User Management Low Level Design - SafeRide

Team: #0000FF Thunder
An Ta
Andy Lee
Leon Chen
Orion Tang (Team Leader)

Date Issued:
December 13th, 2021
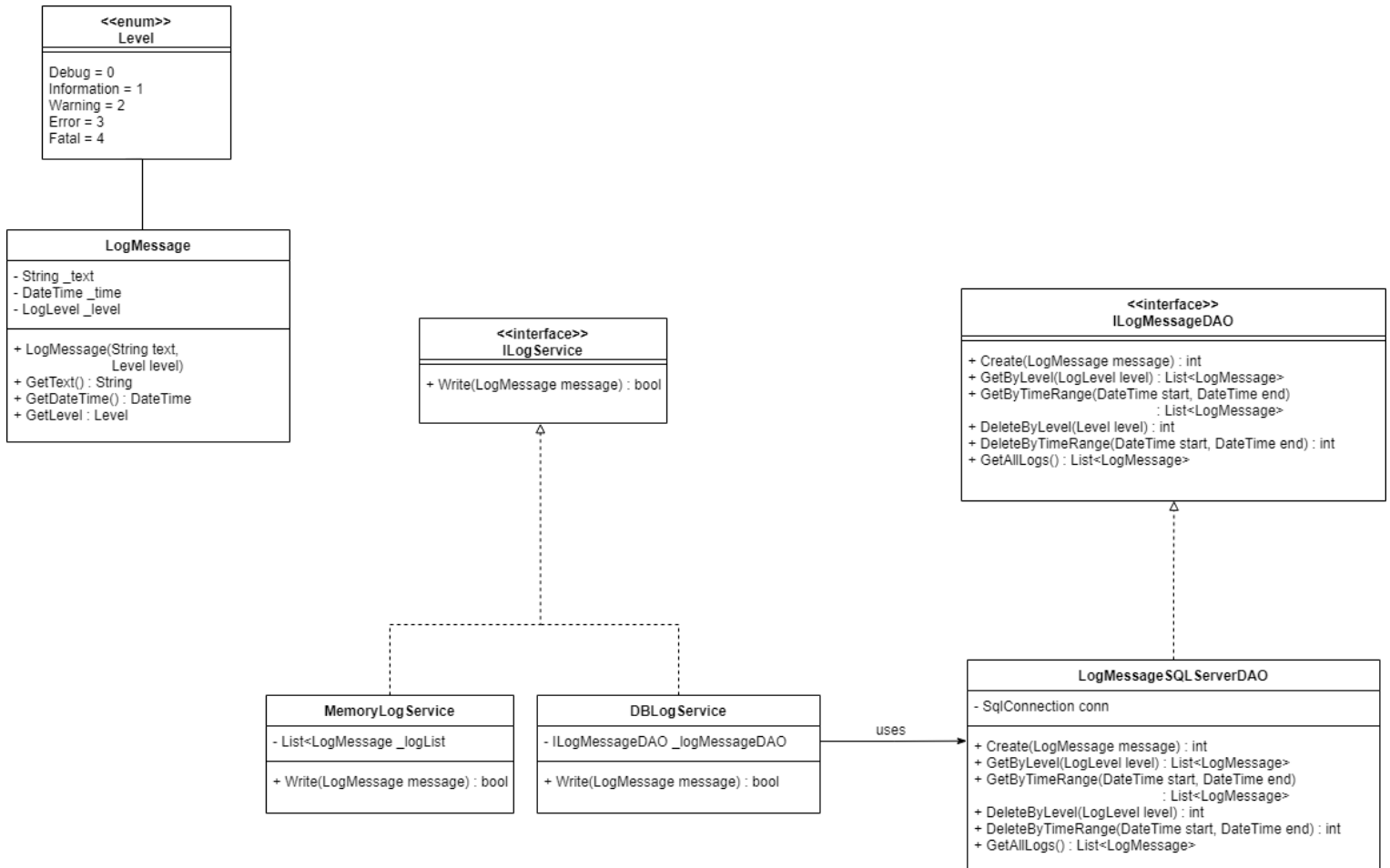
User Management Overview

Purpose
User management represents logic that deals with requirements that deal with user management. We have a um manager that represents code that deals with user security and operations that can be used at a higher level such as UpdateUser and UpdateUsers. It uses IUserDAO which is an interface that can be implemented by data store technologies that is used for User CRUD operations + enable and disable. The User model represents a real user which can be mapped to database objects and fields.

Dependencies
- Must have a database set up with a valid connection string and tables for "Users" exists
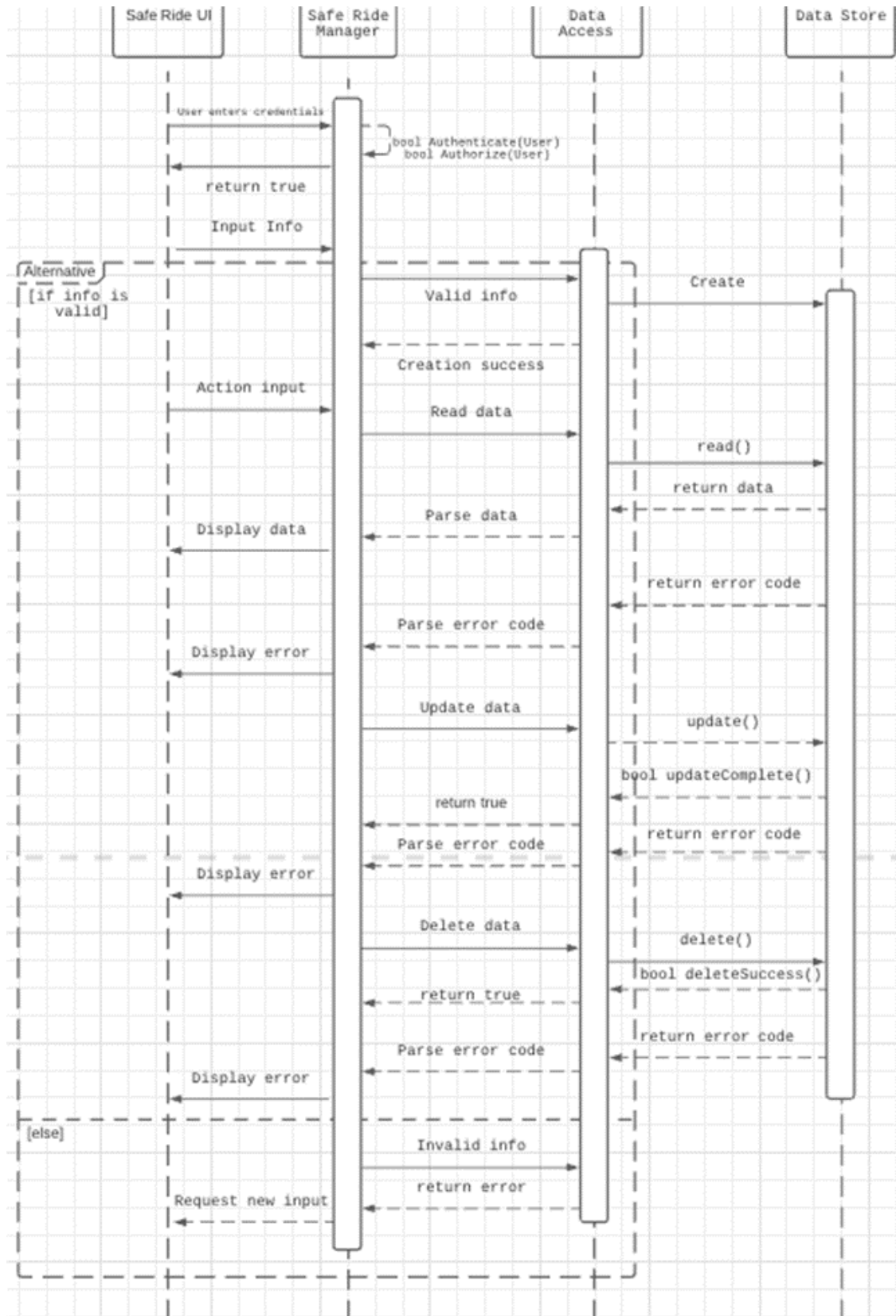   - SQL Server 2019 Express will be used for this application

User Management - UML

UML diagram represents relationship and high level methods and fields that could be within the class itself.

```
┌─────────────────────────┐
│        <<enum>>         │
│         Level           │
├─────────────────────────┤
│ Debug = 0               │
│ Information = 1         │
│ Warning = 2            │
│ Error = 3              │
│ Fatal = 4              │
└─────────────────────────┘
```

```
┌─────────────────────────┐
│       LogMessage        │
├─────────────────────────┤
│ - String _text          │
│ - DateTime _time        │
│ - LogLevel _level       │
├─────────────────────────┤
│ + LogMessage(String text,│
│            Level level) │
│ + GetText() : String    │
│ + GetDateTime() : DateTime│
│ + GetLevel : Level      │
└─────────────────────────┘
```

```
┌─────────────────────────────────┐
│          <<interface>>          │
│           ILogService           │
├─────────────────────────────────┤
│ + Write(LogMessage message) : bool│
└─────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────┐
│                 <<interface>>                    │
│                ILogMessageDAO                     │
├─────────────────────────────────────────────────┤
│ + Create(LogMessage message) : int               │
│ + GetByLevel(LogLevel level) : List<LogMessage>  │
│ + GetByTimeRange(DateTime start, DateTime end)   │
│                      : List<LogMessage>          │
│ + DeleteByLevel(Level level) : int               │
│ + DeleteByTimeRange(DateTime start, DateTime end) : int│
│ + GetAllLogs() : List<LogMessage>                │
└─────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│        MemoryLogService         │
├─────────────────────────────────┤
│ - List<LogMessage _logList      │
├─────────────────────────────────┤
│ + Write(LogMessage message) : bool│
└─────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│          DBLogService           │
├─────────────────────────────────┤
│ - ILogMessageDAO _logMessageDAO │
├─────────────────────────────────┤
│ + Write(LogMessage message) : bool│
└─────────────────────────────────┘
```

uses

```
┌─────────────────────────────────────────────────┐
│            LogMessageSQLServerDAO                │
├─────────────────────────────────────────────────┤
│ - SqlConnection conn                             │
├─────────────────────────────────────────────────┤
│ + Create(LogMessage message) : int               │
│ + GetByLevel(LogLevel level) : List<LogMessage>  │
│ + GetByTimeRange(DateTime start, DateTime end)   │
│                          : List<LogMessage>      │
│ + DeleteByLevel(LogLevel level) : int            │
│ + DeleteByTimeRange(DateTime start, DateTime end) : int│
│ + GetAllLogs() : List<LogMessage>                │
└─────────────────────────────────────────────────┘
```

User Management - Sequence Diagram

Sequence diagram represents roughly the logic implemented to do CRUD operations to the database, as well as important operations such as authorization and authentication.

User Management - CRC Cards

CRC Cards represents the classes used in the User management process which is the model, dao interface and dao implementation. Manager not included.

User	
First name	
Last name	
ID	
Phone number	
request input	

Data Access Object Interface	
getAllUsers()	User
getUser()	
updateUser()	
deleteUser()	

Data Access Object Implementation	
List of Users	Data Access Object Interface
getUser()	
updateUser()	
deleteUser()	