UM Low Level:BitOHealth

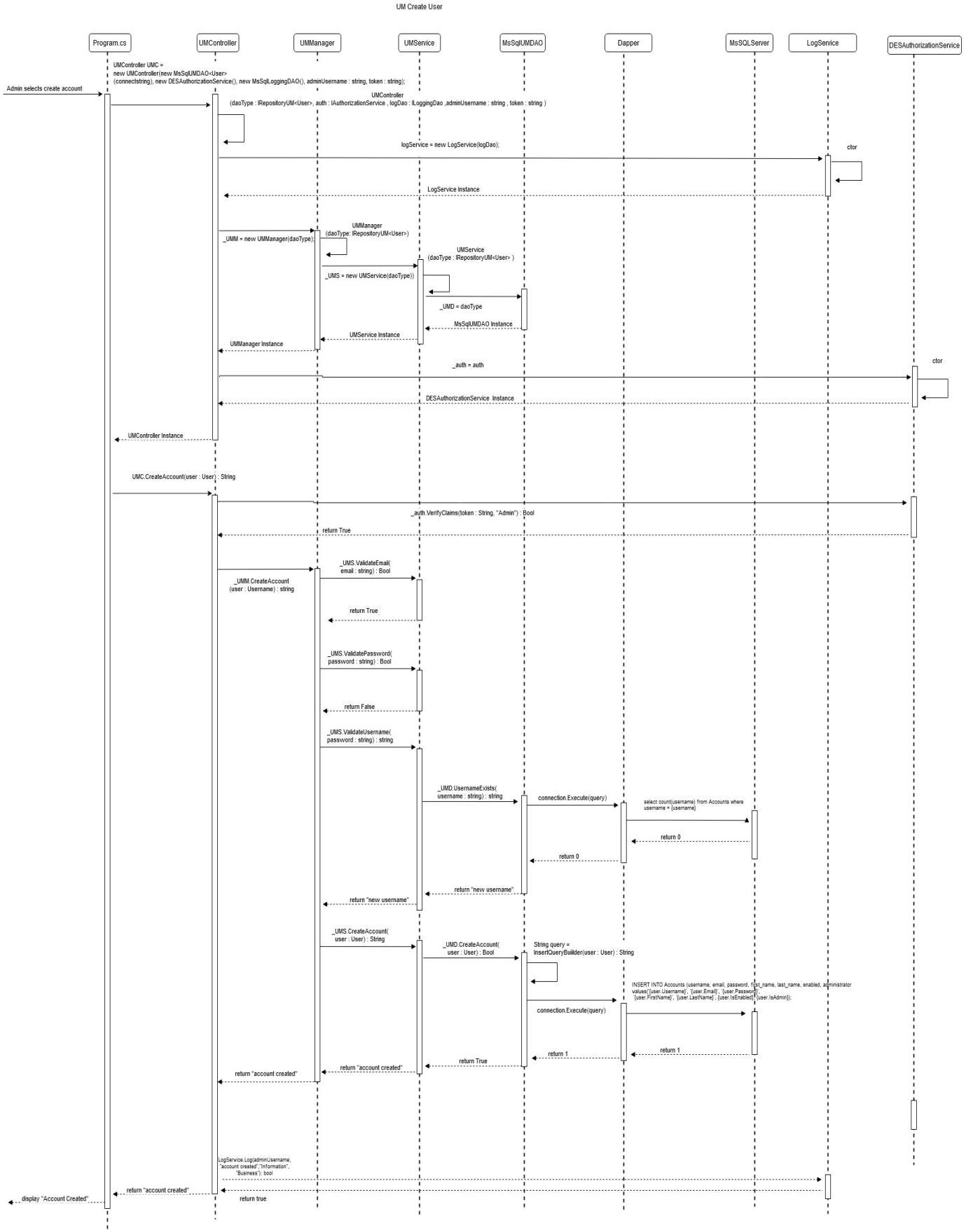
The 6 Bits:
Angel Cueva
Dante Secundino
Emily Sahyoun
Jacob Munoz
Rami Iskender
Sebastian Vasquez (Team Leader)
12/15/2021

1 Create User

Valid Inputs:
Password:
Password must be a minimum of 8 characters and a maximum of 30 characters, Password must contain uppercase, lowercase, number, and special character
Password must consist of only valid characters: a-z, A-Z, 0-9, .,@!
Username:
Username must be 7-15 character and length and only contain a-z,A-Z,0-9, .,@!
Username must not be in database
Email
Email must follow the proper email format.-name@domain Email may be maximum of 255 characters.

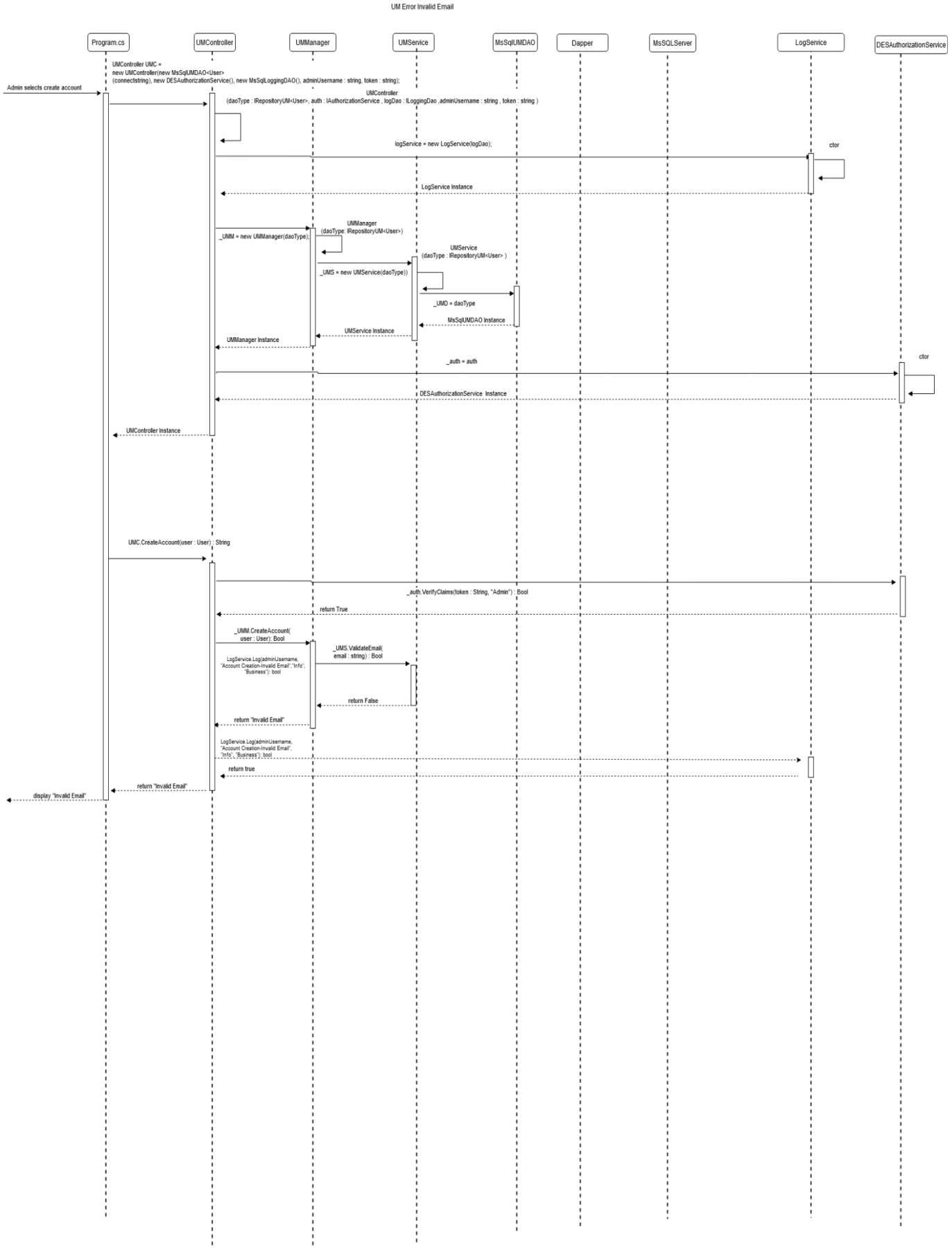
1.1 Successful User Creation

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Inputs are valid.


UM Create User
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects create account
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.CreateAccount(user : User) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMS.ValidateEmail(
email : string) : Bool
_UMM.CreateAccount
(user : Username) : string
return True
_UMS.ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
_UMD.UsernameExists(
username : string) : string
connection.Execute(query)
select count(username) from Accounts where
username = {username}
return 0
return 0
return "new username"
return "new username"
_UMS.CreateAccount(
user : User) : String
_UMD.CreateAccount(
user : User) : Bool
String query =
InsertQueryBuilder(user : User) : String
INSERT INTO Accounts (username, email, password, first_name, last_name, enabled, administrator
values('{user.Username}', '{user.Email}', '{user.Password}',
'{user.FirstName}', '{user.LastName}', {user.IsEnabled}, {user.IsAdmin});
connection.Execute(query)
return 1
return 1
return True
return "account created"
return "account created"
LogService.Log(adminUsername,
"account created","Information",
"Business"): bool
return "account created"
return true
display "Account Created"

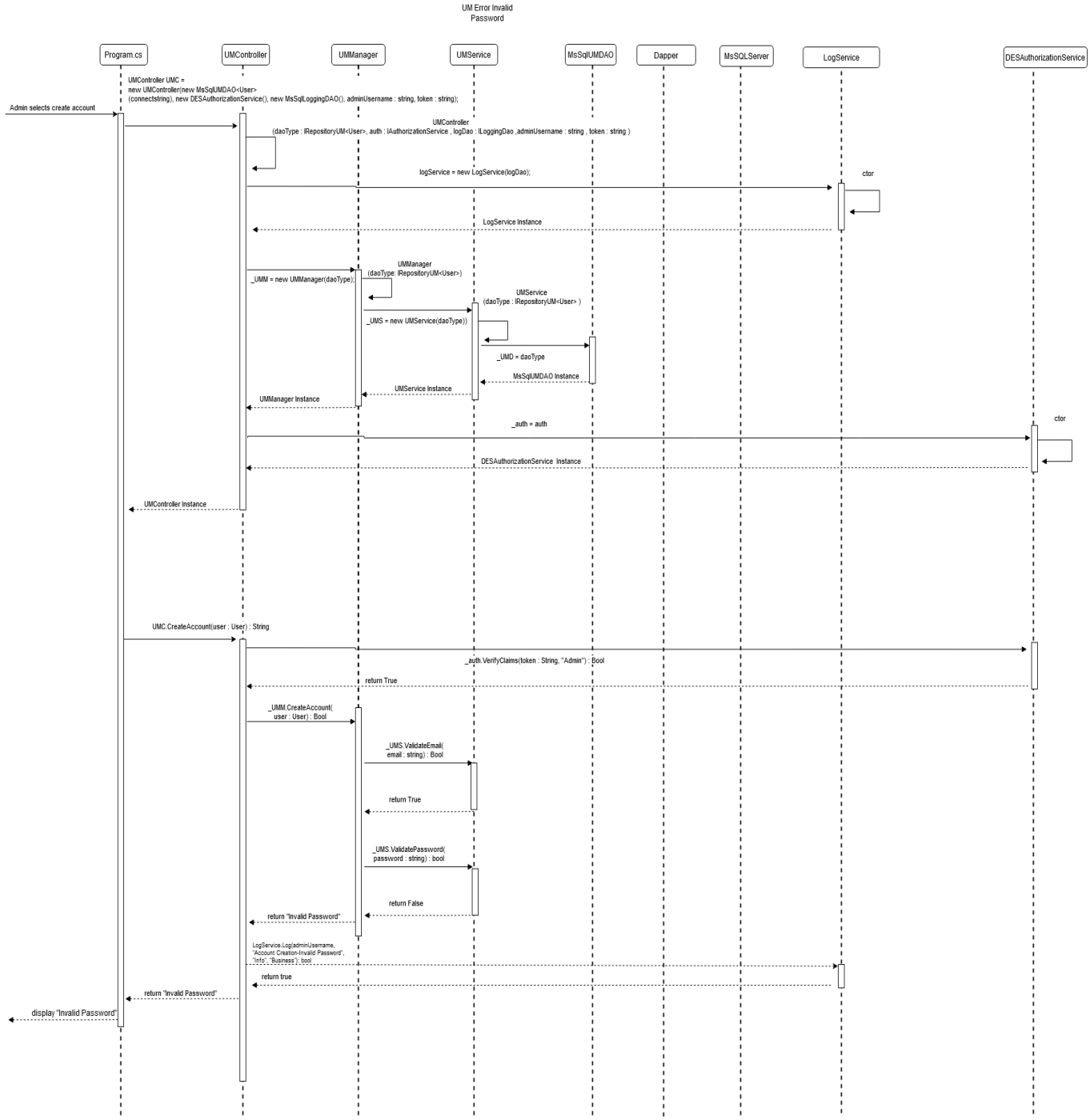
1.2 Create Account-Invalid Email

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Email Input is Invalid


UM Error Invalid Email
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects create account
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.CreateAccount(user : User) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMM.CreateAccount(
user : User): Bool
_UMS.ValidateEmail(
email : string) : Bool
LogService.Log(adminUsername,
"Account Creation-Invalid Email","Info",
"Business"): bool
return False
return "Invalid Email"
LogService.Log(adminUsername,
"Account Creation-Invalid Email",
"Info", "Business"): bool
return true
return "Invalid Email"
display "Invalid Email"

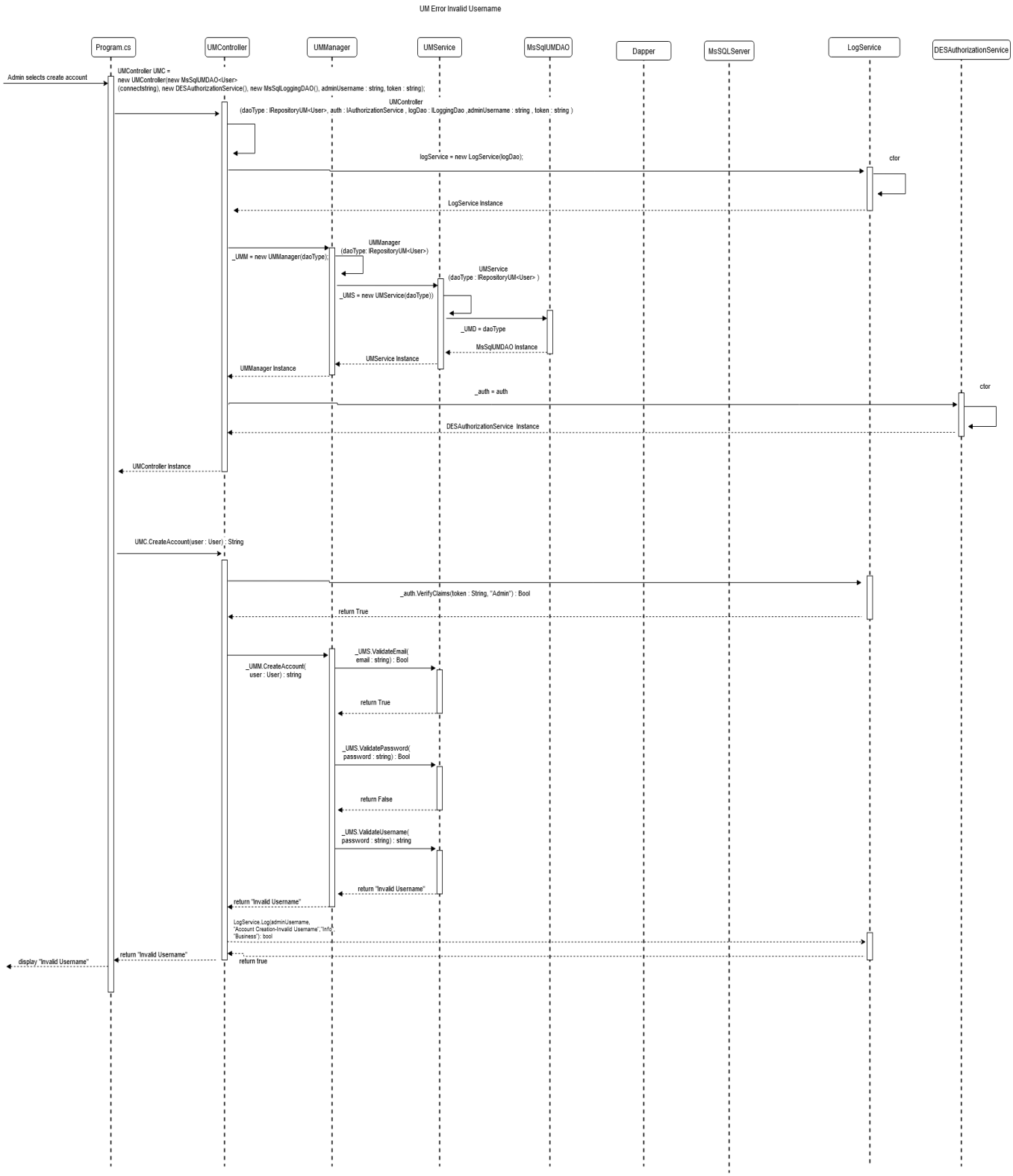
1.3 Create Account-Invalid Password

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Password Input is Invalid



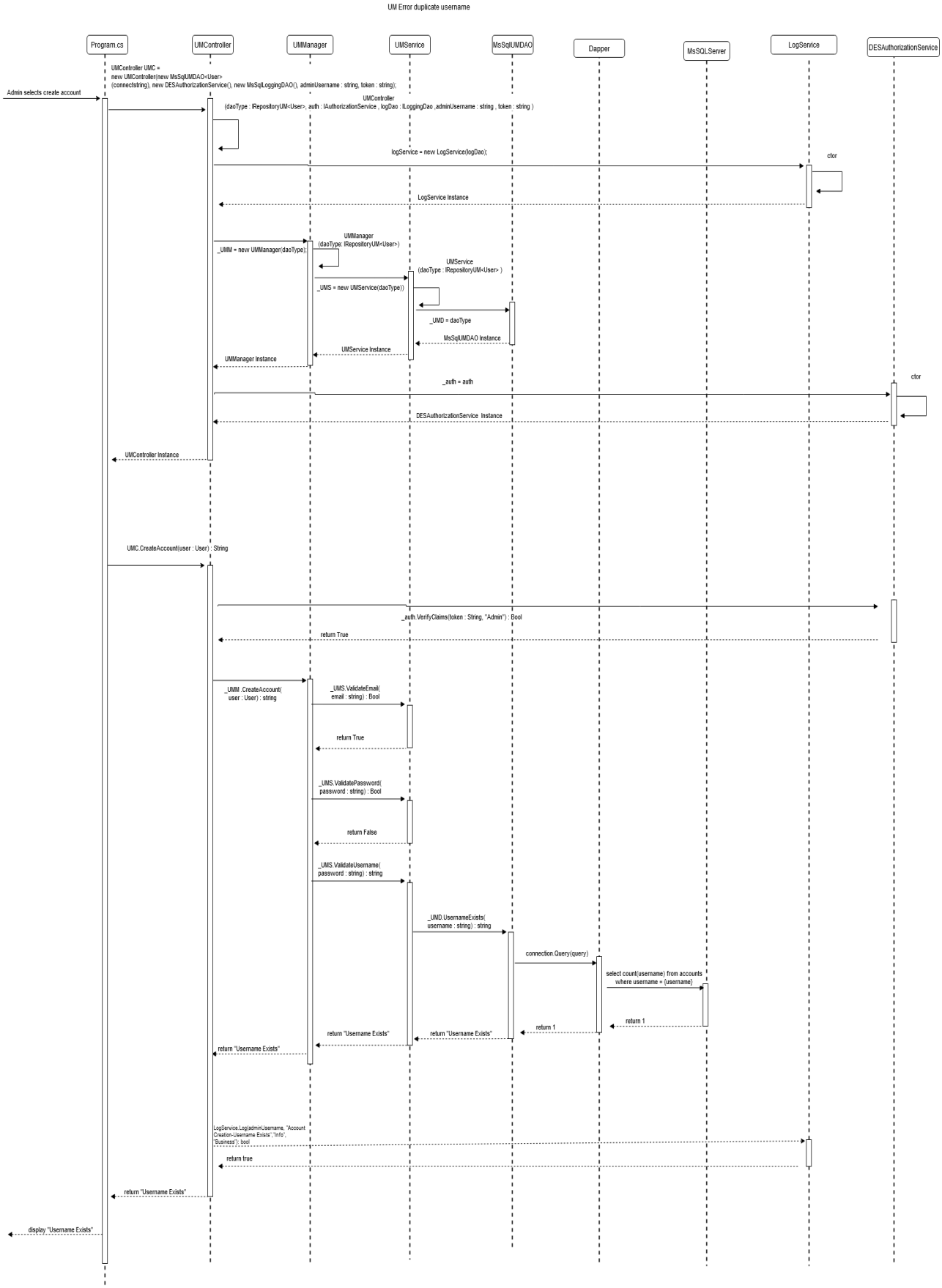
1.4 Create Account-Invalid Username

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Username Input is Invalid


UM Error Invalid Username
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
Admin selects create account
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
ctor
_auth = auth
DESAuthorizationService Instance
UMController Instance
UMC.CreateAccount(user : User) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMS.ValidateEmail(
email : string) : Bool
_UMM.CreateAccount(
user : User) : string
return True
_UMS.ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
return "Invalid Username"
return "Invalid Username"
LogService.Log(adminUsername,
"Account Creation=Invalid Username","Info",
"Business"): bool
return true
return "Invalid Username"
display "Invalid Username"

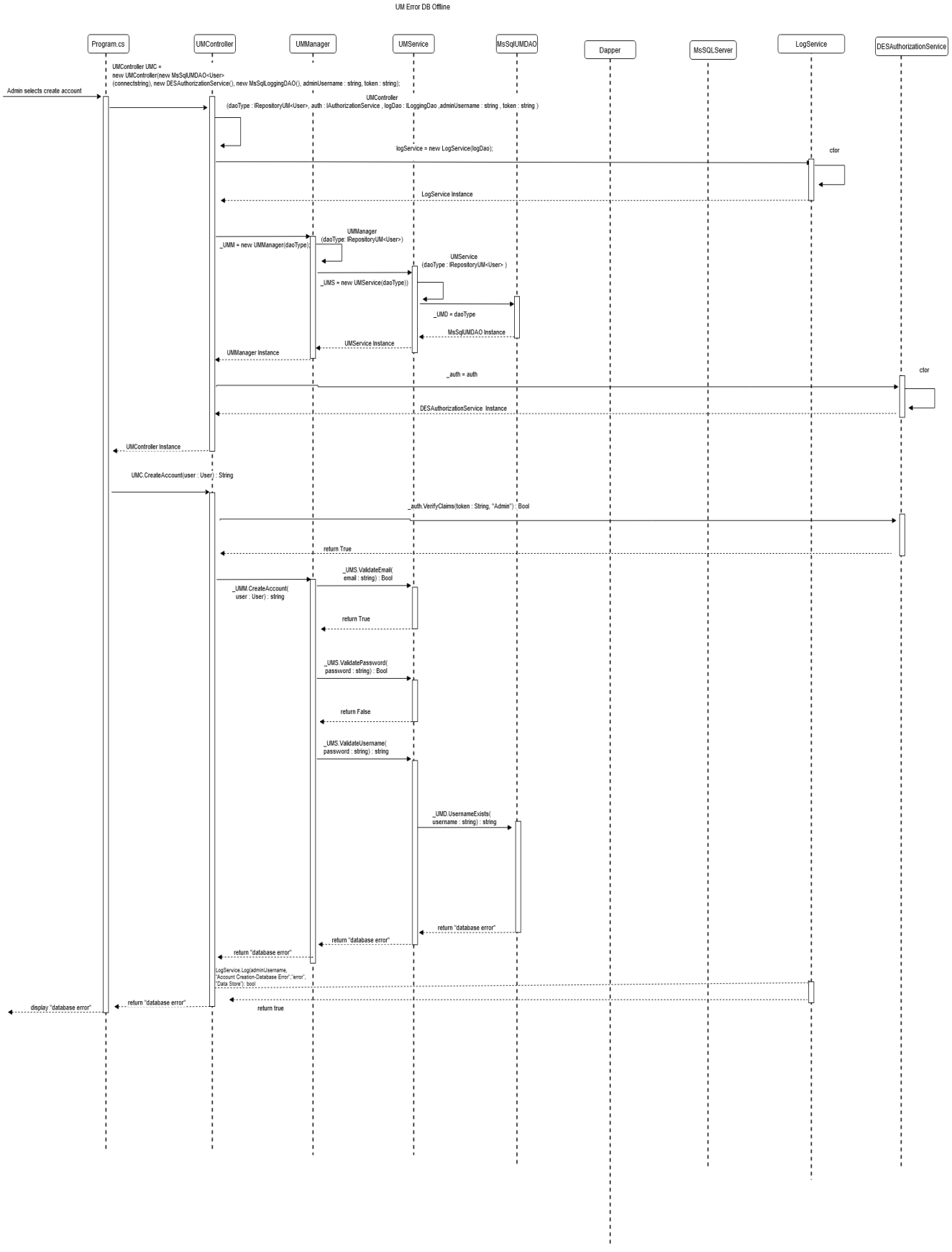
1.5 Create Account-Duplicate Username

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Username Input is invalid, already exists


UM Error duplicate username
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects create account
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.CreateAccount(user : User) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMM .CreateAccount(
user : User) : string
_UMS.ValidateEmail(
email : string) : Bool
return True
_UMS.ValidatePassword(
passwword : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
_UMD.UsernameExists(
username : string) : string
connection.Query(query)
select count(username) from accounts
where username = (username)
return 1
return 1
return "Username Exists"
return "Username Exists"
return "Username Exists"
LogService.Log(adminUsername, "Account
Creation-Username Exists","Info",
"Business"): bool
return true
return "Username Exists"
display "Username Exists"

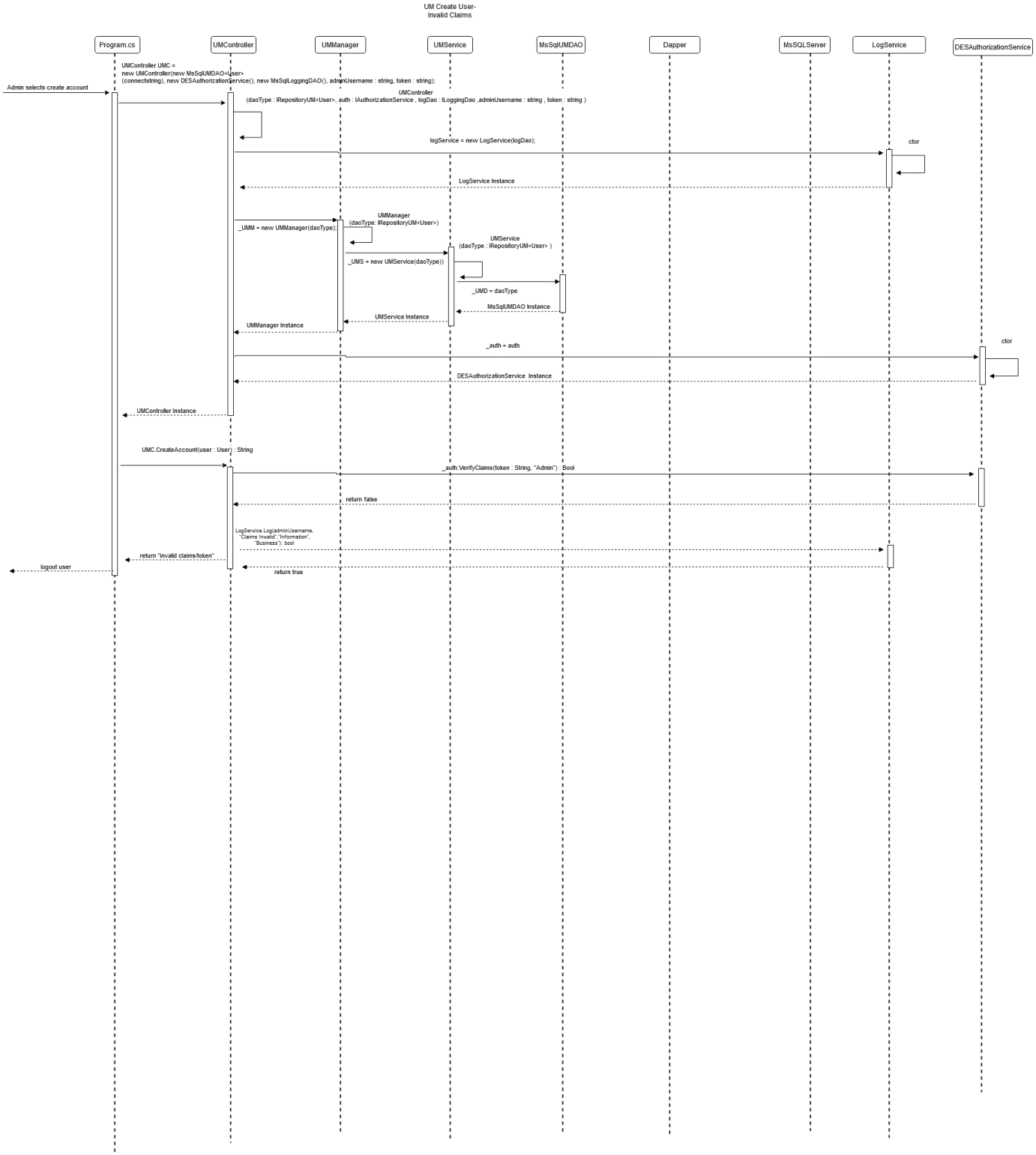
1.6 Create Account- Database Error

Precondition:
Administrator performs the action.
Database is offline.


UM Error DB Offline
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects create account
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.CreateAccount(user : User) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMS.ValidateEmail(
email : string) : Bool
_UMM.CreateAccount(
user : User) : string
return True
_UMS.ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
_UMD.UsernameExists(
username : string) : string
return "database error"
return "database error"
return "database error"
LogService.Log(adminUsername,
"Account Creation-Database Error", "error",
"Data Store"); bool
return "database error"
return true
display "database error"

1.7 Invalid Claims

Preconditions:
User has hacked into UM without logging into admin account
Token must not be generated by our private key or claims must be for regular Member

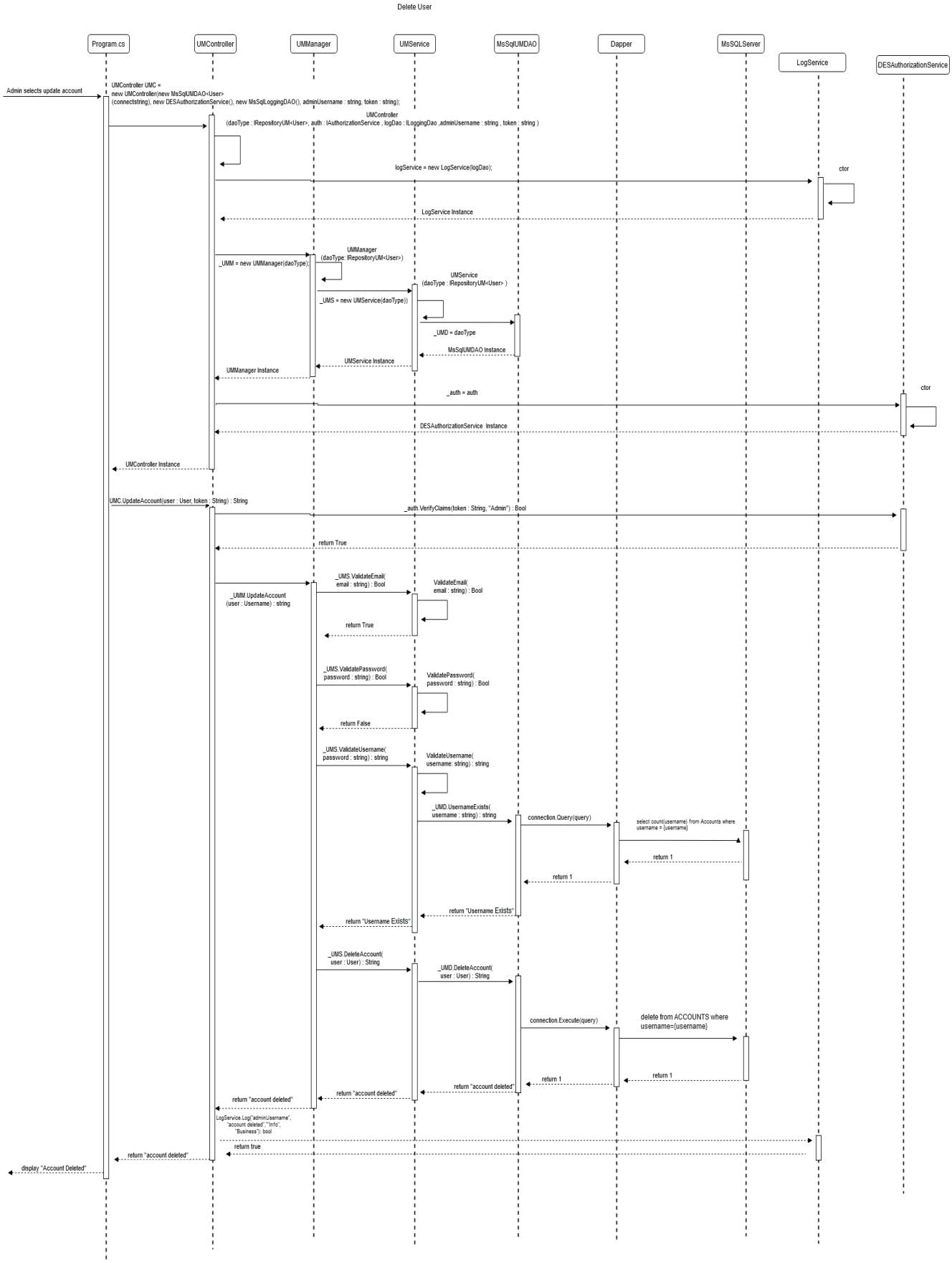


2 Delete User

Valid Inputs:
Username:
Username must be 7-15 character and length and only contain a-z,A-Z,0-9, .,@!
Usernam must be in database

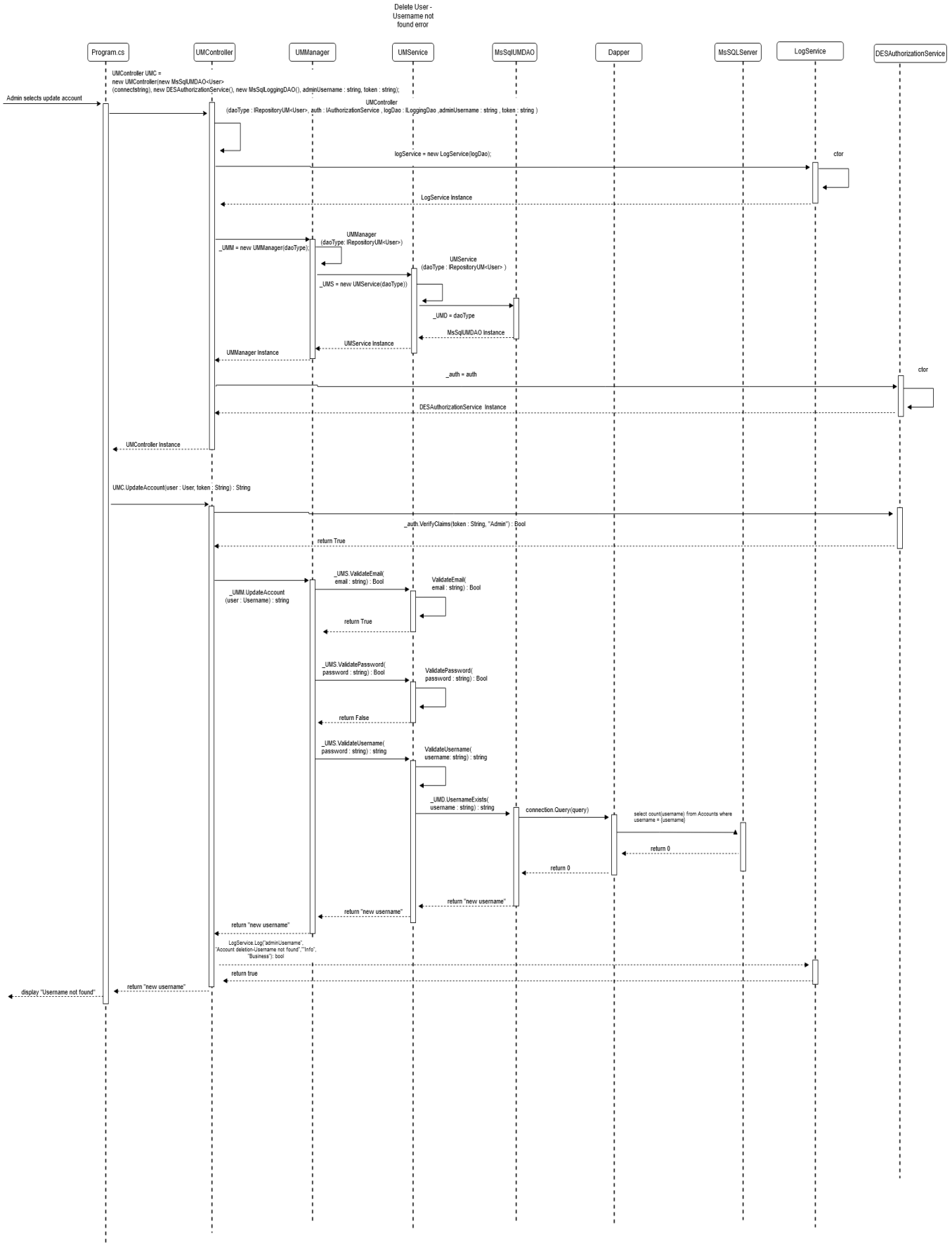
2.1 Successful Delete User

Precondition:
Administrator performs the action
Data Store is connected
Inputs are valid


Delete User
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
Admin selects update account
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.UpdateAccount(user : User, token : String) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMS.ValidateEmail(
email : string) : Bool
ValidateEmail(
email : string) : Bool
_UMM.UpdateAccount
(user : Username) : string
return True
_UMS.ValidatePassword(
password : string) : Bool
ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
ValidateUsername(
username: string) : string
_UMD.UsernameExists(
username : string) : string
connection.Query(query)
select count(username) from Accounts where
username = {username}
return 1
return 1
return "Username Exists"
return "Username Exists"
_UMS.DeleteAccount(
user : User) : String
_UMD.DeleteAccount(
user : User) : String
connection.Execute(query)
delete from ACCOUNTS where
username={username}
return 1
return 1
return "account deleted"
return "account deleted"
return "account deleted"
LogService.Log("adminUsername",
"account deleted","Info",
"Business"); bool
return true
return "account deleted"
display "Account Deleted"

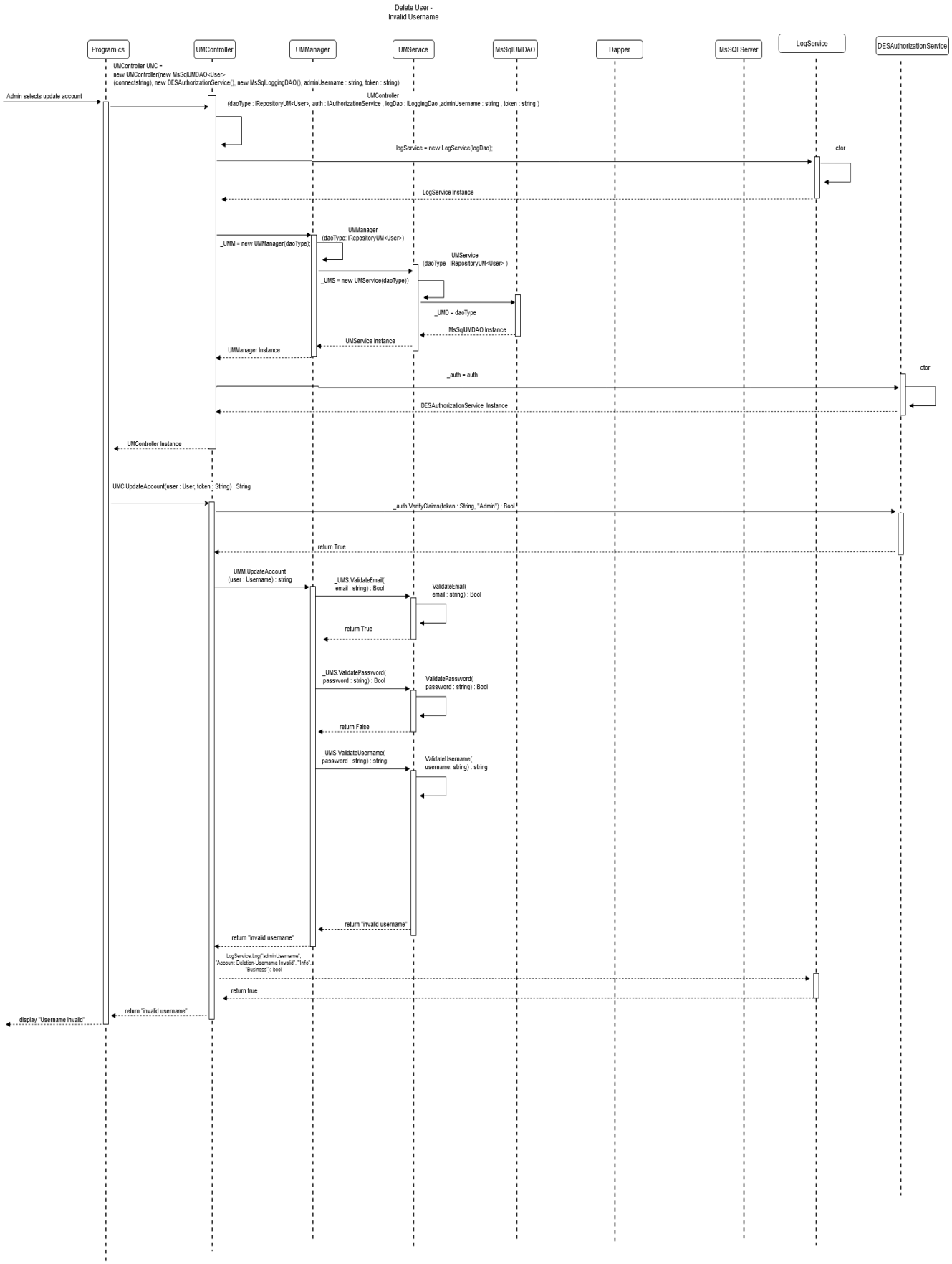
2.2 Delete User no Username found

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Username provided not in database


Delete User - Username not found error
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC = new UMController(new MsSqlUMDAO<User> (connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects update account
UMController (daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager (daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService (daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.UpdateAccount(user : User, token : String) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMS.ValidateEmail(email : string) : Bool
ValidateEmail(email : string) : Bool
_UMM.UpdateAccount (user : Username) : string
return True
_UMS.ValidatePassword(password : string) : Bool
ValidatePassword(password : string) : Bool
return False
_UMS.ValidateUsername(password : string) : string
ValidateUsername(username: string) : string
_UMD.UsernameExists(username : string) : string
connection.Query(query)
select count(username) from Accounts where username = (username)
return 0
return 0
return "new username"
return "new username"
return "new username"
LogService.Log("adminUsername", "Account deletion-Username not found","Info", "Business"): bool
return true
return "new username"
display "Username not found"

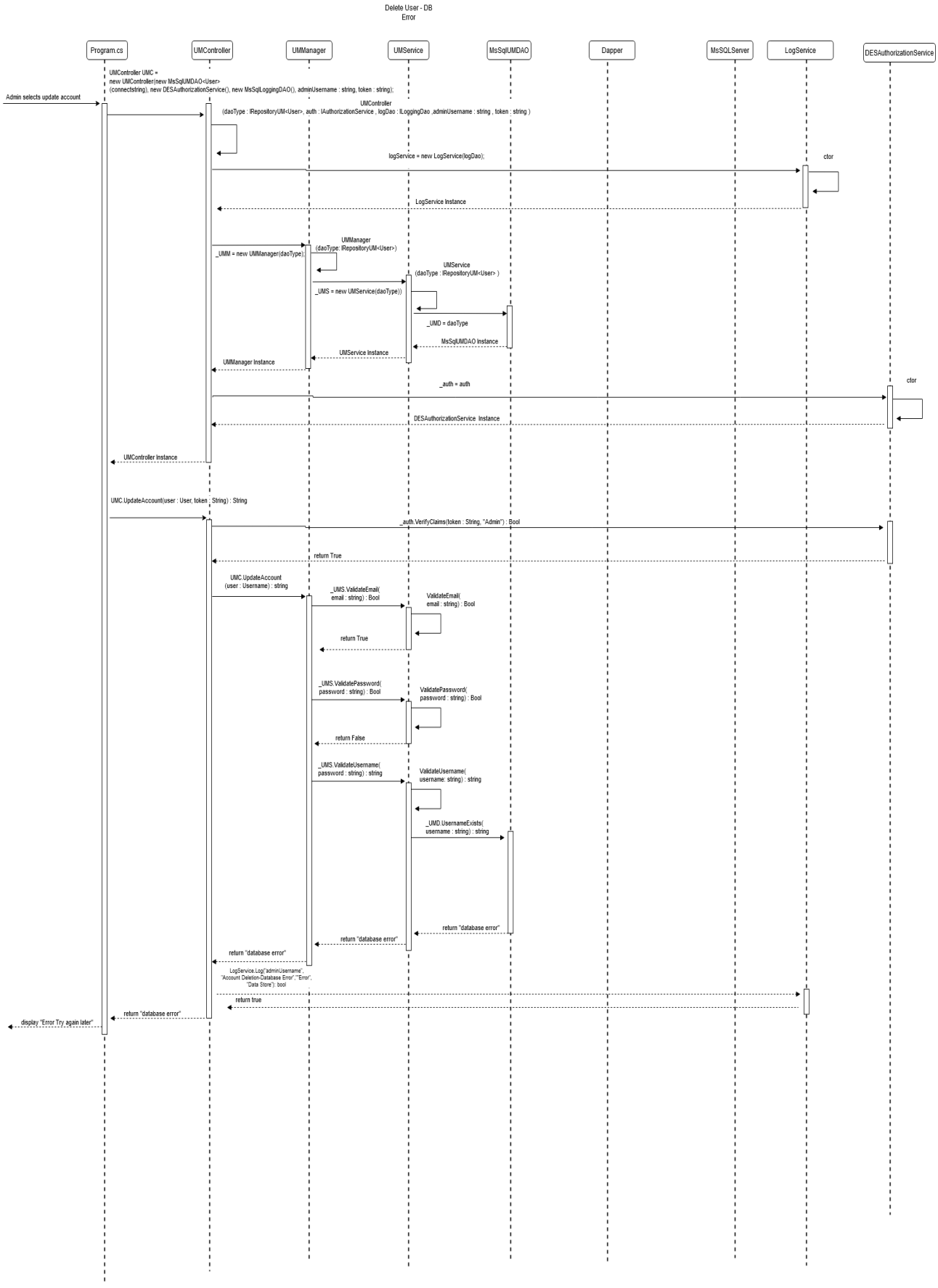
2.3 Delete User- invalid username

Precondition:
Administrator performs the action
SQL Server Data Store connection error
Username provided not following Business requirements


Delete User -
Invalid Username
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectString), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects update account
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
ctor
_auth = auth
DESAuthorizationService Instance
UMController Instance
UMC.UpdateAccount(user : User, token : String) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
UMM.UpdateAccount
(user : Username) : string
_UMS.ValidateEmail(
email : string) : Bool
ValidateEmail(
email : string) : Bool
return True
_UMS.ValidatePassword(
password : string) : Bool
ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
passwword : string) : string
ValidateUsername(
username: string) : string
return "invalid username"
return "invalid username"
LogService.Log("adminUsername",
"Account Deletion-Username Invalid","Info",
"Business"): bool
return true
return "invalid username"
display "Username Invalid"

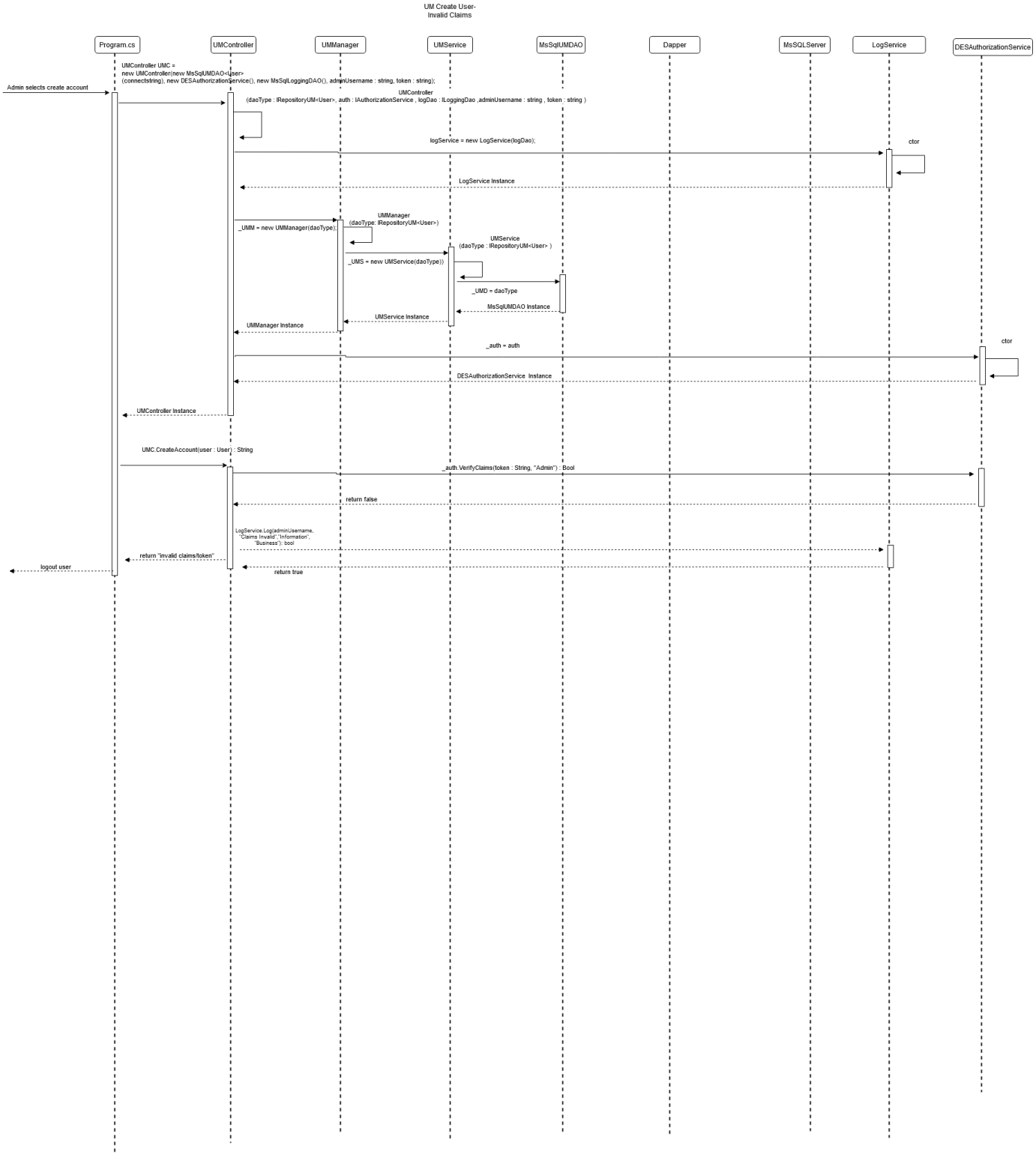
2.4 Delete User- Database Error

Precondition:
Administrator performs the action
Data Store is connected
Input Is valid


Delete User - DB Error
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
UMController UMC = new UMController(new MsSqlUMDAO<User> (connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
Admin selects update account
UMController (daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager (daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService (daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = auth
ctor
DESAuthorizationService Instance
UMController Instance
UMC.UpdateAccount(user : User, token : String) : String
_auth.VerifyClaims(token : String, "Admin") : Bool
return True
UMC.UpdateAccount (user : Username) : string
_UMS.ValidateEmail(email : string) : Bool
ValidateEmail(email : string) : Bool
return True
_UMS.ValidatePassword(password : string) : Bool
ValidatePassword(password : string) : Bool
return False
_UMS.ValidateUsername(password : string) : string
ValidateUsername(username: string) : string
_UMD.UsernameExists(username : string) : string
return "database error"
return "database error"
return "database error"
LogService.Log("adminUsername", "Account Deletion-Database Error","Error", "Data Store"): bool
return true
return "database error"
display "Error.Try again later"

2.5 Invalid Claims

Preconditions:
User has hacked into UM without logging into admin account
Token must not be generated by our private key or claims must be for regular Member

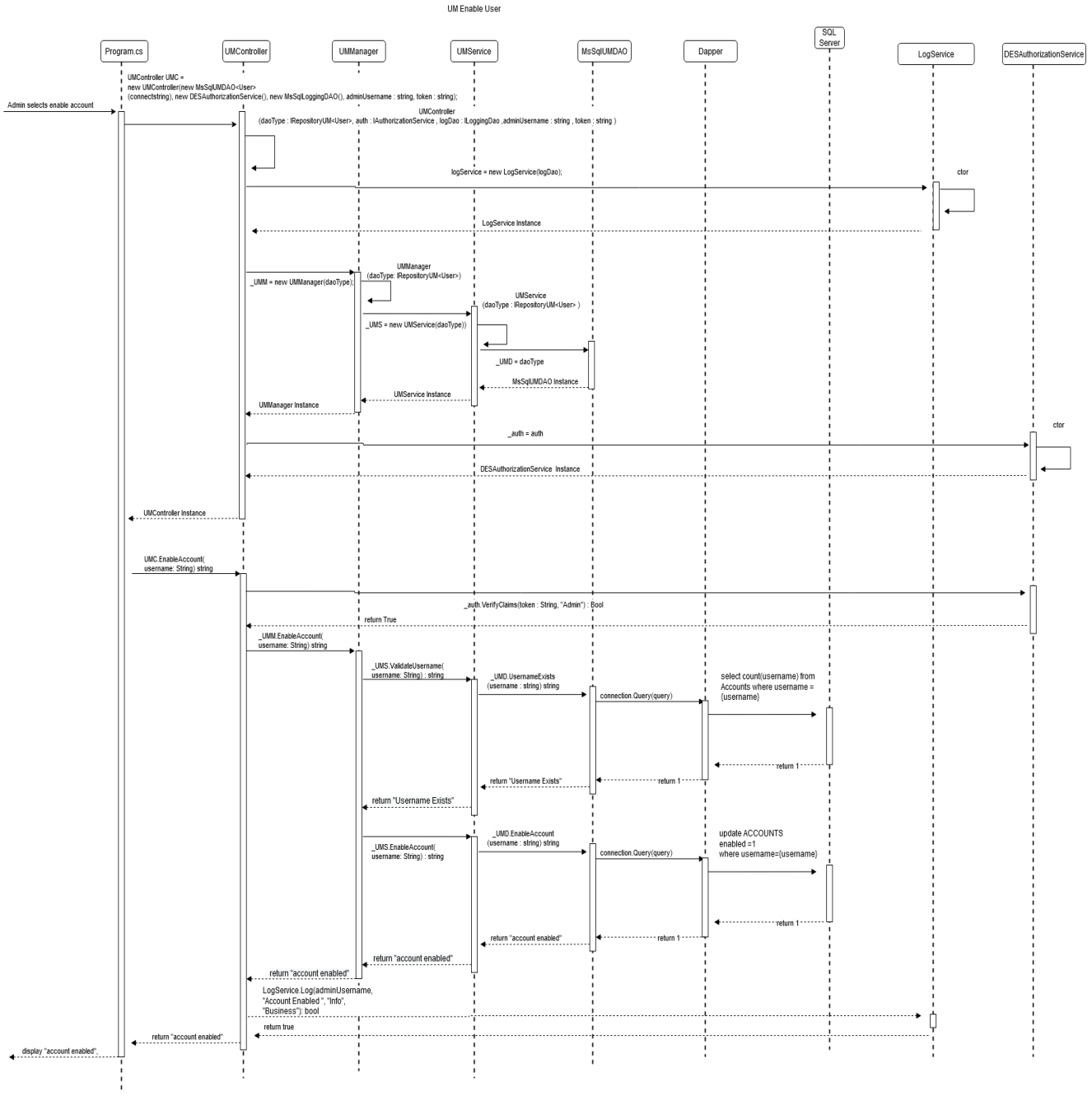


3 Enable Account

Valid Inputs:
Username:
Username must be 7-15 character and length and only contain a-z,A-Z,0-9, .,@!
Usernam must be in database

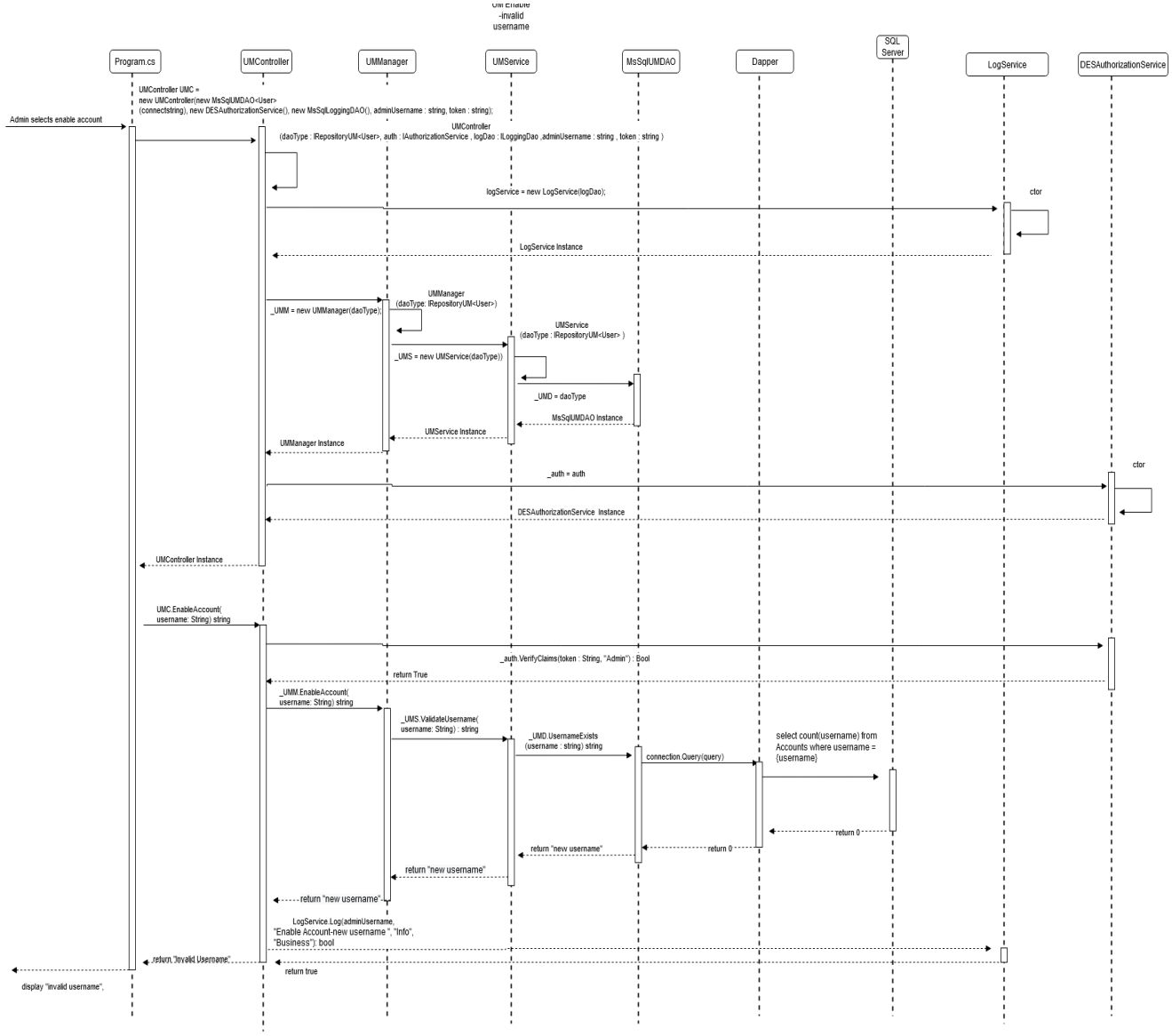
3.1 Enable User Successful

Preconditions
Administrator performs the action
Data Store is connected
Inputs are valid



3.2 Enable Username Invalid

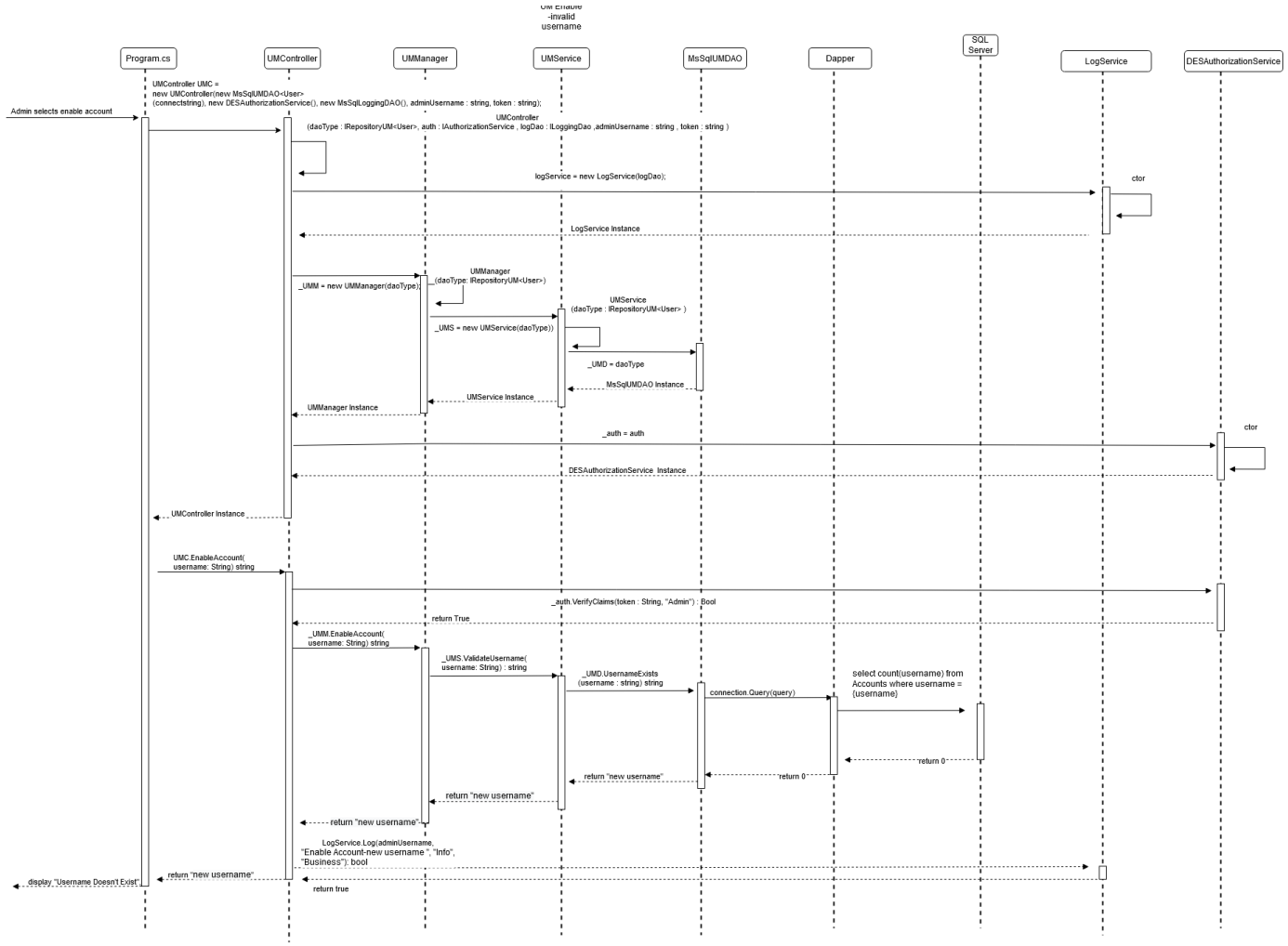
Precondition:
Administrator performs the action
Data Store is connected
Inputs are invalid



3.3 Enable User No Username Found

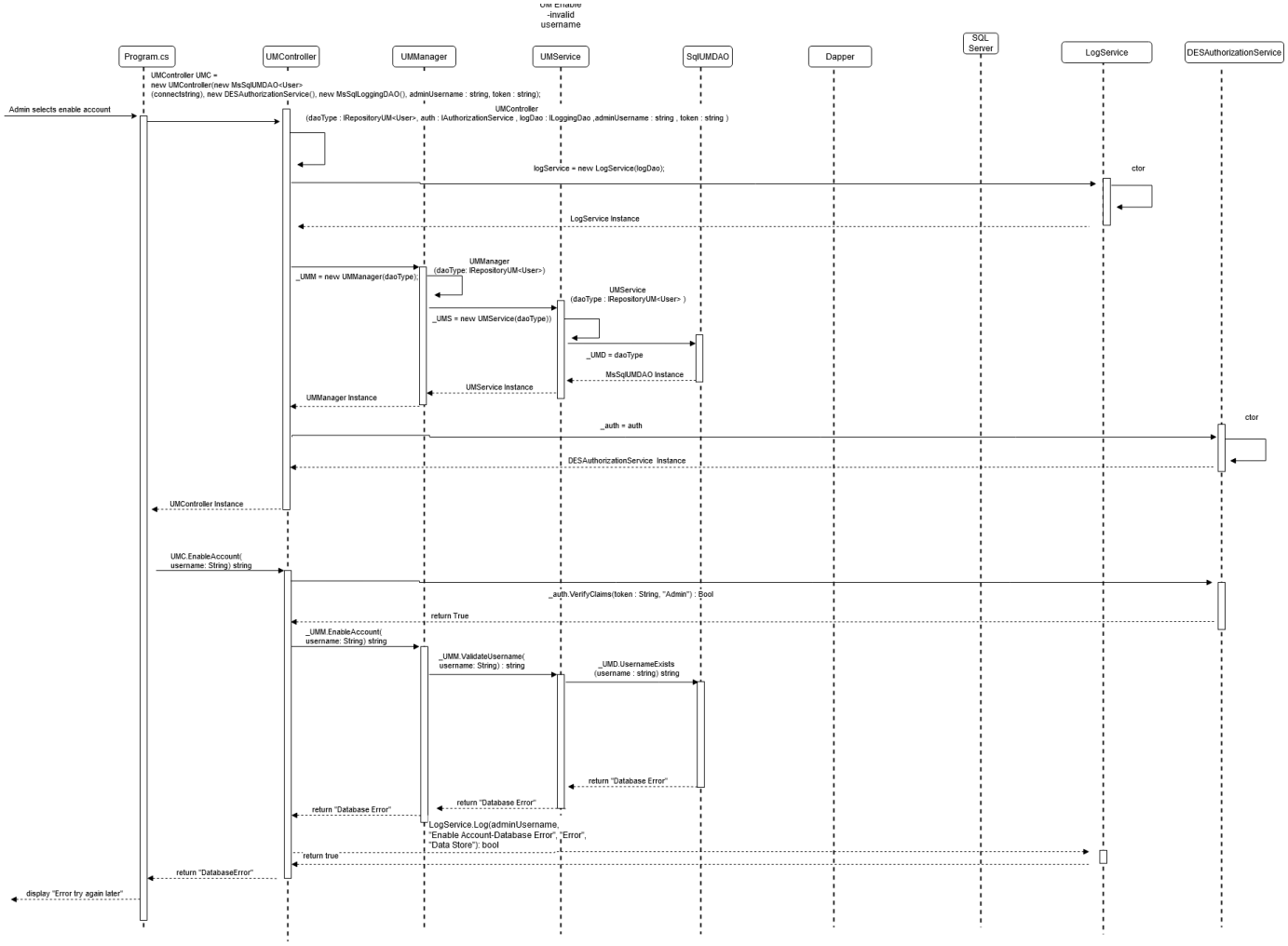
Precondition:
Administrator performs the action
Data Store is connected

Inputs are not in database



3.4 Enable User Database Error

Precondition:
Administrator performs the action
Data Store is improperly connected
Inputs are valid

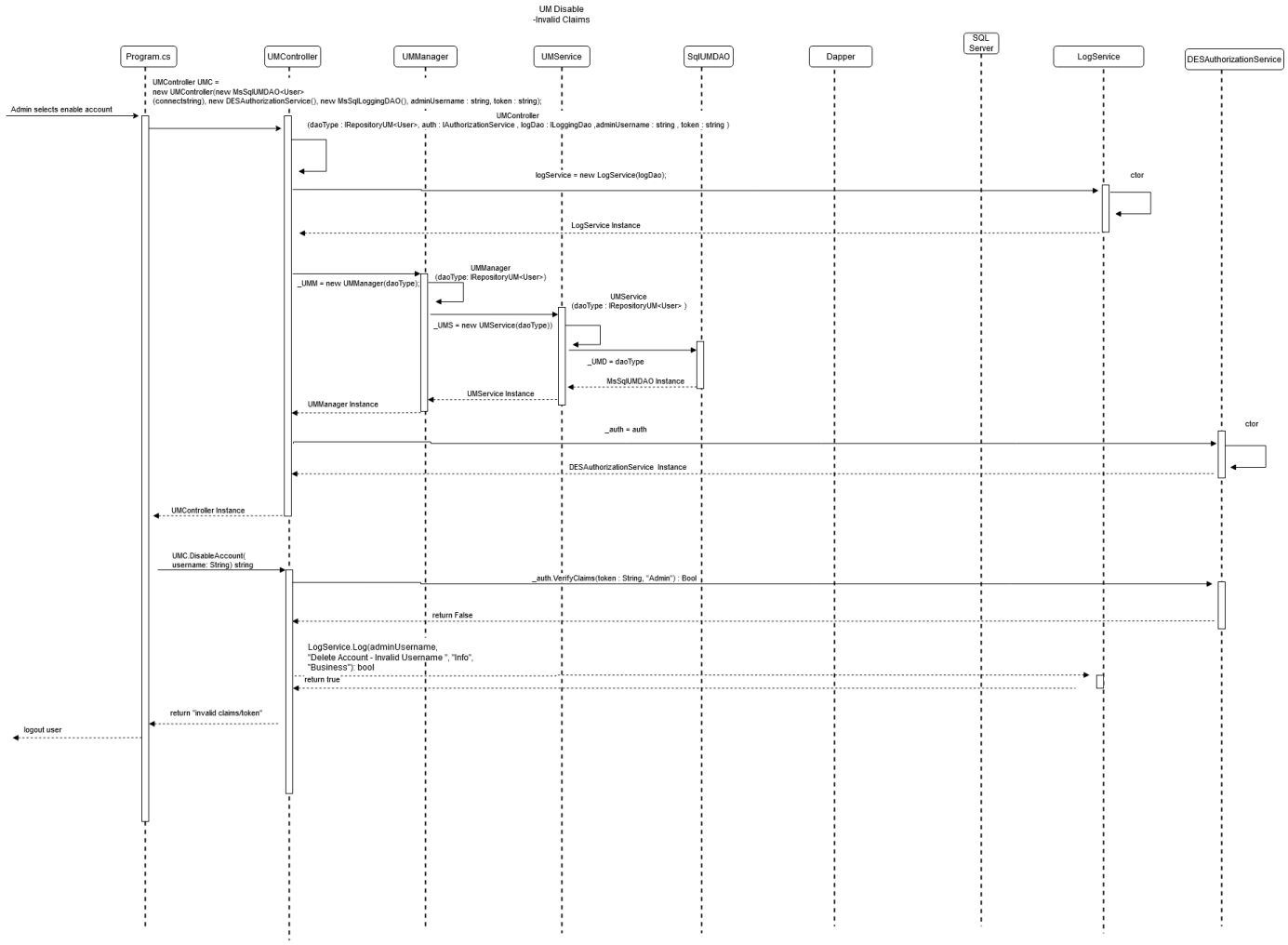


3.5 Invalid Claims

Preconditions:

User has hacked into UM without logging into admin account

Token must not be generated by our private key or claims must be for regular Member

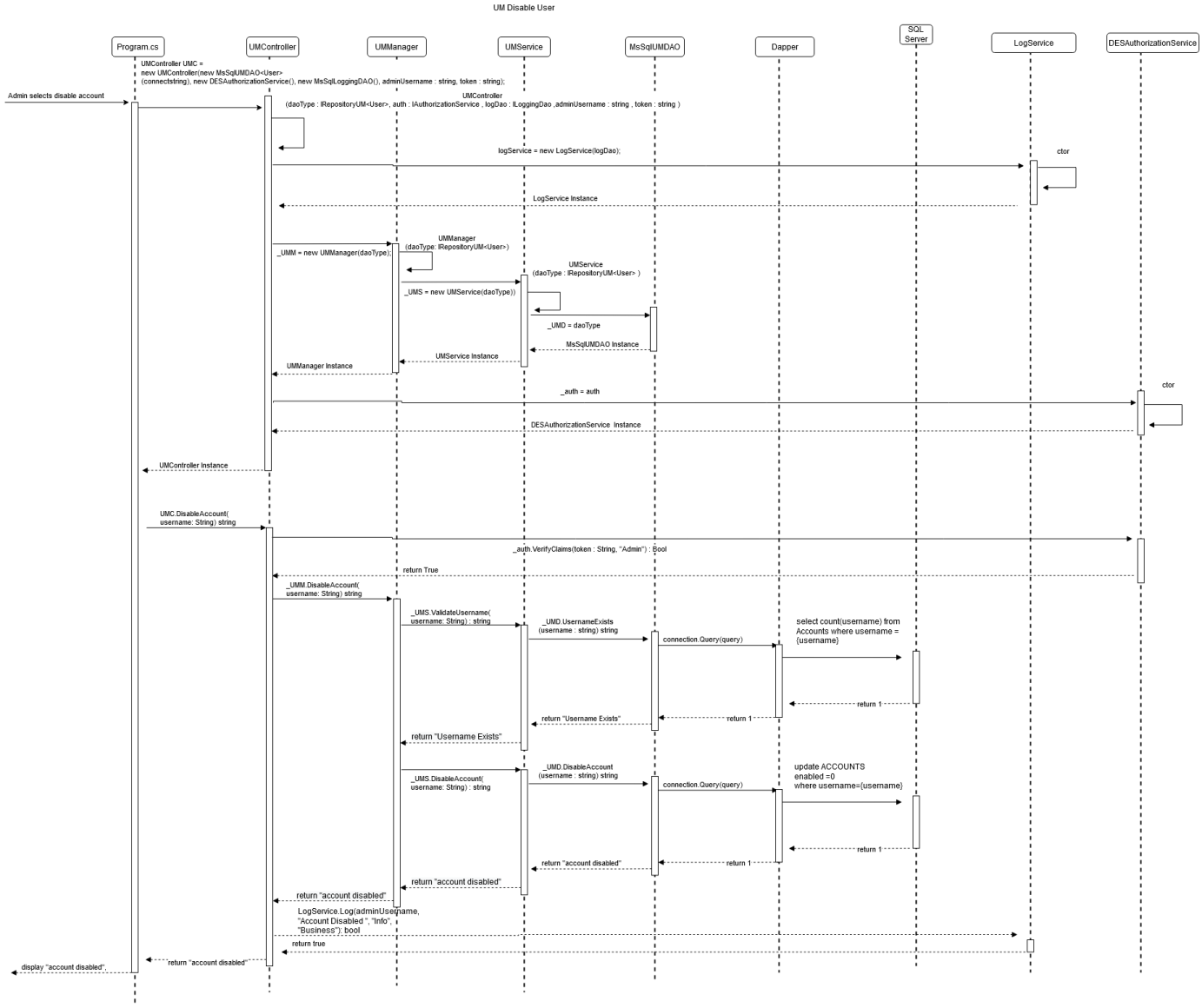


4 Disable User

Valid Inputs:
Username:
Username must be 7-15 character and length and only contain a-z,A-Z,0-9, .,@!
Usernam must be in database

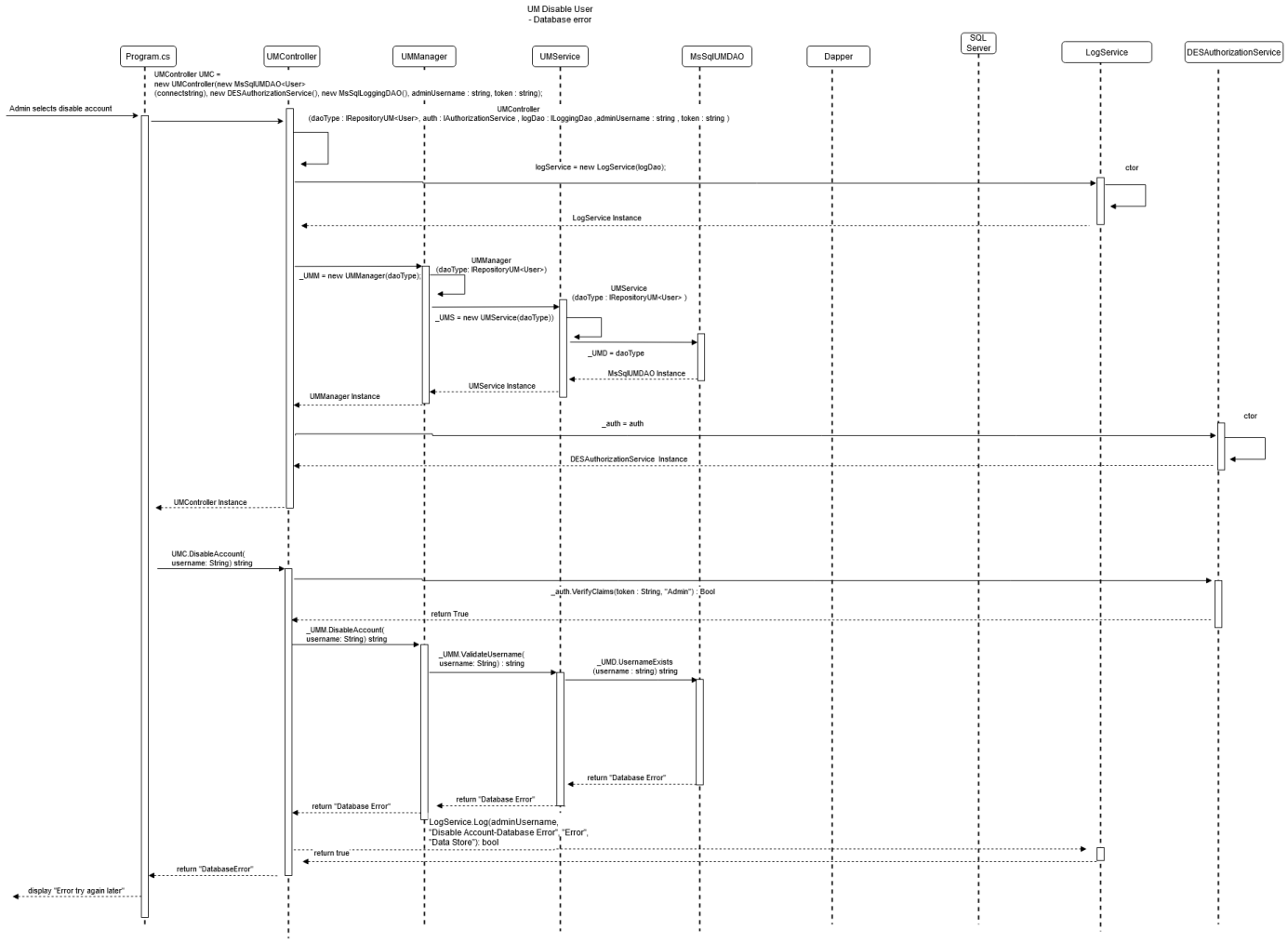
4.1 Disable User Successful

Precondition:
Administrator performs the action
Data Store is connected
Inputs are valid



4.2 Disable User Database Error

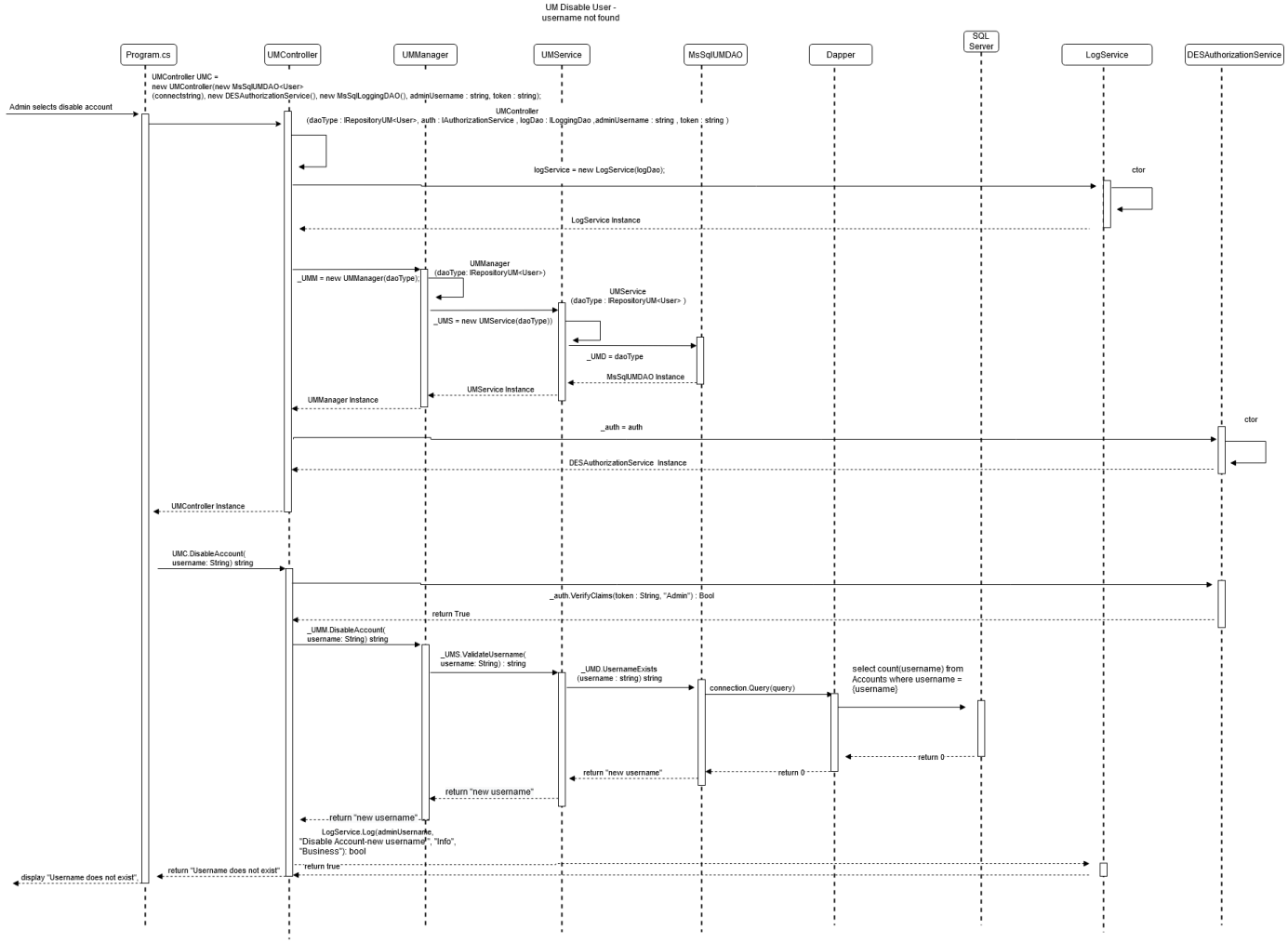
Precondition:
Administrator performs the action
Data Store is improperly connected
Inputs are invalid



4.3 Enable User No Username Found

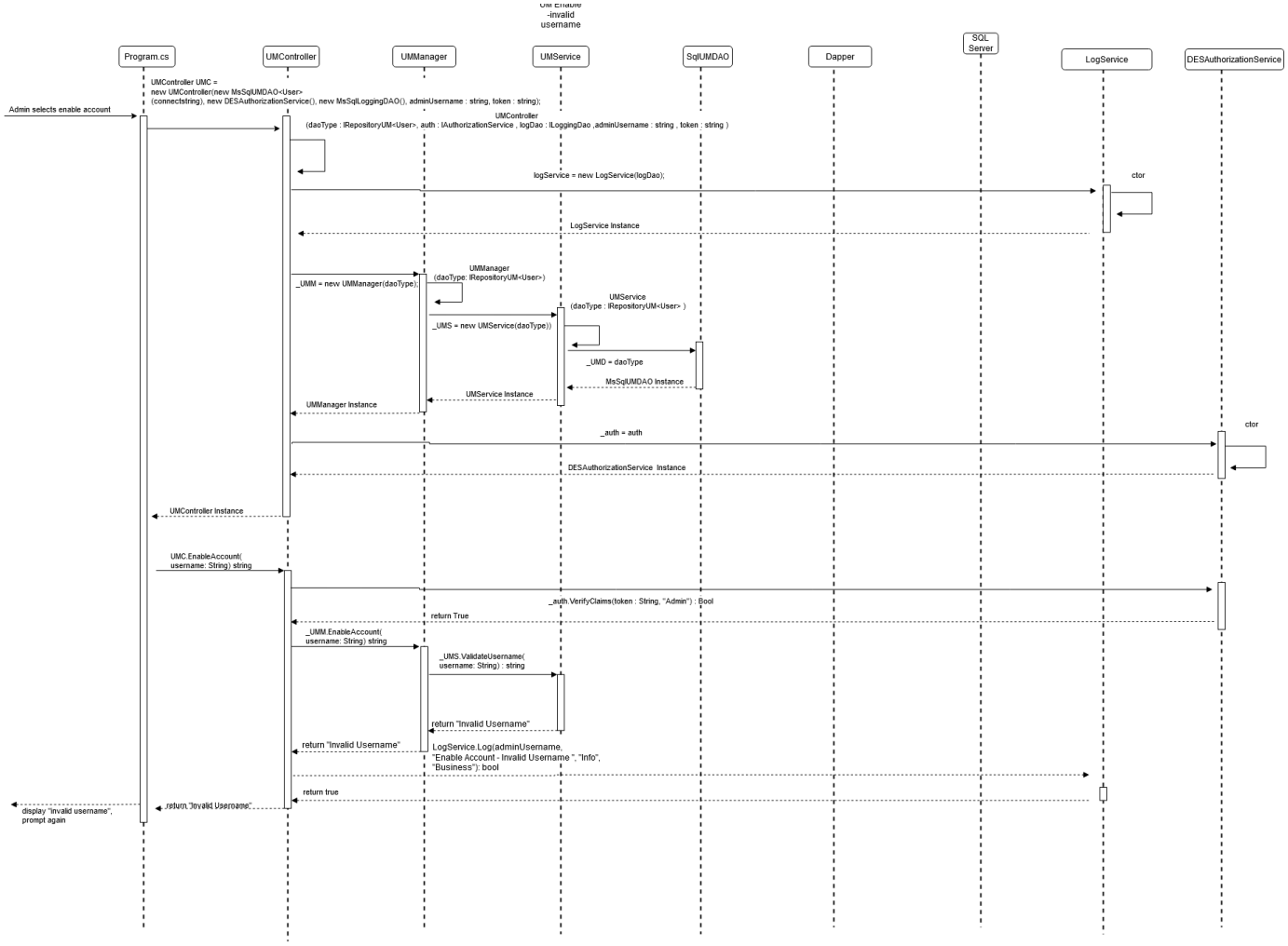
Precondition:
Administrator performs the action
Data Store is connected

Inputs are not database



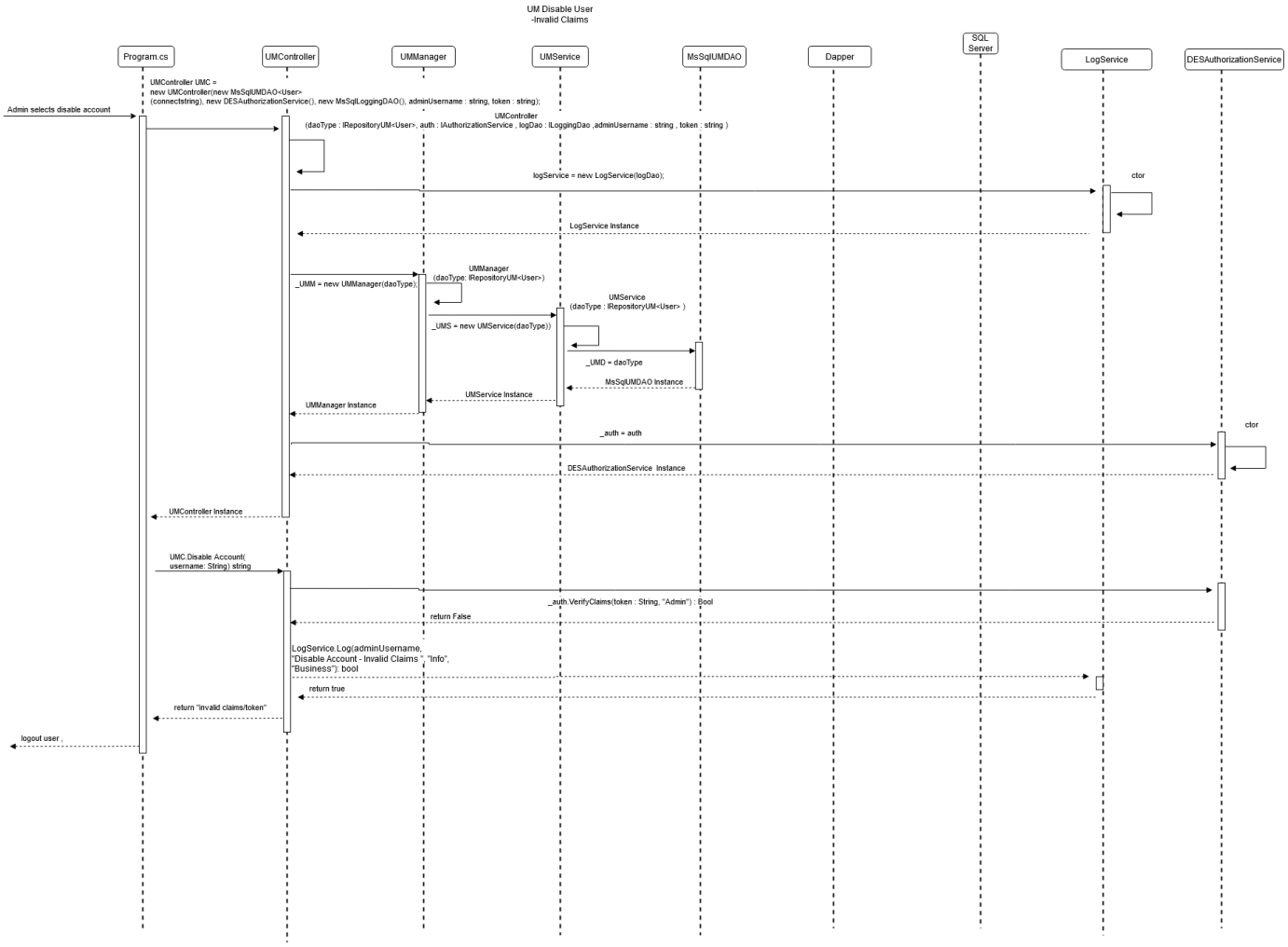
4.4 Enable Username Invalid

Precondition:
Administrator performs the action
Data Store is connected
Inputs are invalid



4.5 Invalid Claims

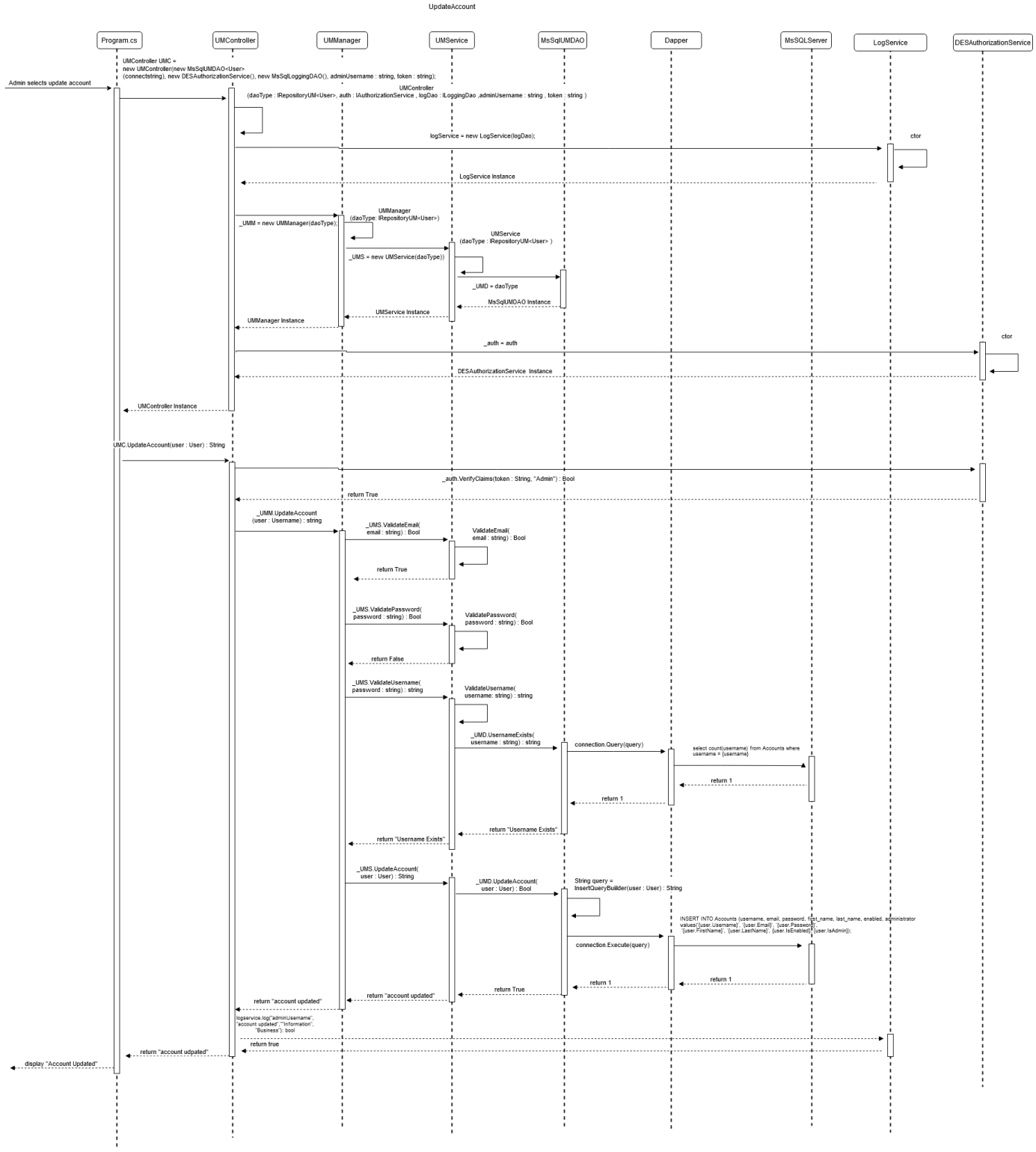
Preconditions:
User has hacked into UM without logging into admin account
Token must not be generated by our private key or claims must be for regular Member



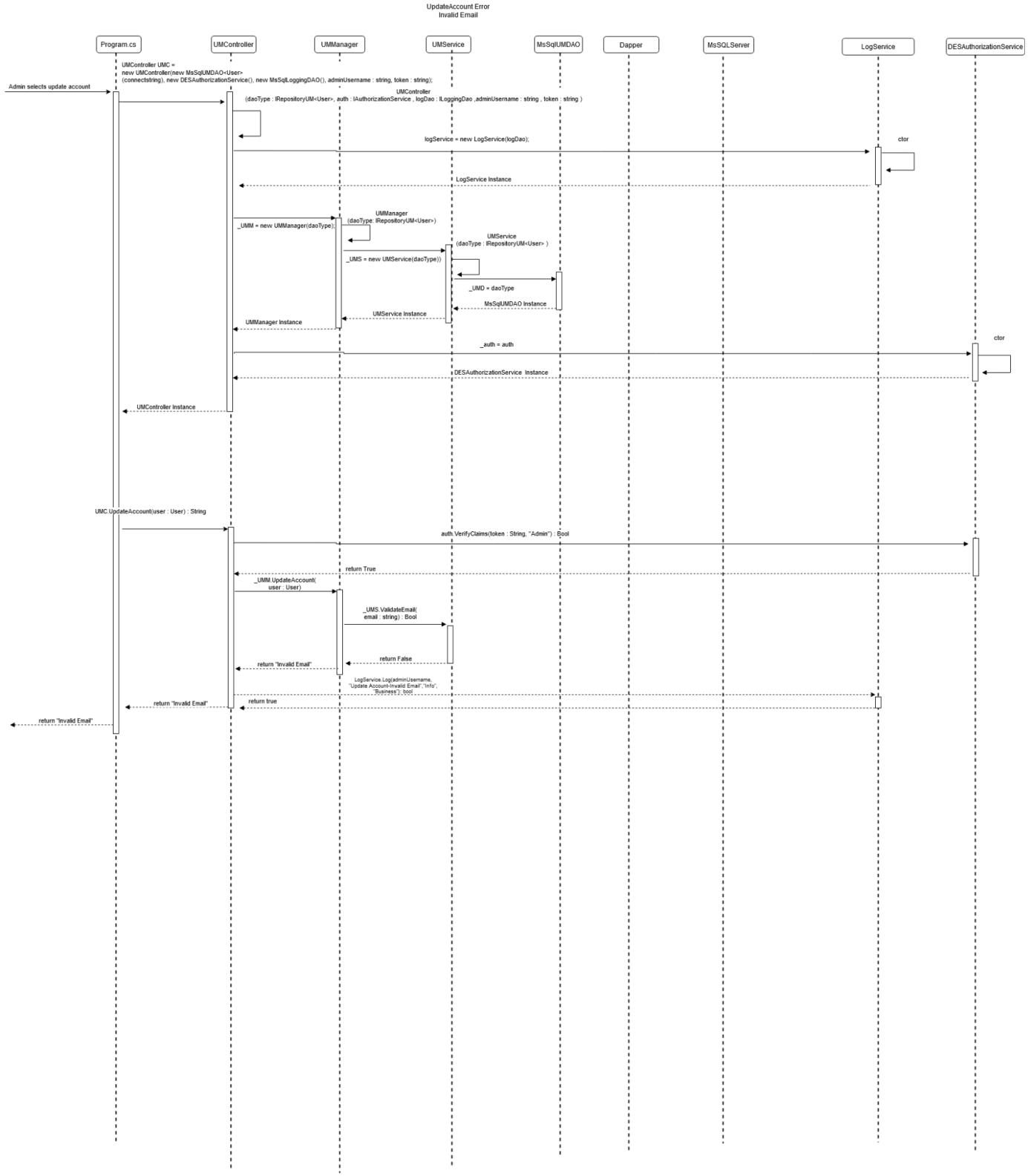
5 Update User

Valid Inputs:
Username:
Username must be 7-15 character and length and only contain a-z,A-Z,0-9, .,@!
Usernam must be in database

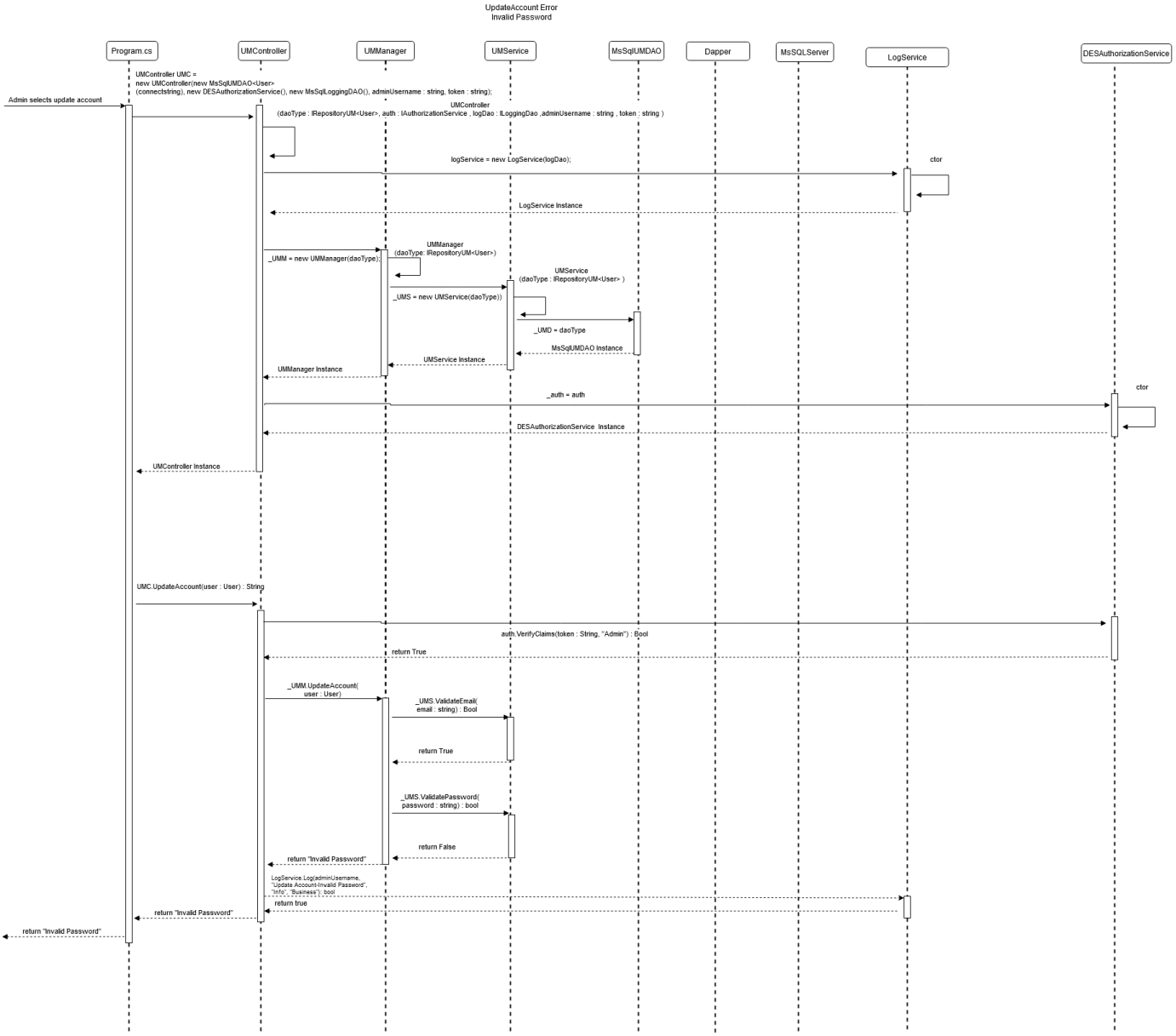
5.1 Update User



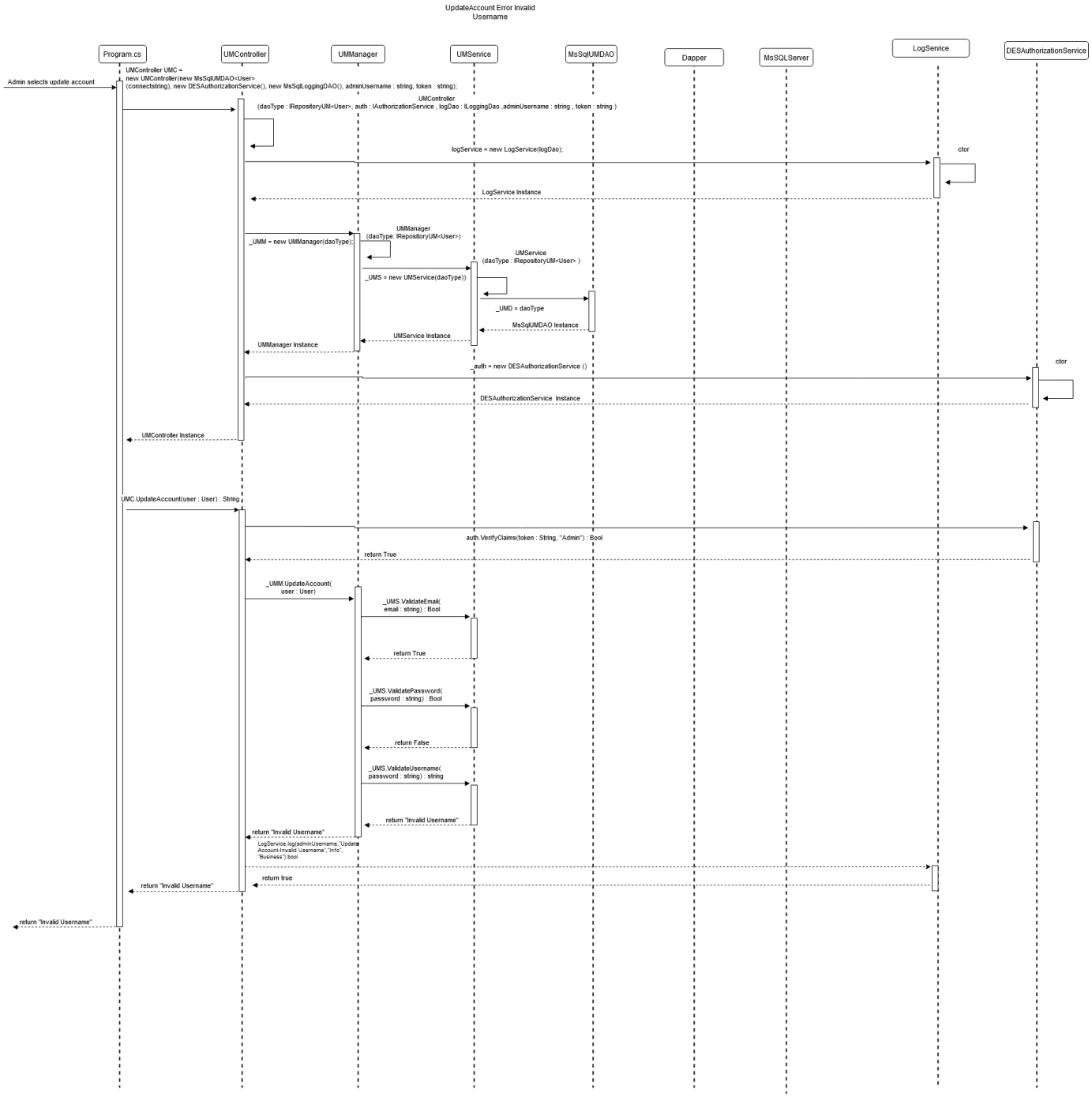
5.2 Invalid Email



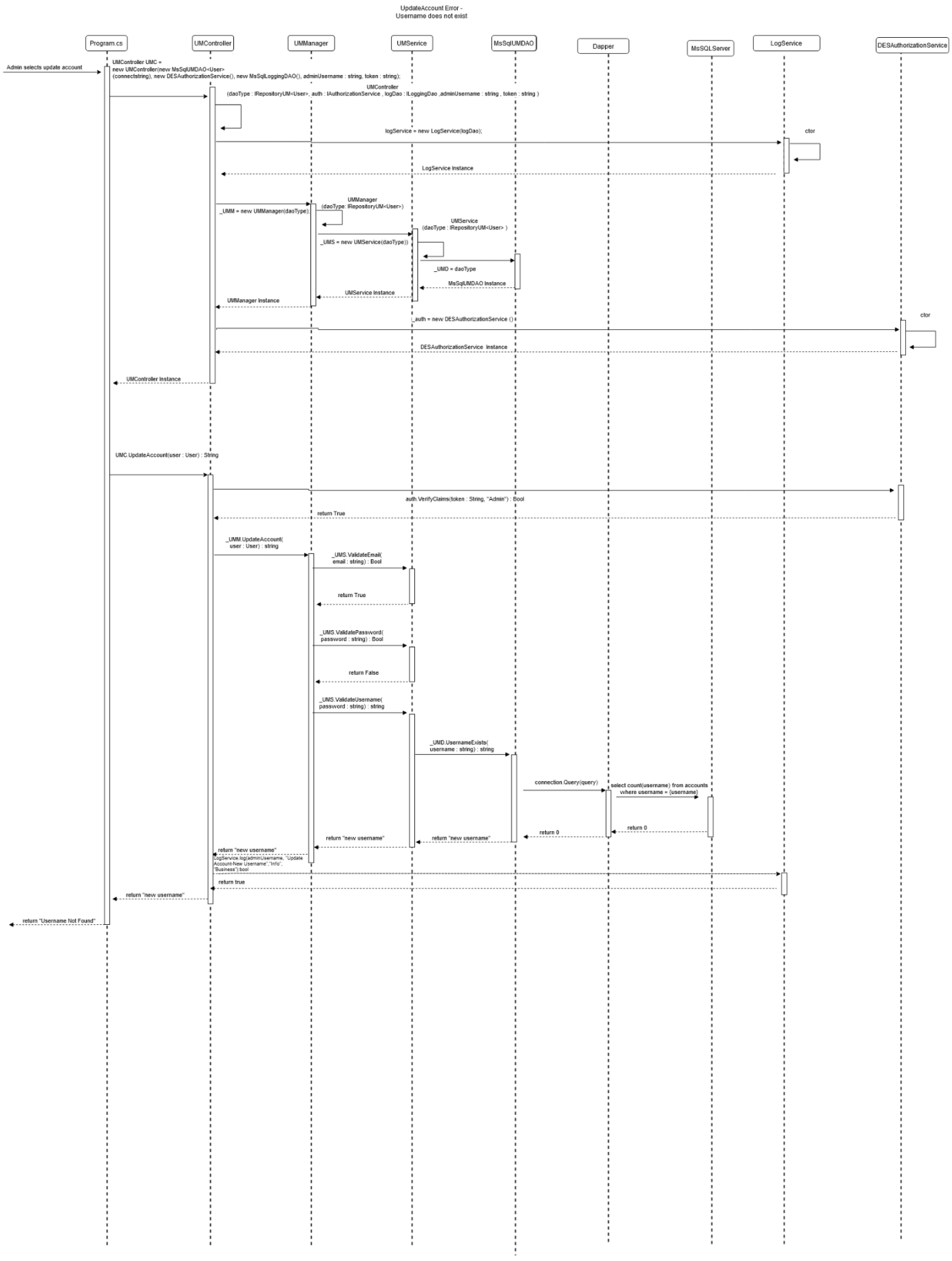
5.3 Invalid Password



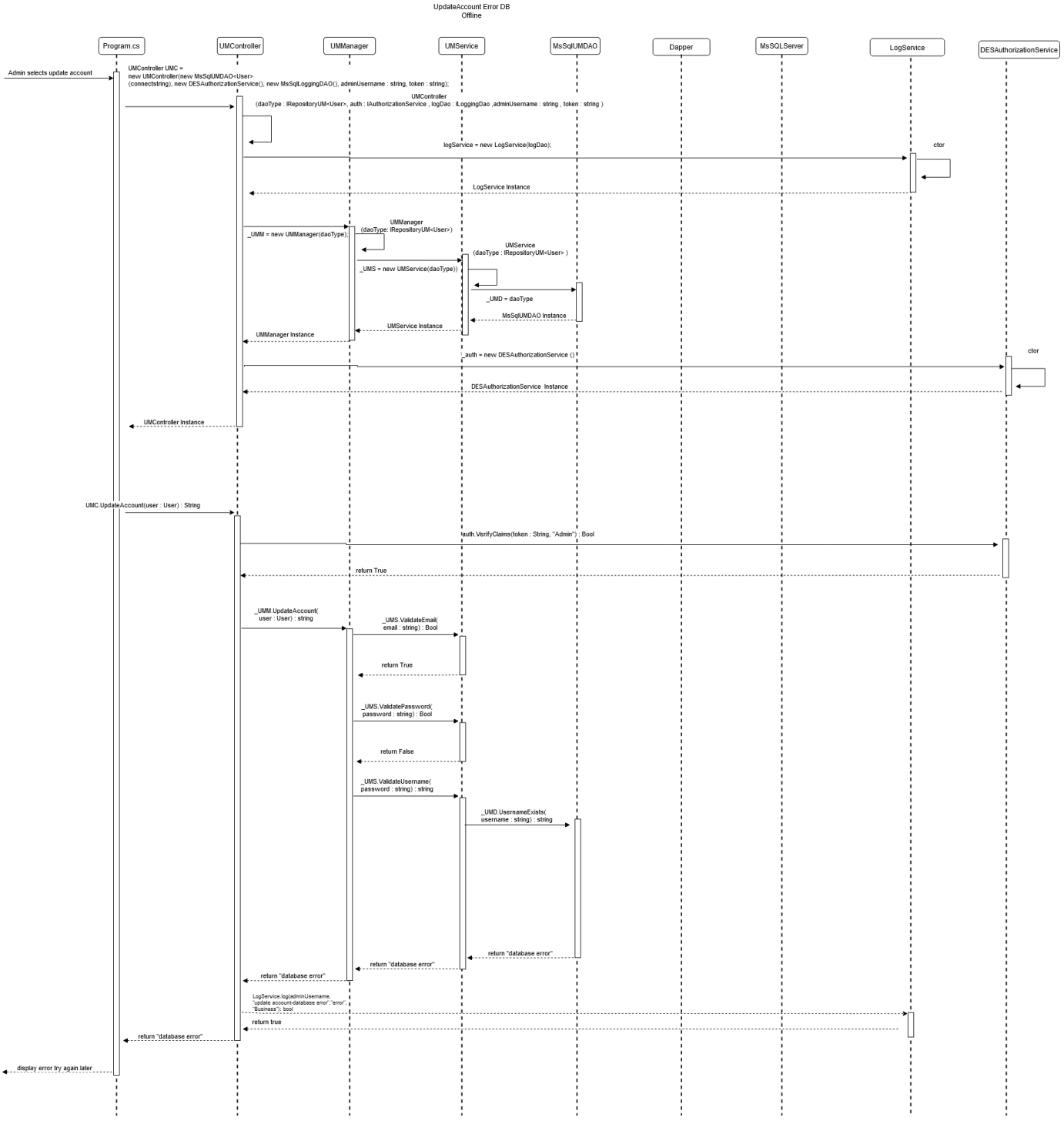
5.4 Invalid Username



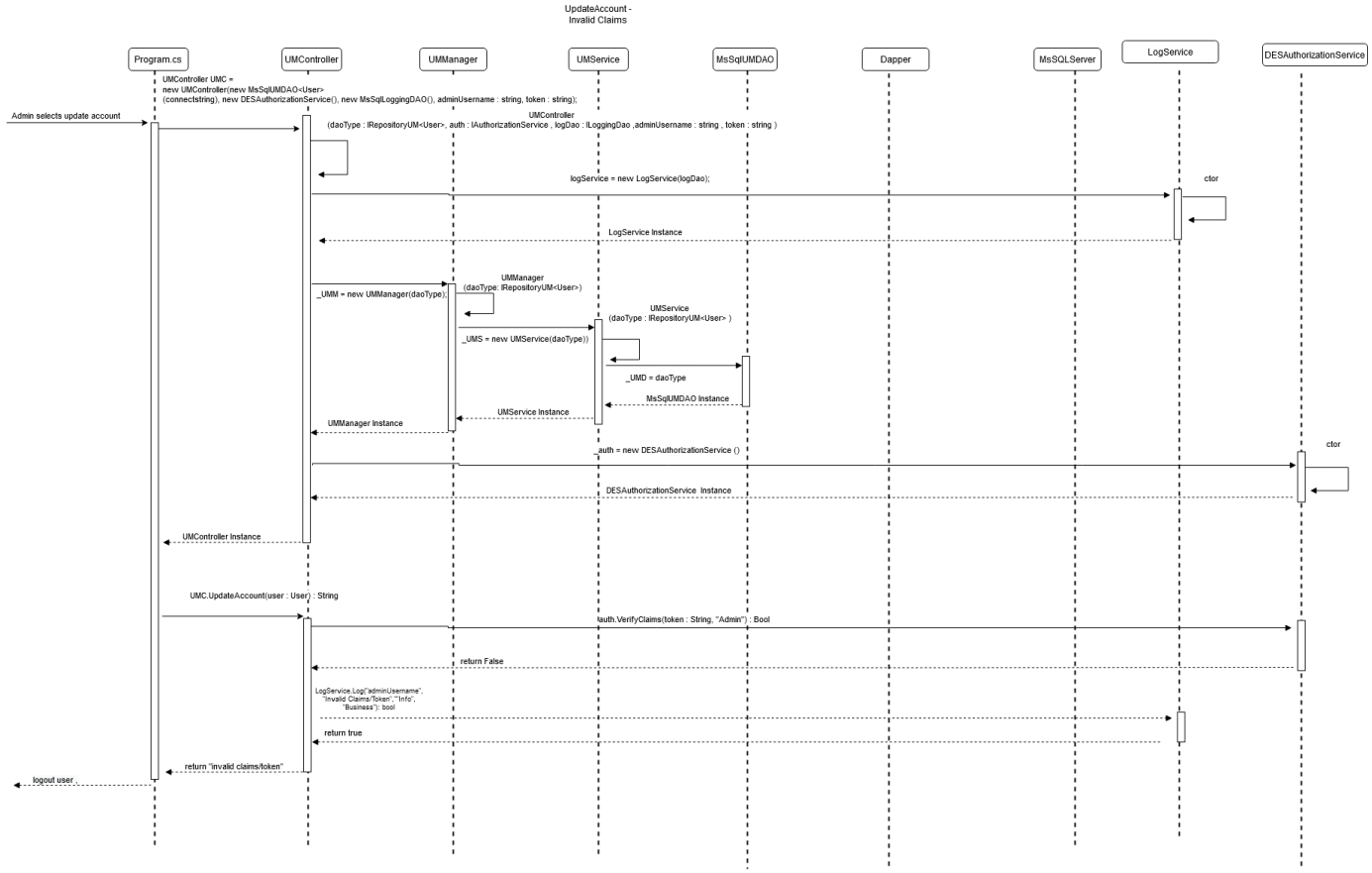
5.5 No Username Found


UpdateAccount Error -
Username does not exist
Program.cs
UMController
UMManager
UMService
MsSqlUMDAO
Dapper
MsSQLServer
LogService
DESAuthorizationService
Admin selects update account
UMController UMC =
new UMController(new MsSqlUMDAO<User>
(connectstring), new DESAuthorizationService(), new MsSqlLoggingDAO(), adminUsername : string, token : string);
UMController
(daoType : IRepositoryUM<User>, auth : IAuthorizationService , logDao : ILoggingDao ,adminUsername : string , token : string)
logService = new LogService(logDao);
ctor
LogService Instance
UMManager
(daoType: IRepositoryUM<User>)
_UMM = new UMManager(daoType);
UMService
(daoType : IRepositoryUM<User>)
_UMS = new UMService(daoType))
_UMD = daoType
MsSqlUMDAO Instance
UMService Instance
UMManager Instance
_auth = new DESAuthorizationService ()
ctor
DESAuthorizationService Instance
UMController Instance
UMC.UpdateAccount(user : User) : String
auth.VerifyClaims(token : String, "Admin") : Bool
return True
_UMM.UpdateAccount(
user : User) : string
_UMS.ValidateEmail(
email : string) : Bool
return True
_UMS.ValidatePassword(
password : string) : Bool
return False
_UMS.ValidateUsername(
password : string) : string
_UMD.UsernameExists(
username : string) : string
connection.Query(query)
select count(username) from accounts
where username = (username)
return 0
return 0
return "new username"
return "new username"
return "new username"
LogService.log(adminUsername, "Update
Account-New Username","Info",
"Business");bool
return true
return "new username"
return "Username Not Found"

5.6 Disable User Database Error



5.7 Invalid Claims

Preconditions:

User has hacked into UM without logging into admin account
Token must not be generated by our private key or claims must be for regular Member