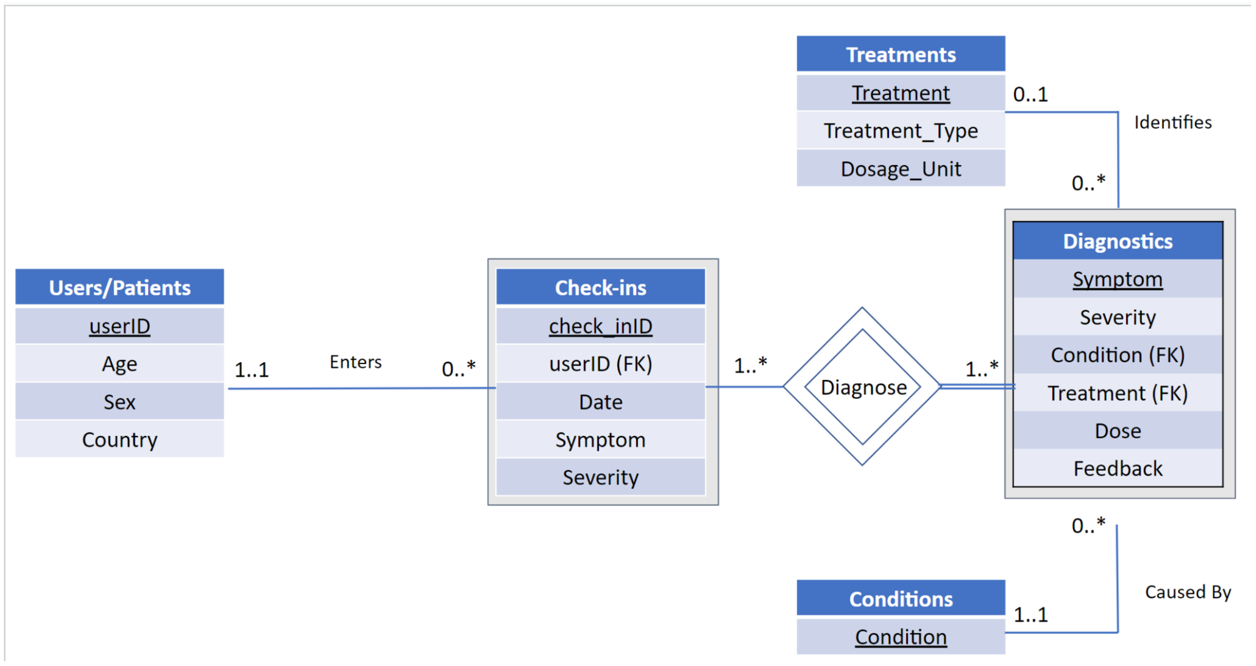
ER Diagram



Assumptions/Description for Each Entity

- Users
  - Users contains the user's personal information such as their age, sex, and country as well as their User ID which allows us to uniquely identify each user.
- Check-in
  - Check-in allows us to store the user's data entry whether they are searching for a treatment based on their symptoms and severity or if they are providing feedback on a treatment that they had taken.
  - A check-in is a weak entity specific to the user and allows us to track each data entry based on a unique check-in ID (for a certain user) and the user's ID.
- Diagnostic
  - Diagnostic essentially keeps all data based off the user's symptoms and severity. In addition to the possible conditions and treatments, the Diagnostic also contains the feedback of users that have already went through the possible treatments. The Diagnostic table is a weak entity, and a check-in id is required to uniquely identify one diagnostic.
- Treatments
  - Treatments contains the information of possible treatments based on the symptoms as well as the severity. This includes the treatment type as well as the dosage unit if applicable. Treatments are uniquely identified by the treatment name itself.

- Conditions
    - Conditions contains information about the condition itself. It is uniquely identified by the condition name itself.

Cardinality/Description for Each Relationship

Users Enters Check-Ins: 1..1 - 0..* (1-many)

A user can enter as many check-ins as they want.

Diagnostics Diagnoses Check-Ins: 1..* - 1..* (many-many)

Check-Ins can have many Diagnostics, and diagnostics can diagnose many check-ins.

Diagnostics Caused By Conditions: 0..* - 1..1 (many - 1)

Diagnostics are always caused by exactly 1 condition. A condition causes diagnostics, but there could be a condition not linked to an existing diagnostic for any of the users.

Diagnostics Identifies Treatments: 0..* - 0..1 (many - 1)

Diagnostics always identify a treatment, but there could be no treatment linked to the symptom. A treatment is linked to many diagnostics as one treatment can treat various symptoms.

Sample Workflow

1. We would ask the user to register and fill in their personal information, and the user is assigned a personal ID. After registration, the user could start an input everytime they feel ill or have an injury. We call every input a “check-in”. One check-in comes from exactly one user.

2. Imagine going to the doctors. The doctor could tell you’re fine and just record this visit, or tell you what is wrong and give you different medicines, shots or therapies. In our database we store every unique condition+treatment combination from a single visit as one row (One check-in leads to at least one diagnostic).

3. The same diagnostic could be given to many different check-ins. For example, people

who strain their arm muscles or leg muscles could have the same medicine prescription by the doctor.

4. One diagnostic would suggest the condition the user may have, but there are cases where no treatment is needed.

Logical Design

```
Users(
userID: VARCHAR(255) [PK],
Age: INT,
Sex: VARCHAR(100),
Country: VARCHAR(255)
);

Check-ins(
check-inID: INT [PK],
userID: VARCHAR(255) [FK to Users.userID],
Date: VARCHAR(10),
Symptom: VARCHAR(255),
Severity: INT
);

Treatments(
Treatment: VARCHAR(255) [PK],
Treatment_Type: VARCHAR(255),
Dosage_Unit: VARCHAR(50)
);

Conditions(
Condition: VARCHAR(255) [PK]
);

Enters(
```

userID: VARCHAR(255) [PK] [FK to Users.userID],
check_inID INT [PK] [FK to Check-ins.check-inID]
);

Diagnose(
check_inID [PK] [FK to Check-ins.check_inID],
Symptom [PK] [FK to Diagnostics.Symptom]
);

Identifies(
Treatment [PK] [FK to Treatments.Treatment],
Symptom [PK] [FK to Diagnostics.Symptom]
);

Caused By(
Symptom [PK] [FK to Diagnostics.Symptom],
Condition [PK] [FK to Conditions.Condition]
);

Fixes to Stage 1

We made it clear that we are making this a databases project rather than a website project. There seemed to be confusion when we said that we could use APIs and different libraries, when we meant to have that as a creative component for extra credit, so we made sure to emphasize that part is just an optional creative component that we came up with. Additionally, we made sure to emphasize how we are using the database provided to us from Kaggle to make sure that we are able to insert new records, update records, or delete records.

Our project will serve as a treatment recommender where the user will input their personalized information such as their ID, age, sex, and country as well as the symptoms they are feeling. Our application will provide the user with info from the Chronic Illnesses database that contains past treatments that users used to combat these symptoms as well as possible conditions that could be the cause of these symptoms. Each user will be able to add their records to the Chronic Illnesses database that we are using, as well as update their record to add feedback details or delete their record.