UM Low Level:BitOHealth

The 6 Bits:
Angel Cueva
Dante Secundino
Emily Sahyoun
Jacob Munoz
Rami Iskender
Sebastian Vasquez (Team Leader)
12/4/2021

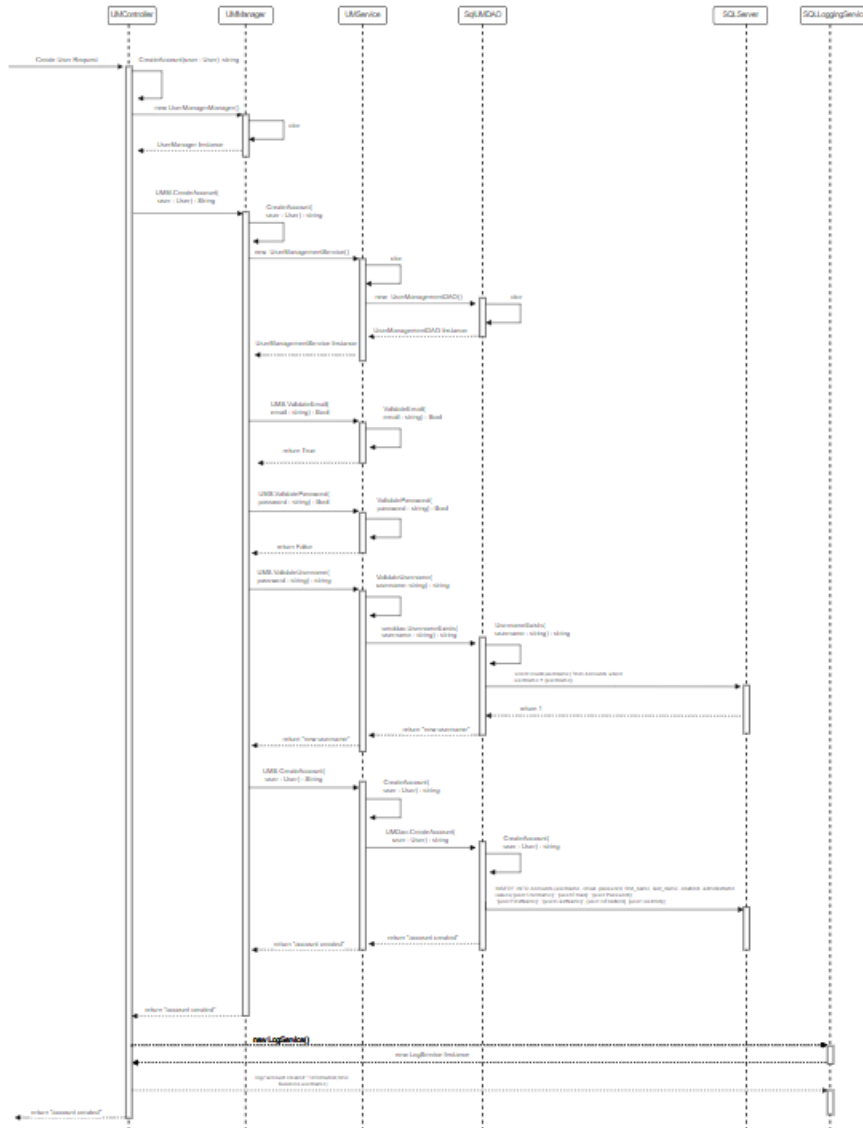# 1 Create User

1.1 Successful User Creation

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

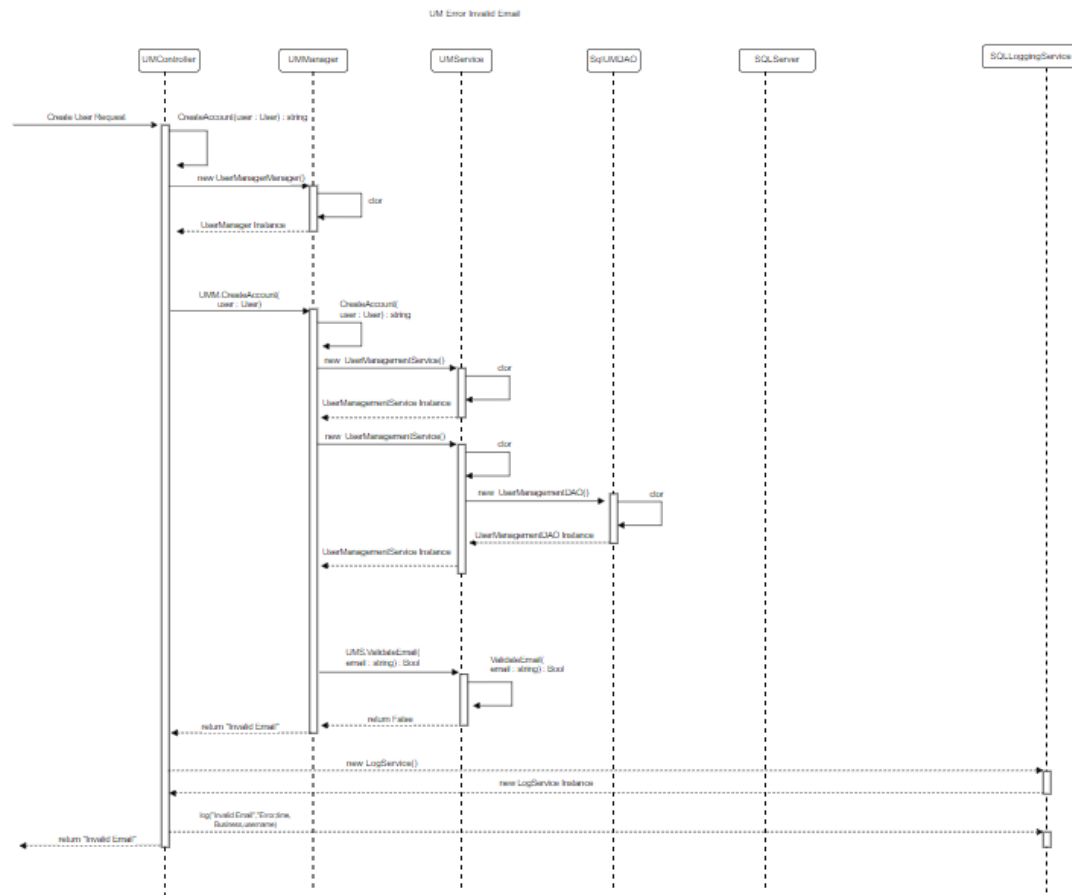## 1.2 Create Account-Invalid Email

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

UM Error Invalid Email

1.3 Create Account-Invalid Password

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

UM Error Invalid Password

1.4 Create Account-Invalid Username

Precondition:
Administrator performs the action
SQL Server Data Store is connected
Username Input is Invalid
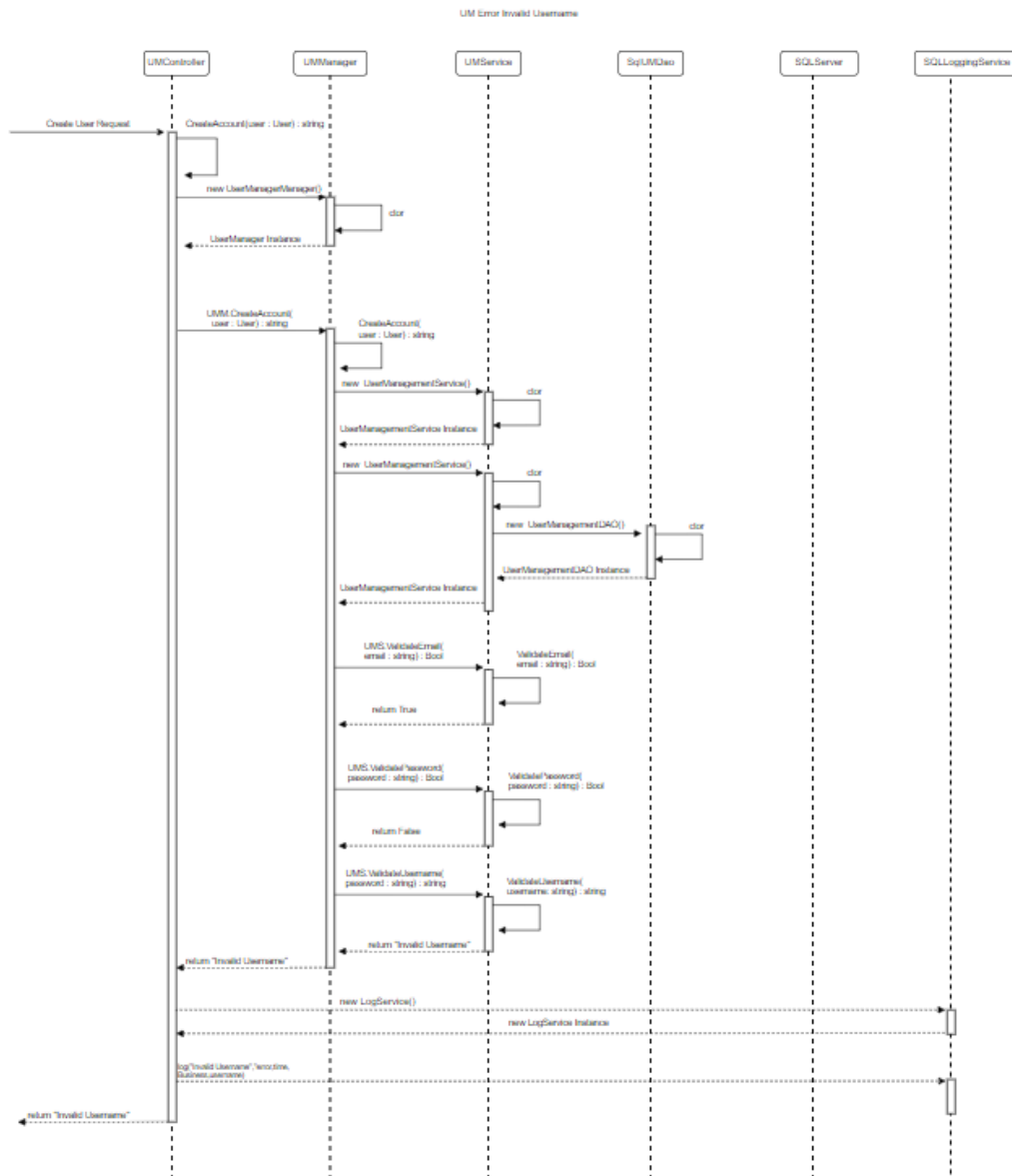
1.5 Create Account-Duplicate Username

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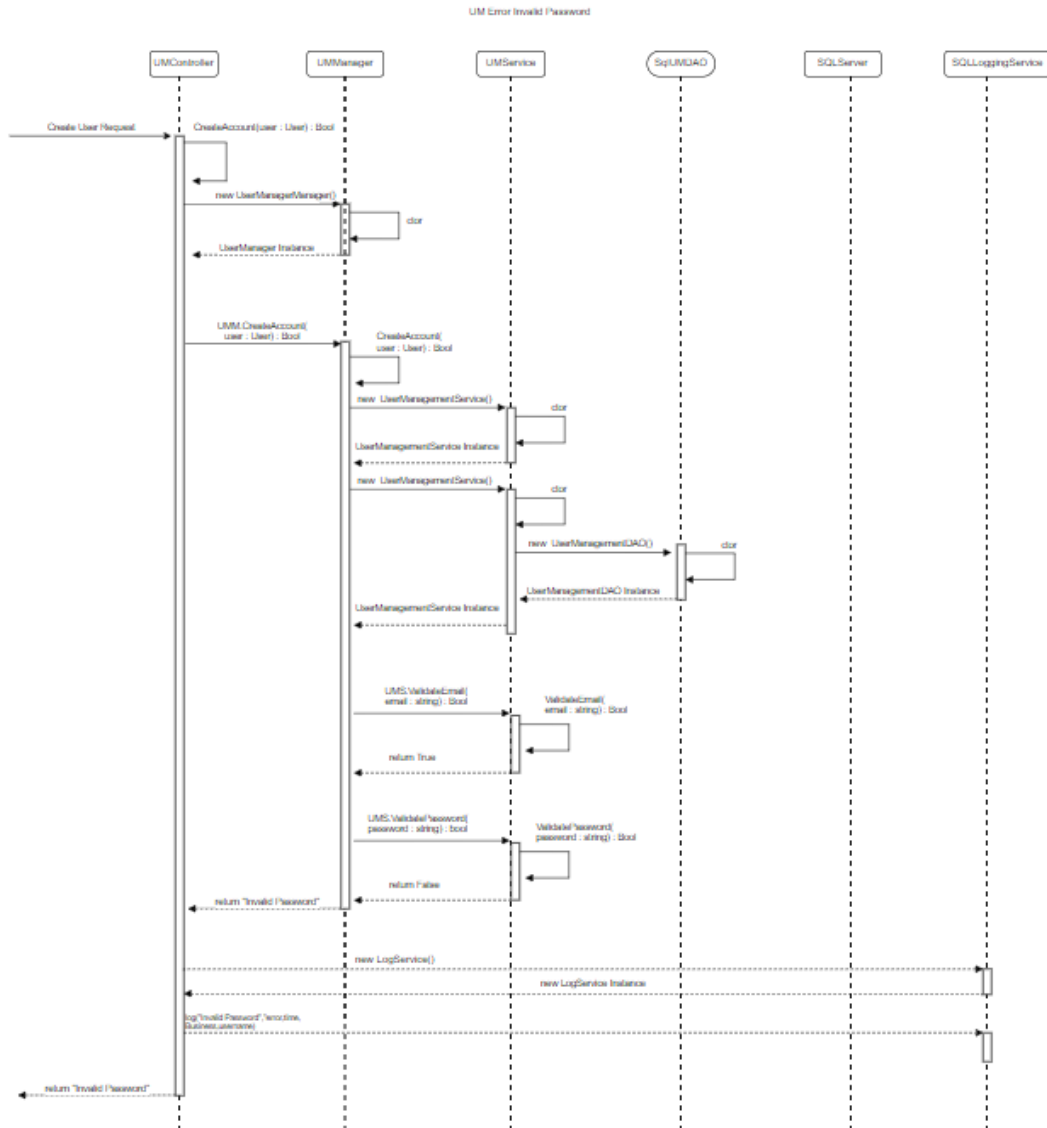Username Input is invalid, already exists

UMController	UMManager	UMService	SqlUMDAO	SQLServer	SQLLoggingService
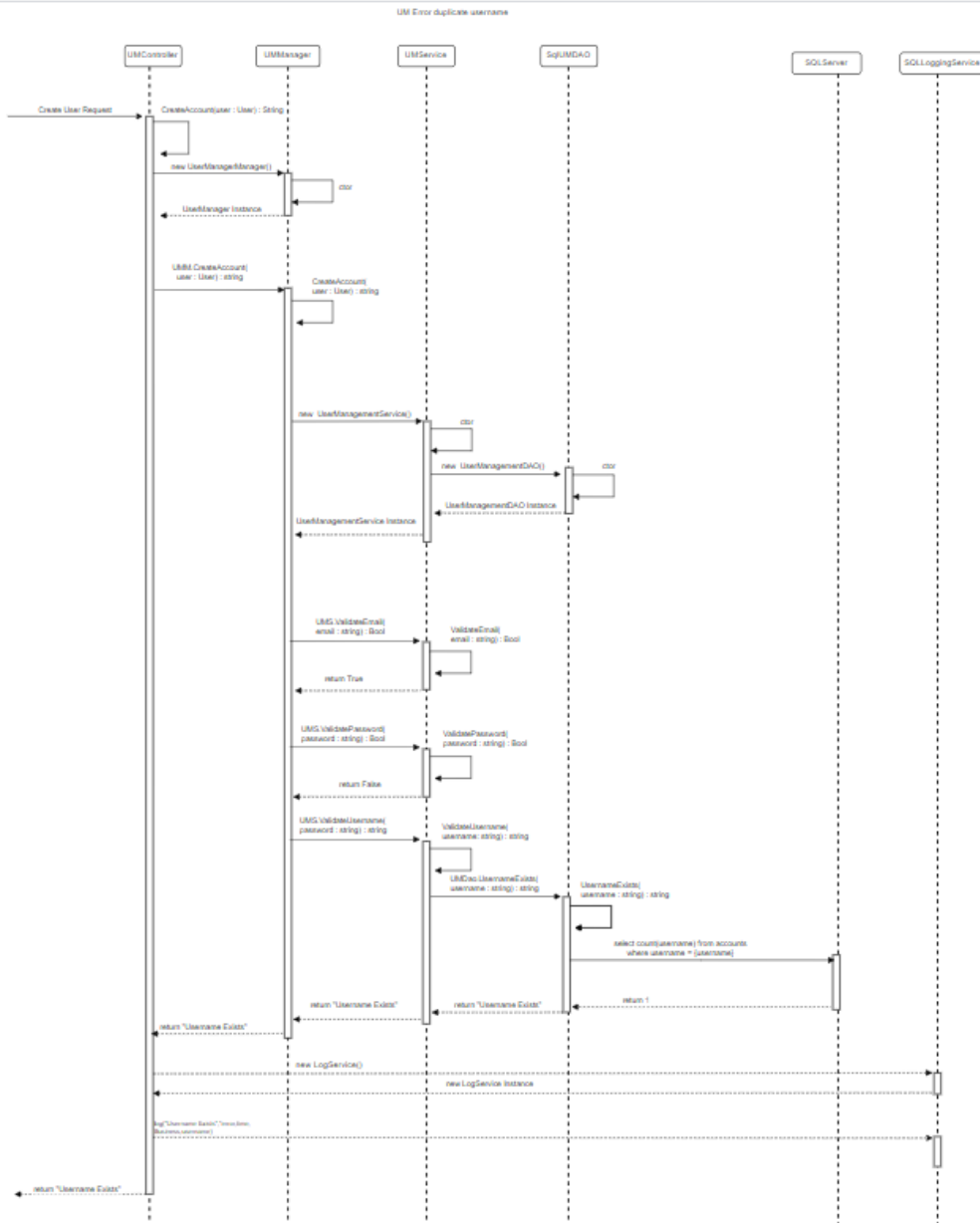
1.6 Create Account- Database Error

Precondition:
Administrator performs the action.
Database is offline.

2 Delete User

2.1   Successful Delete User

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

UM
DeleteAccount

UMController | UMManager | UMService | SqlUMDAO | SQL Server | SQLLoggingService

Delete User Request

DeleteAccount(username : s) string

new UserManagerManager()
ctor
UserManager Instance

UMM.DeleteAccount(username: String) string
DeleteAccount(username: String) : string

new UserManagementService()
ctor
new UserManagementDAO()
ctor
UserManagementDAO Instance
UserManagementService Instance

UMS.ValidateUsername(username: String) : string
ValidateUsername (username:String) string

UMDao.UsernameExists (username : string) string
select count(username) from Accounts where username = {username}

return 1
return "username exists"
return "username exists"

UMS.DeleteAccount(username: String) : string
UMS.DeleteAccount(username: String) : string

UMDao.DeleteAccount(user : User) string
delete from ACCOUNTS where username={username}

return "account Deleted"
return "account Deleted"
return "account Deleted"

new LogService()
new LogService Instance

Log(username,"account deleted", "information", "Business") bool
return "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

UM
DeleteAccount
-no username
found

UMController | UMManager | UMService | SqlUMDAO | SQLLogService | SQLService | SQLLoggingService

Delete User Request

DeleteAccount(username : s) string

new UserManagerManager()

ctor

UserManager Instance

UMM.DeleteAccount(username: String) string

DeleteAccount(username: String) : string

new UserManagementService()

ctor

new SqlUMDAO()

ctor

SqlUMDAO Instance

UserManagementService Instance

UMS.ValidateUsername(username: String) : string

ValidateUsername(username:String) string

SqlUMDAO UsernameExists(username : string) string

select count(username) from Accounts where username = {username}

return 0

return "new username"

return "new username"

return "no username found"

new LogService()

new LogService Instance

Log(username,"no username found", "Error", "Business") bool
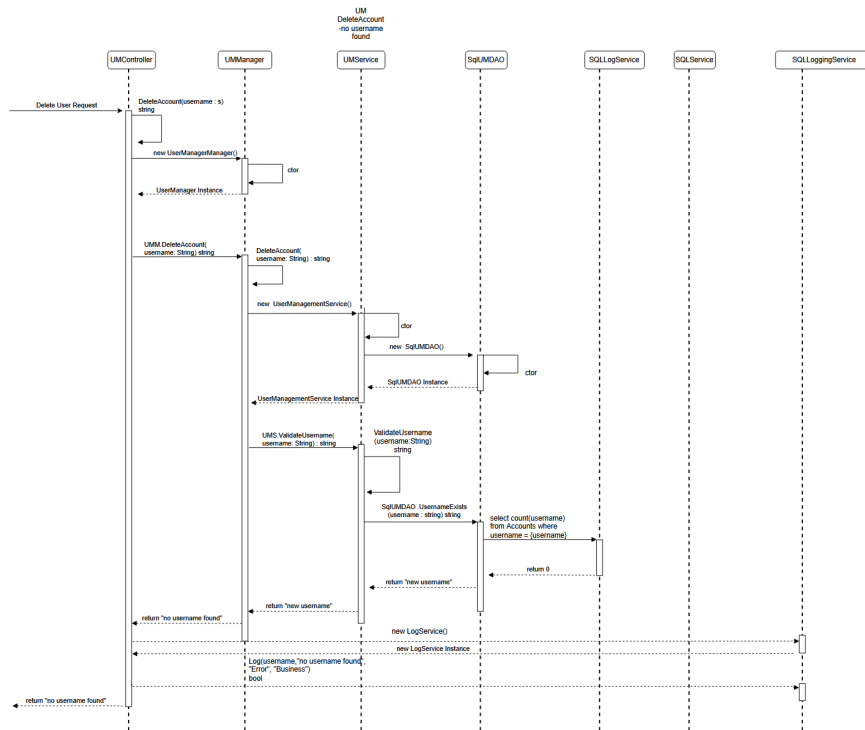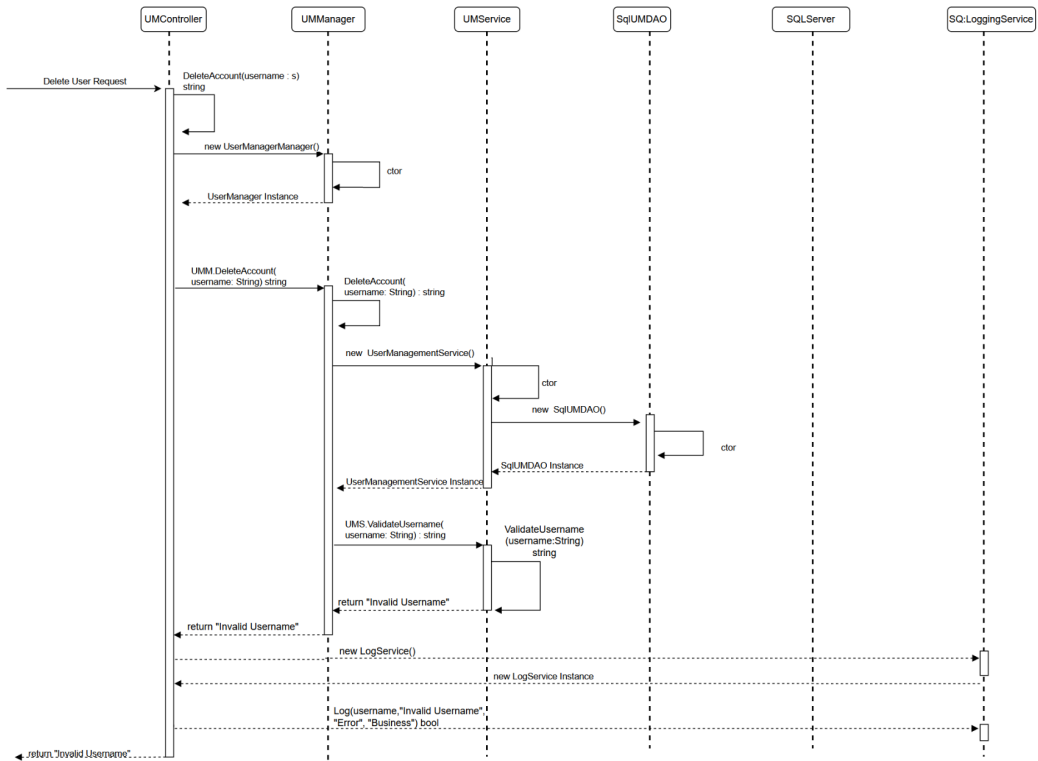
return "no username found"

2.3 Delete User- invalid username

Precondition:

Administrator performs the action

SQL Server Data Store connection error

Username provided not following Business requirements

UMController   **UMManager**   **UMService**   **SqlUMDAO**   **SQLServer**   **SQ:LoggingService**

Delete User Request

DeleteAccount(username : s) string

new UserManagerManager()

ctor

UserManager Instance

UMM DeleteAccount(username: String) string

DeleteAccount(username: String) : string

new UserManagementService()

ctor

new SqlUMDAO()

ctor

SqlUMDAO Instance

UserManagementService Instance

UMS ValidateUsername(username: String) : string

ValidateUsername(username:String) string

return "Invalid Username"

return "Invalid Username"

new LogService()

new LogService Instance

Log(username,"Invalid Username", "Error", "Business") bool
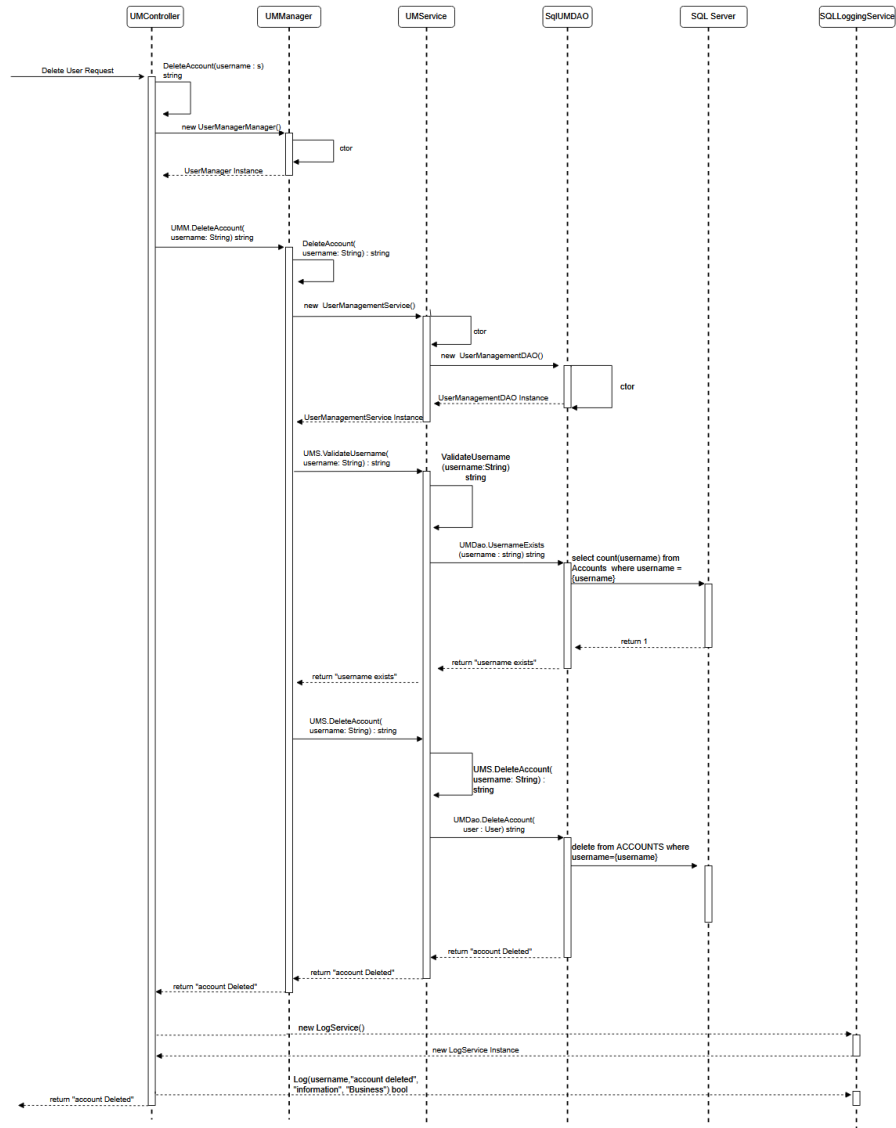
return "Invalid Username"

2.4 Delete User- Database Error

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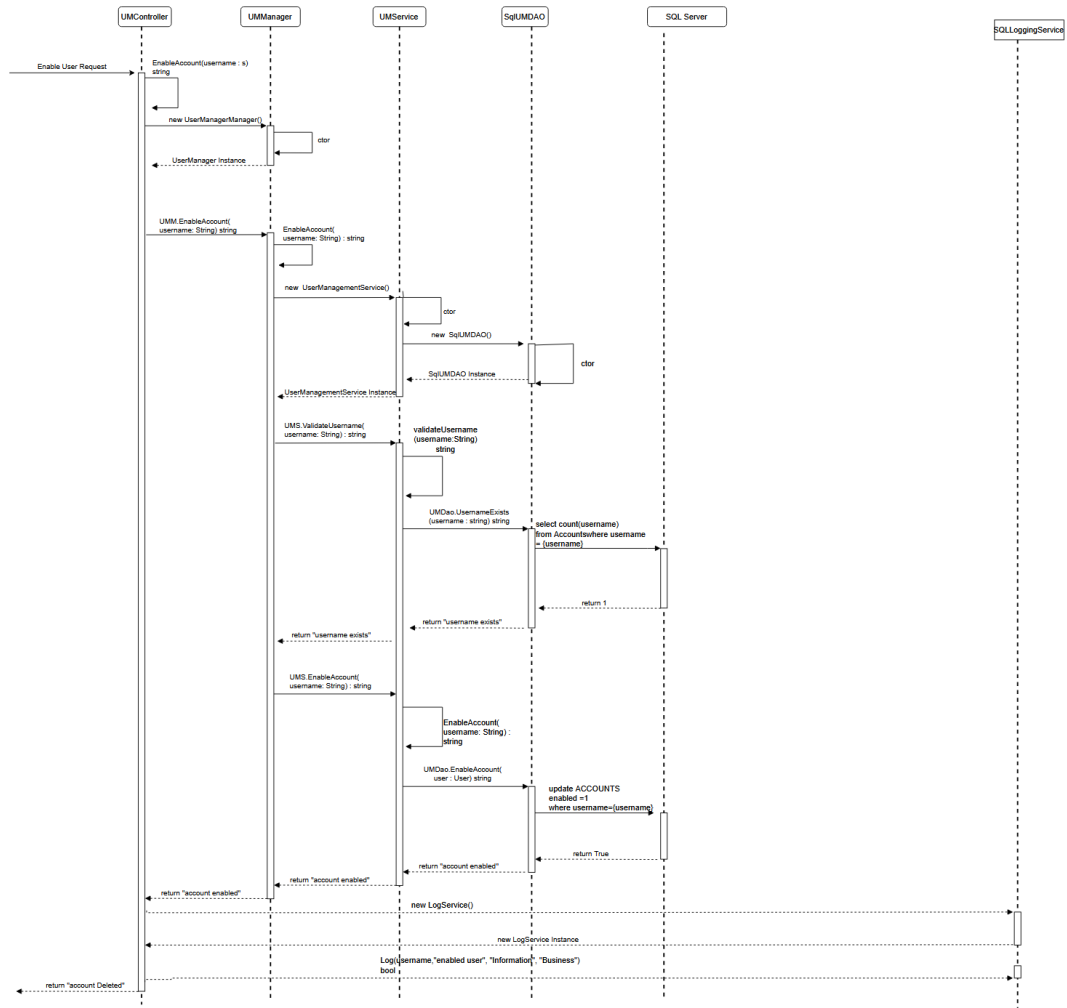Data Store is connected
Input Is valid

3 Enable Account

3.1 Enable User Successful

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

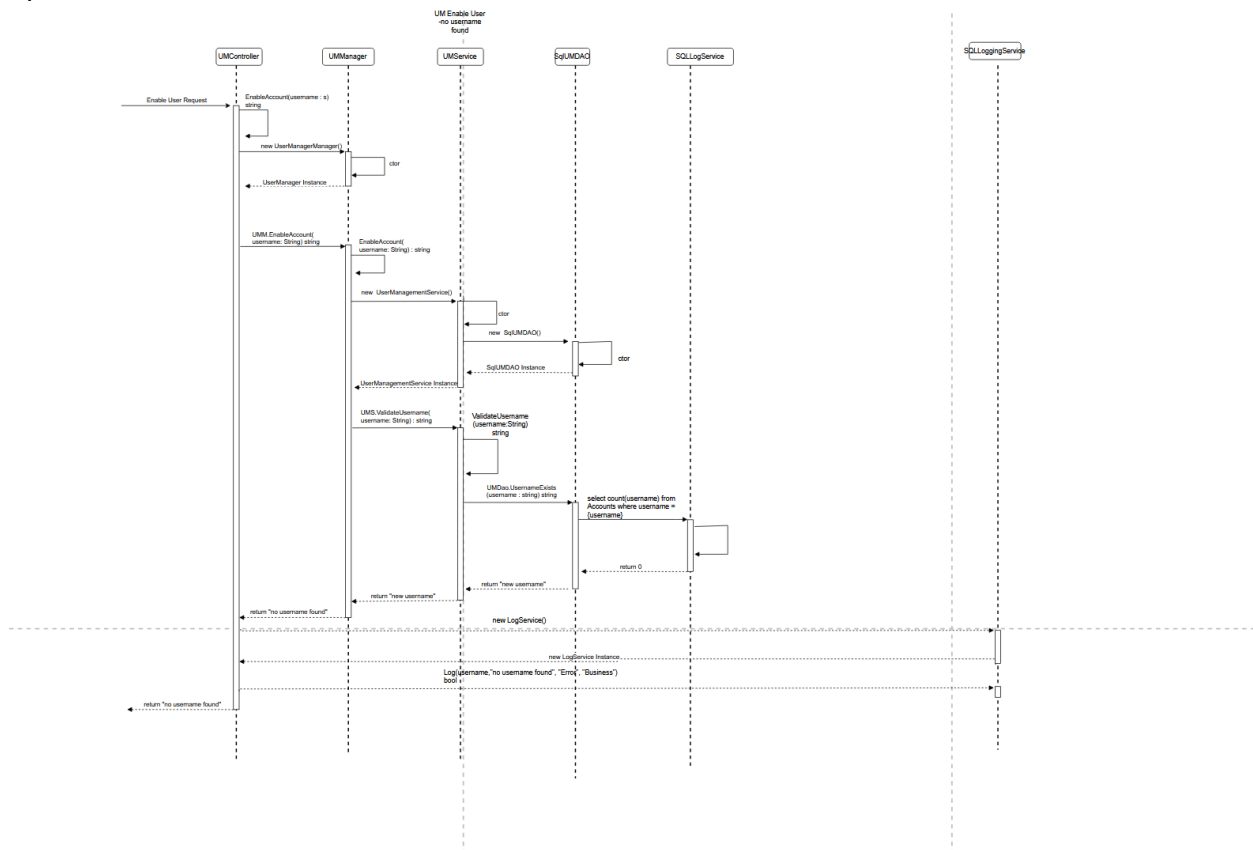Sequence diagram participants: UMController, UMManager, UMService, SqlUMDAO, SQL Server, SQLLoggingService

- Enable User Request
- EnableAccount(username : s) string
- new UserManagerManager()
- ctor
- UserManager Instance
- UMM.EnableAccount(username: String) string
- EnableAccount(username: String) : string
- new UserManagementService()
- ctor
- new SqlUMDAO()
- ctor
- SqlUMDAO Instance
- UserManagementService Instance
- UMS.ValidateUsername(username: String) : string
- validateUsername(username:String) string
- UMDao.UsernameExists(username : string) string
- select count(username) from Accounts where username = (username)
- return 1
- return "username exists"
- return "username exists"
- UMS.EnableAccount(username: String) : string
- EnableAccount(username: String) : string
- UMDao.EnableAccount(user : User) string
- update ACCOUNTS enabled =1 where username=(username)
- return True
- return "account enabled"
- return "account enabled"
- return "account enabled"
- new LogService()
- new LogService Instance
- Log(username,"enabled user", "Information", "Business") bool
- return "account Deleted"

3.2 Enable Username Invalid

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

UMController	UMManager	UMService	SqlUMDAO	SQL Server	SQLLoggingService

Enable User Request

DeleteAccount(username : s) string

new UserManagerManager()

ctor

UserManager Instance

UMM.EnableAccount(
username: String) string

EnableAccount(
username: String) : string

new UserManagementService()

ctor

new UserManagementDAO()

UserManagementDAO Instance

ctor

UserManagementService Instance

UMS.ValidateUsername(
username: String) : string

ValidateUsername
(username:String)
string

return "Invalid Username"

return "Invalid Username"

new LogService()

new LogService Instance

Log((username,"Invalid Username ", "Error", "Business")
bool

return "Invalid Username"

3.3 Enable User No Username Found

Precondition:

Administrator performs the action
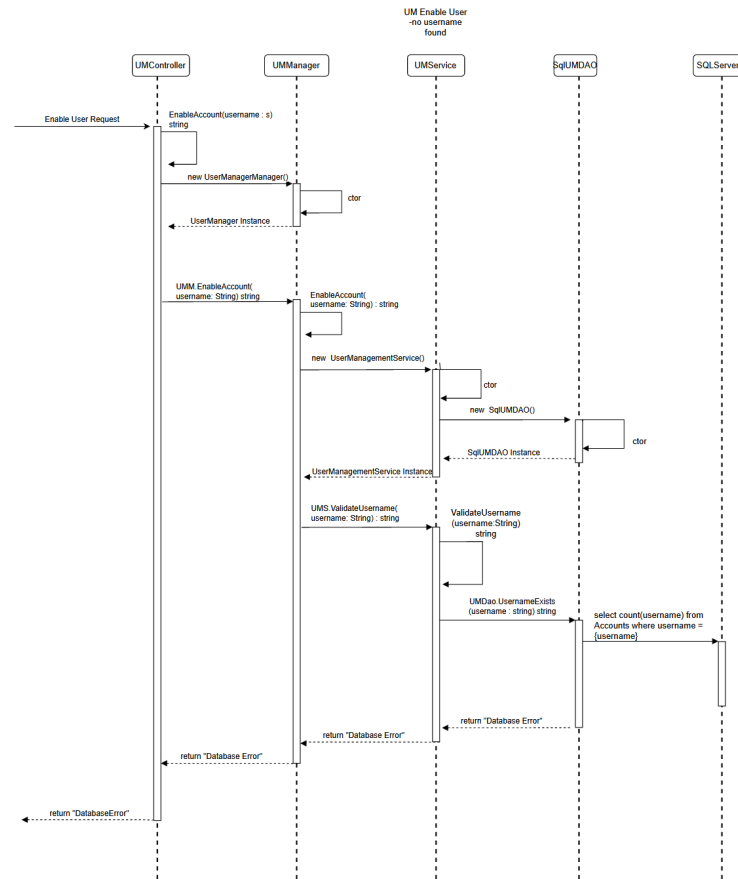
Data Store is connected

Inputs are not in database

UM Enable User
-no username
found

UMController | UMManager | UMService | SqlUMDAO | SQLLogService | SQLLoggingService

Enable User Request

EnableAccount(username : s) string

new UserManagerManager()

ctor

UserManager Instance

UMM.EnableAccount(username: String) string

EnableAccount(username: String) : string

new UserManagementService()

ctor

new SqlUMDAO()

ctor

SqlUMDAO Instance

UserManagementService Instance

UMS.ValidateUsername(username: String) : string

ValidateUsername(username:String) string

UMDao.UsernameExists(username : string) string

select count(username) from Accounts where username = {username}

return 0

return "new username"

return "new username"

return "no username found"

new LogService()

new LogService Instance

Log(username,"no username found", "Error", "Business") bool s

return "no username found"

3.4 Enable User Database Error

Precondition:
Administrator performs the action
Data Store is improperly connected

Inputs are valid

4 Disable User

4.1 Disable User Successful

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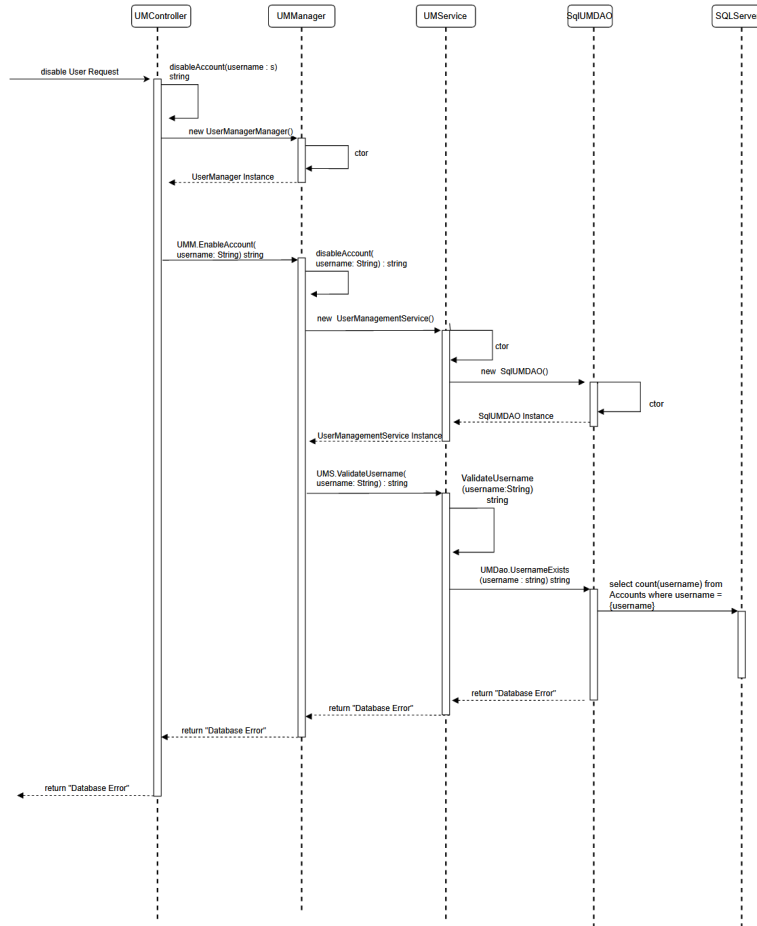Inputs are valid

4.2 Disable User Database Error

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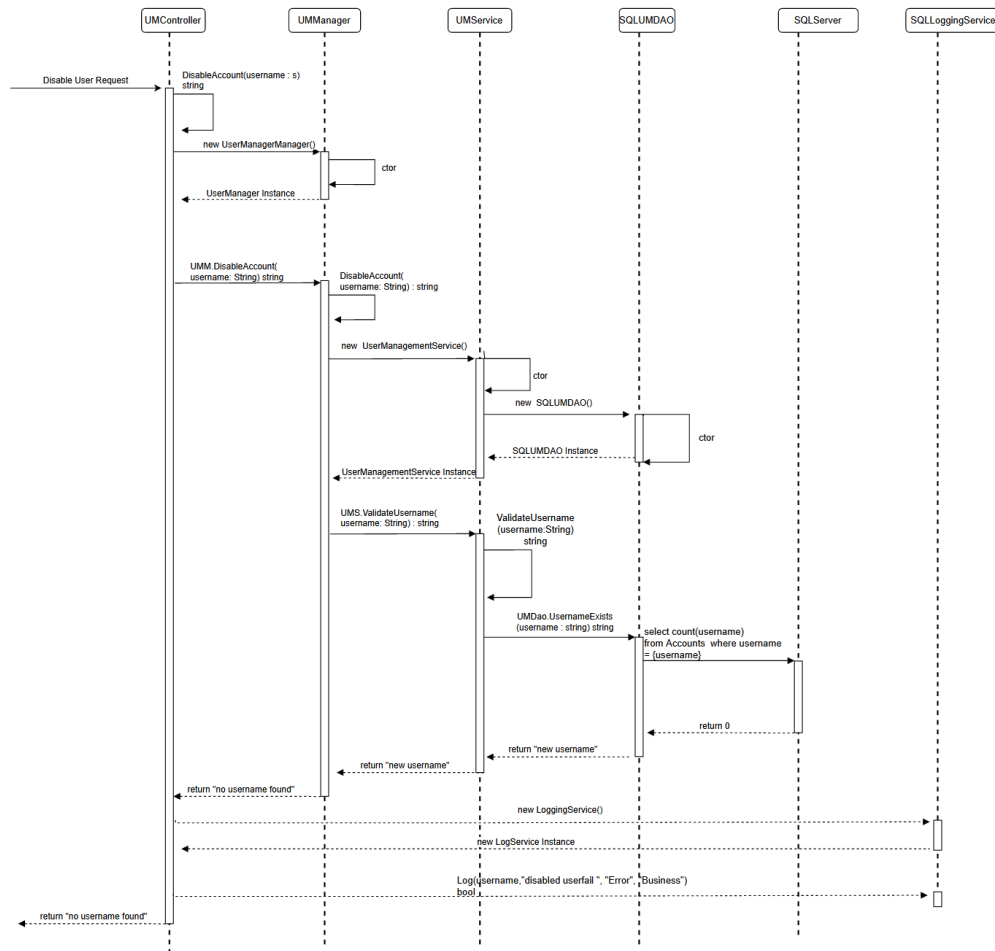Data Store is improperly connected
Inputs are invalid

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