Edit User Record

Participants: User, Controller.cs, RDSFactory, UserAccount, SystemAccountManager, AccountService, SqlDAO, SqlServer(MariaDB)

- User → Controller.cs: Request to edit user by a different user
- Controller.cs → RDSFactory: factory = new RDSFactory()
- RDSFactory: Constructor()
- RDSFactory → Controller.cs: return RDSFactory
- Controller.cs → SqlDAO: IDataSource factory.GetDataSource(string DbName, string connInfo)
- SqlDAO: Constructor(string connInfo)

Data Store Timed Out
- SqlDAO: Catch SqlException
- SqlDAO → Controller.cs: return "Data Source Timed Out"

Data Store Online
- SqlDAO → Controller.cs: return SqlDAO
- Controller.cs → SystemAccountManager: manager = new SystemAccountManager(IDataSource conn)
- SystemAccountManager: Constructor(IDataSource conn)
- SystemAccountManager → Controller.cs: New SystemAccountManager
- Controller.cs → UserAccount: new UserAccount(string username, string password, string email, DateTime RegistrationTimeStamp)
- UserAccount: string username, string password, string email, DateTime RegistrationTimeStamp
- UserAccount → Controller.cs: New UserAccount
- Controller.cs → SystemAccountManager: manager.UpdateUserRecord(user:UserAccount, ActiveUser:user, UpdateObj: userUpdate)
- SystemAccountManager → SqlDAO: bool isAdmin(user:UserAccount)
- SqlDAO: SqlRead confirmAdmin()
- SqlServer(MariaDB): Select * from Admins where Username = @UserName and Password = @Pwd

Unauthorized User
- SqlServer(MariaDB) → SqlDAO: SqlException
- SqlDAO: Catch SqlException
- SqlDAO → SystemAccountManager: return false
- SystemAccountManager → Controller.cs: return "Access Denied: Unauthorized"
- Controller.cs → User: return "Access Denied: Unauthorized"

Unauthorized User
- SqlServer(MariaDB) → SqlDAO: return false
- SqlDAO → SystemAccountManager: return false
- SystemAccountManager → Controller.cs: return "Access Denied: Unauthorized"
- Controller.cs → User: return "Access Denied: Unauthorized"

Authorized User
- SqlServer(MariaDB) → SqlDAO: return true
- SqlDAO → SystemAccountManager: return true
- SystemAccountManager: isInputValid(user: UserAccount)

Invalid Information
- SystemAccountManager → Controller.cs: return "Invalid input"
- Controller.cs → User: return "Invalid input"

Valid Information
- SystemAccountManager → AccountService: AccountService.CreateUserRecord(user:UserAccount, ActiveUser:user, UpdateObj: userUpdate)
- AccountService → SqlDAO: SqlDAO.CreateUserRecord(user:UserAccount, ActiveUser:user, UpdateObj: userUpdate)
- SqlDAO → SqlServer(MariaDB): isUser(user:UserAccount)
- SqlServer(MariaDB): Select * From Users where Username = @UserName OR email = @Email

data store timed out
- SqlServer(MariaDB) → SqlDAO: SqlException
- SqlDAO: Catch SqlException
- SqlDAO → AccountService: return "Database timed out"
- AccountService → SystemAccountManager: return "Database timed out"
- SystemAccountManager → Controller.cs: return "Database timed out"
- Controller.cs → User: return "Database timed out"

data store is online

User does exist
- return True

SQL insert
user credentials

return 1

return "User information updated successfully"

return "User information updated successfully"

return "User information updated successfully"

return "User information
updated successfully"

User does not exist

return False

SQL insert
user credentials

return 1

return "User is now registered"

return "User is now registered"

return "User is now registered"

return
"User is now registered"