Marcus Leong
Megana Chinalachaiagari
EECS 647
Project 1 Report

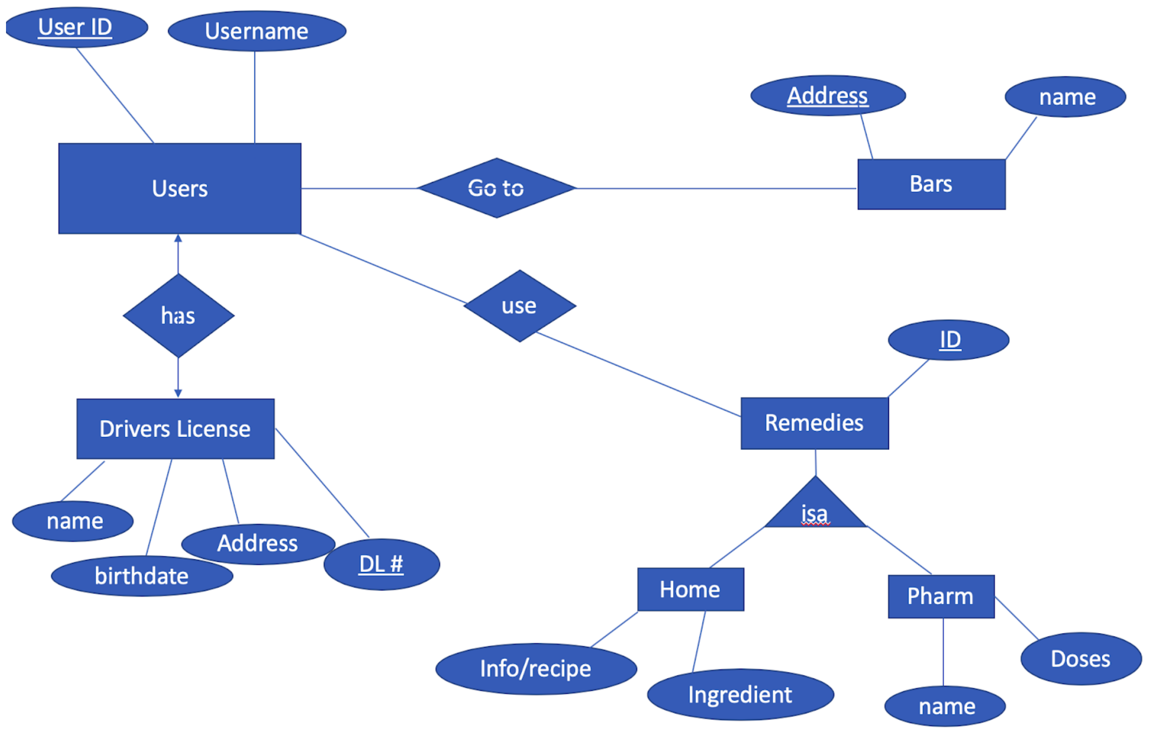
1. Introduction: what is your scenario (describe your mini-world)?

Our scenario focuses on a Bar Crawl type application. We will have the user create an account in which they will create a username, and thus be given a user ID as well (primary key). Each User will also scan into the system, a unique Driver's License. The Driver's license verifies the user's name as well as their address and birthdate. The primary key to be used will be a Driver License number. These multiple users will be able to go to multiple bars listed within the application. Each bar will be identified by its address, but also will store a name for the bar. Additionally, each user may need information on how to treat hangovers. As a result, they can submit types of remedies or view these remedies posted by other users. Remedies are broken up as either home remedies or Over the Counter "pharmacy necessary" remedies. For home remedies, information stored will include the ingredients needed and the recipe/steps to make the remedy. For OTC/Pharmacy remedies, the name of the drug and its doses will be made available.

2. Requirements analysis (be brief): data and operations on data.

Bar Crawl (people go to bars, get drinks, have hangovers, hangovers have remedies, people use remedies, populate popular bars and remedies)

3. Conceptual design: ER-diagram and constrains



4. Logical design: convert your ER-diagram to relational schemas. Normalize your relations, if needed.

User(uuid: int,username: string)
DriversLicense(number: string, name: string, address: string, birthdate: date)
Bars(address: string, name: string)
HomeRemedies(id: int, info: string, ingredients: string)
PharmaceuticalRemedies(id: int, doses: string, name: string)
hasADriversLicense(DLnumber, username)
goesToBar(uuid, barAddress)
usesRemedies(uuid, id)