Page
OnGet() returns empty list of bookings
_unitOfWork.Bookings.GetAll(where user
== currentuser)
Books.GetAll(where u
=current user)
IEnumerable(bookings)

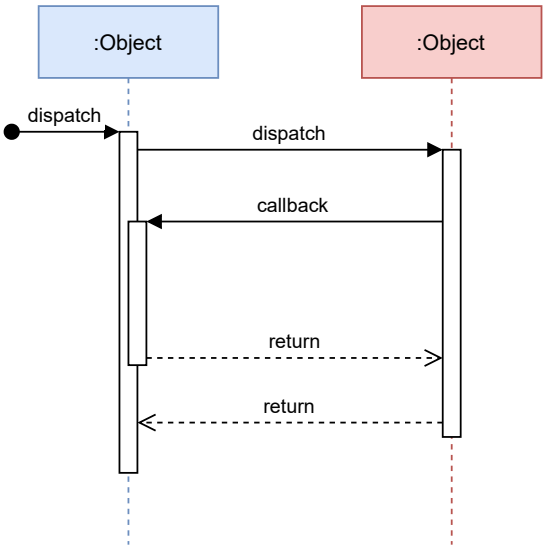

Edit a Booking
Customer
Webpage
Booking
Click on edit for
an appointment
OnGet() _unitOfWork.Booking.Get(id == id)
Fills out form and submits
Edit Booking Page
OnPost() _unitOfWork.Booking.update(b
OnPost() _unitOfWork.Commit
Student Home Page
OnGet() returns list of bookings
_unitOfWork.Bookings.GetAll(where user
== currentuser)
Bookings.GetAll(wh
== currentus
IEnumerable(bookings)

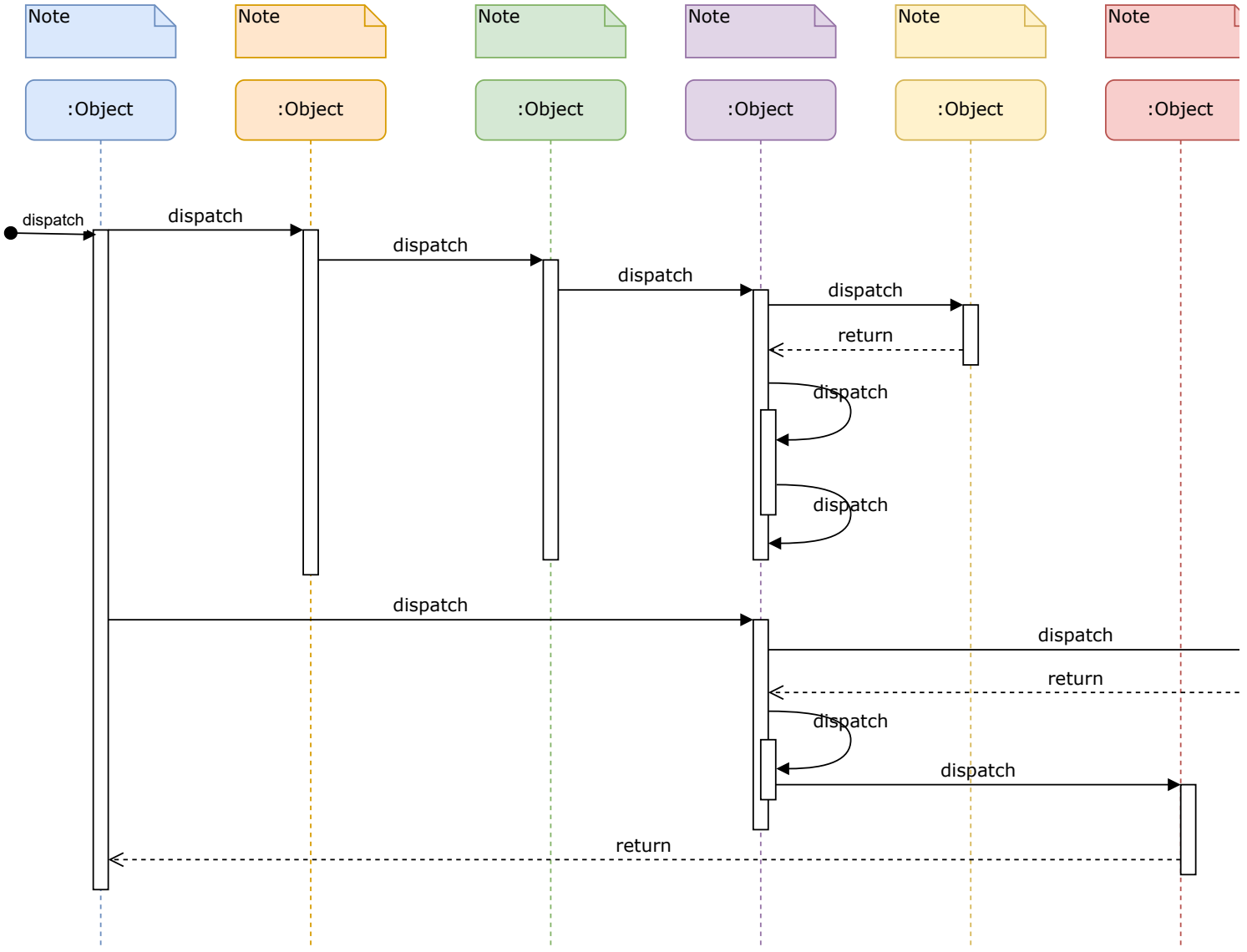

GetAll(where user == currentuser)
_dbContext.Set()
.Where(user == currentuser)
.AsEnumerable();
IEnumerable(bookings)
IEnumerable(bookings)

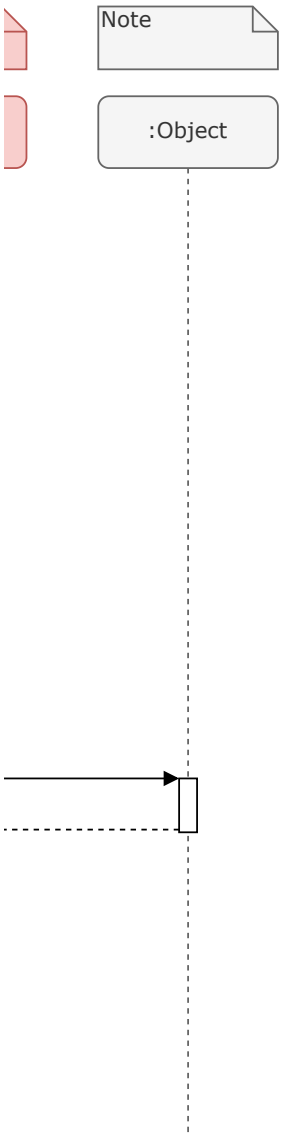

UnitOfWork
Generic Repo
Database
Get(booking)
Get(booking)
Booking
Update(bookling)
Update(booking)
GetAll(where user == currentuser)
_dbContext.Set()
.Where(user == currentuser)
.AsEnumerable();
IEnumerable(bookings)
IEnumerable(bookings)

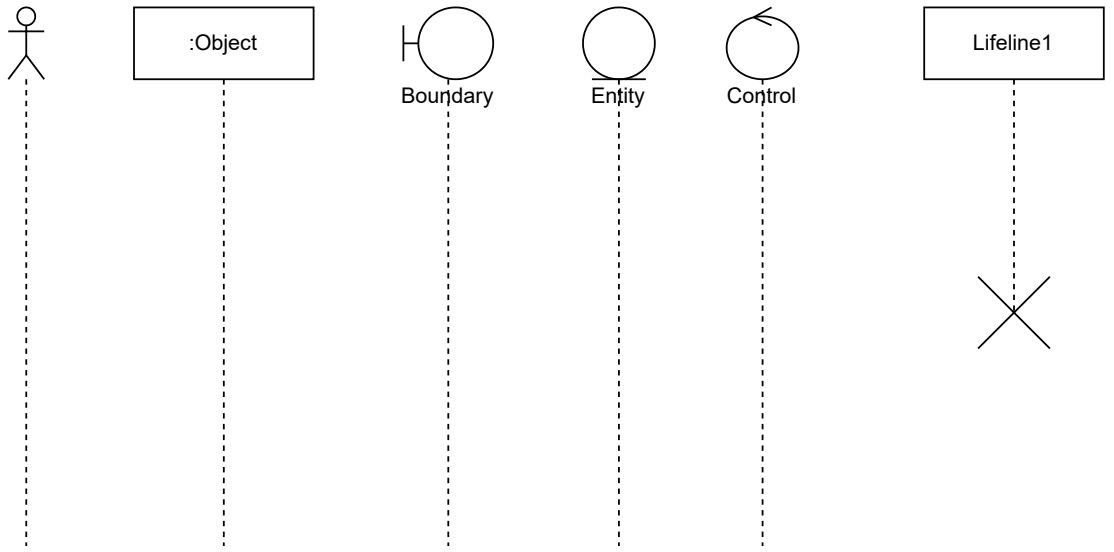


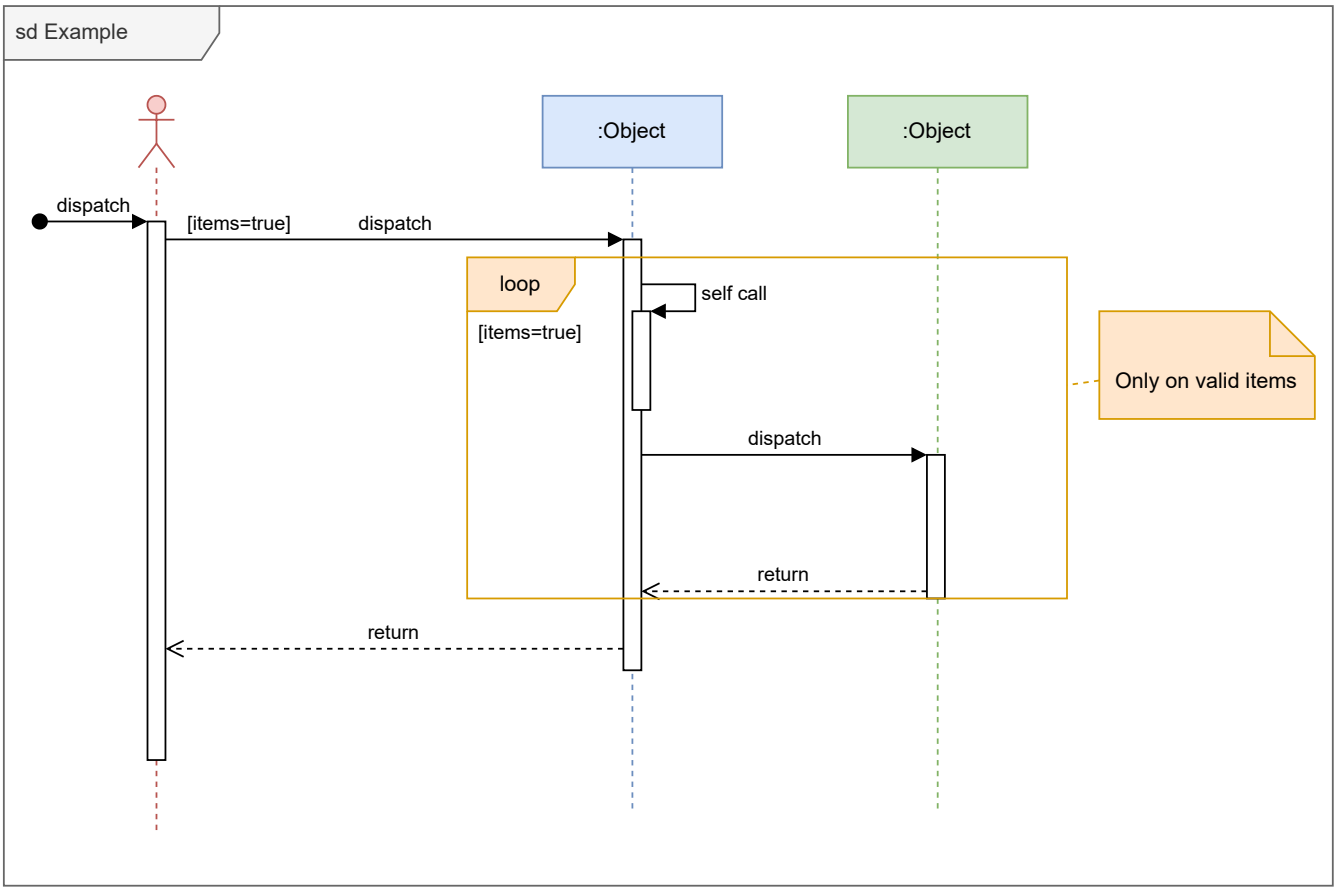


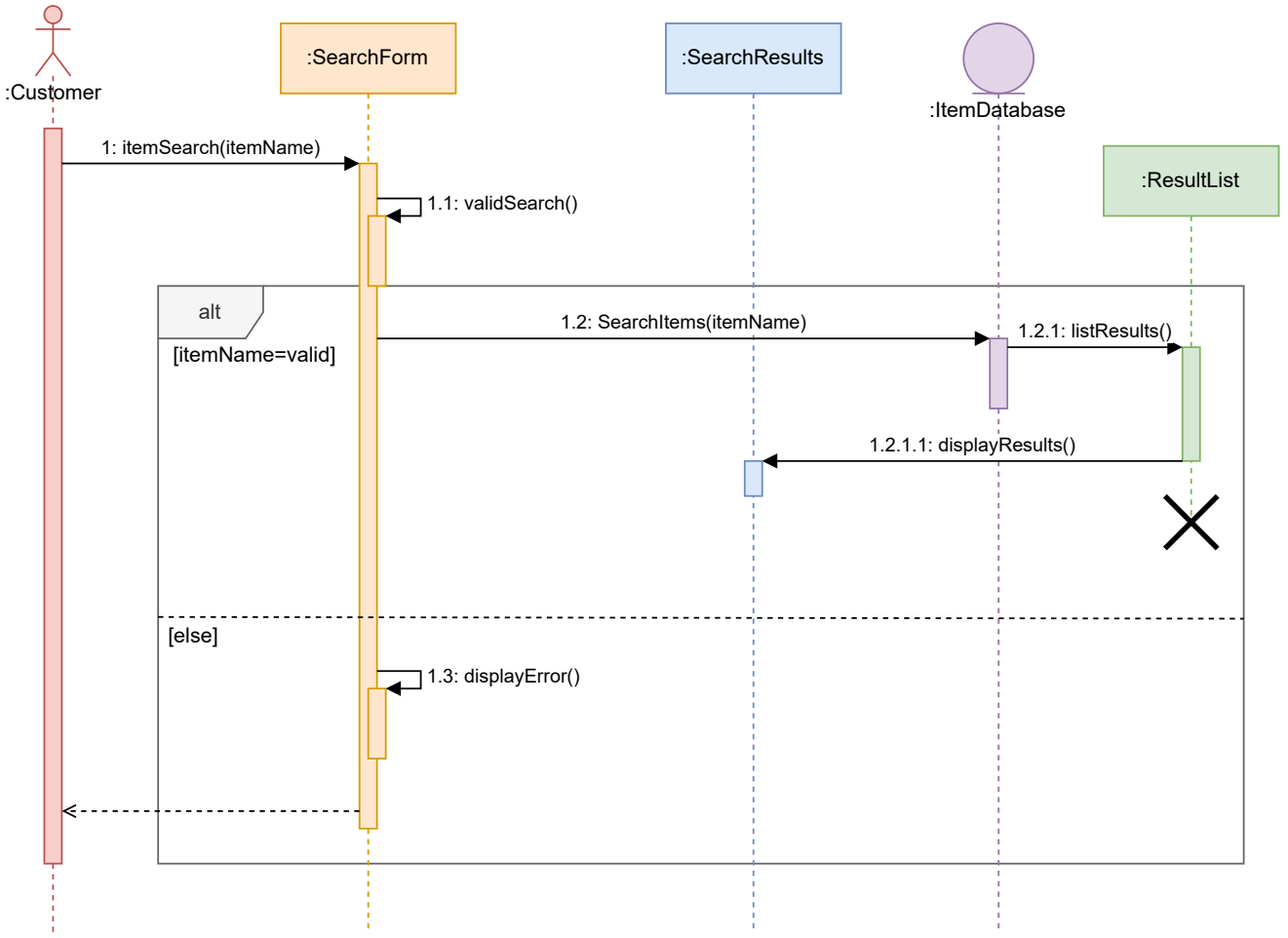

:Object
:Object
dispatch
dispatch
callback
return
return


Note
Note
Note
Note
Note
Note
:Object
:Object
:Object
:Object
:Object
:Object
dispatch
dispatch
dispatch
dispatch
dispatch
return
dispatch
dispatch
dispatch
dispatch
return
dispatch
dispatch
return


Note
:Object


:Object
Boundary
Entity
Control
Lifeline1


sd Example
:Object
:Object
dispatch
[items=true]
dispatch
loop
[items=true]
self call
Only on valid items
dispatch
return
return


:Customer
:SearchForm
:SearchResults
:ItemDatabase
:ResultList
1: itemSearch(itemName)
1.1: validSearch()
alt
[itemName=valid]
1.2: SearchItems(itemName)
1.2.1: listResults()
1.2.1.1: displayResults()
[else]
1.3: displayError()