HASKELL REPORT - COURSEWORK 2

Name - Mohammed Rayaan Hussain
ID - 221070112

STRUCTURE

The project source is split into 3 files -

- Main.hs
- UserActivity.hs
- Types.hs

Types.hs -

This module exports the **User** and **Message** types and their constructors. These types respectively define the types for the User object and the Message object.

The User type represents a user with the fields **username** and **receivedMessages** which contain the name of the user and a mutable list (**MVar**) which acts as a mailbox of all the messages received by that user.

The Message type represents a type with fields **fromUser** and **content** which represent the sender and the message content respectively.

The **Show** instance is provided for the **Message** type, allowing messages to be displayed in a readable format.

UserActivity.hs -

This module exports the **userActivity** and **createUser** functions which are responsible for user-related activities.

The **createUser** function creates a user with the given username and an empty mailbox (**MVar**)

The **sendMessage** function sends a message from one user to another by updating the recipient's mailbox.

The **userActivity** function simulates user activity by creating messages and sending them to random users with random time intervals.

Main.hs -

There is a loop which goes over a list of usernames, creates users and spawns corresponding user activity threads.

The **forkIO** function is used to create a new lightweight thread in Haskell. This represents a user's activity.

The **threadDelay** function is used to introduce a delay (sleep) in the execution of a thread.

The **forM_** function is used to iterate over the list of users.

The **readMVar** function is used to read the **MVar** variable which in this case is the contents of **receivedMessages** from the user.

DESIGN DECISIONS

Data Types -

For the **Message** type, the fields are kept simple as we need just the username of the recipient and the message content itself. Both fields are of string type.

For the **User** type, the **username** field is kept simple as it's just a unique name of the user. The **receivedMessages** field is set to **MVar [Message]** representing a mutable variable containing the list of messages received by the user.

Using an **MVar** for **receivedMessages** ensures thread-safety when dealing with concurrency. It allows multiple threads to modify and access the list of messages in a safe and synchronized manner.
It guarantees that only one thread can modify the list of messages at a time, preventing data races and inconsistencies. The threads can safely update the message list without interfering with one another.

Functions -

modifyMVar_ allows atomic modifications to the **MVar**, avoiding potential race conditions and fully committing to transactions or discarding them.

randomRIO is a function used to introduce random time intervals for message exchanges and random user selections. Introducing randomness adds realism to the social network as there is no predicted or predefined way of the order of events.

Separation of Concerns -

The project source is split into 3 files which have their own responsibility promoting separation of concerns. Modularization enhances code readability and maintainability. Each module encapsulates related functionality, making it easier to understand and modify.