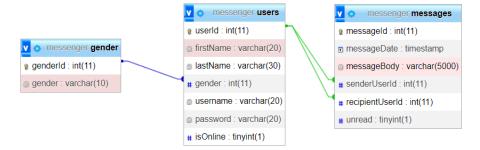
## Message exchange flow Client Server Events: -SQL query for username/password || register new -Send<REST API>: CredentialModel/ UserModel datauser -Login / Register If ok: -Update auth state -Update online users list (socket\_id:username) -Return<REST API>: Token--Return token -SQL query for all users -Send<REST API>: getAllUsers()--Route to home -Return users -Update users store -Return<REST API>: userModel[] arr--SQL query to messages of 2 users with -User click on chat partner, -Click on users list<REST API>: getUsersMessages(id1,id2)usernames(join) order by date -Update chat store -Return extended messages ·Return msg arr<REST API>: ExtendedMessage[]— (ExtendedMessages, activePartner) Chat using socket io -If recipient online: -Click on msg send btn <WS>: ExtendedMessage--User send Extendedmessage \*SQL query update messages(unread = false) -Update chat store with -Return posted message<WS>: ExtendedMessage with id-\*Emit the message to recipient(wait for ack? => received extendedMessage this will unsure that message delivered, but complicate code); -If recipient offline: \* SQL query update messages(unread = true) -Return posted extendedMessage with ID Project name: Date: 07-03-2023 Messanger Design: Drawing name/no Draw: Ver. 1.0 Approved:

## Database tables and relations



User Model: Credentials Model:

Message Model:

Extended Message Model:

-User send Extendedmessage -Update chat store with received extendedMessage

	Project name:  Messanger		Date:	07-03-2023	
			Design:		
	Drawing name/no	Ver.	Draw:		
		1.0	Approved:		