UserGlicodeLevel

User_id:
Score:
Value:
Captured_time:

UserSleepData
User_id: :integer
Date: :date
Sleep_start_time: :datetime
End_time: :datetime

UserTargets
User_id:
Target_value:
Target_type:
Enum: {
 Glucose_level: 1
 Sleep_score: 2

…

}
Frequency: {
 Weekly:
Hourly:
daily:
}

```
Def calculate_current_status
    data = []

    If sleep_score?
          If daily?
            UserSleepData.where(user_id: user_id).order(date:
:asc).pluck(:date,:sleep_start_time,:sleep_end_time).map {|x|
Duration = (x.sleep_end_time.to_i - x.sleep_start_time.to_i)/3600
{date: x.date, sleep_duration: Duration, is_target_completed:  Duration > Target_value}
          end
    end
end

Def fetch_metrics_details
```

```
  goals= UserTarget.where(user_id: current_user.id).limit(20).offset() ;nil

End
```

Redis