

Database Rows

$$\begin{aligned}
 UserRow &\triangleq [\\
 &\quad subscribed : \text{BOOLEAN} , \\
 &\quad \text{Forget canceled} \\
 &\quad inTrial : \text{BOOLEAN} , \\
 &\quad trialStartTime : \text{Nat}, \\
 &\quad billedForMonth : \text{Nat}, \\
 &\quad hasHadTrialOrSubscription : \text{BOOLEAN} , \\
 &\quad hasCancelled : \text{BOOLEAN} , \\
 &\quad cancelMonth : \text{Nat}, \\
 &\quad subscribeMonth : \text{Nat} \\
 &]
 \end{aligned}$$

API endpoints

$$\begin{aligned}
 StartSubscription(u) &\triangleq \\
 &\quad \text{Not subscribed} \\
 &\quad \wedge \wedge database.users[u].subscribed = \text{FALSE} \\
 &\quad \quad \wedge \vee database.users[u].inTrial = \text{FALSE} \\
 &\quad \quad \vee database.users[u].trialStartTime = month \\
 &\quad \wedge database' = \\
 &\quad \quad [database \text{ EXCEPT} \\
 &\quad \quad \quad !["users"][u].subscribed = \text{TRUE}, \\
 &\quad \quad \quad !["users"][u].hasHadTrialOrSubscription = \text{TRUE}, \\
 &\quad \quad \quad !["users"][u].hasCancelled = \text{FALSE}, \\
 &\quad \quad \quad !["users"][u].inTrial = \text{FALSE}, \\
 &\quad \quad \quad !["users"][u].subscribeMonth = month, \\
 &\quad \quad \quad !["billQueue"] = \\
 &\quad \quad \quad \quad Append(database.billQueue, \\
 &\quad \quad \quad \quad \quad [user \mapsto u, \\
 &\quad \quad \quad \quad \quad \quad fee \mapsto SubscriptionFee, \\
 &\quad \quad \quad \quad \quad \quad when \mapsto month]) \\
 &\quad \quad] \\
 &\quad \text{Observability required by stub} \\
 &\quad \wedge events' = Append(events, [type \mapsto "startsubscription", user \mapsto u]) \\
 &\quad \wedge \text{UNCHANGED } month
 \end{aligned}$$

$$\begin{aligned}
 CancelSubscription(u) &\triangleq \\
 &\quad \text{LET } updatedBillQueue \triangleq \\
 &\quad \quad \text{IF } \exists i \in 1 \dots Len(database.billQueue) : \\
 &\quad \quad \quad \wedge database.billQueue[i].user = u
 \end{aligned}$$

```

       $\wedge$  database.billQueue[i].fee = SubscriptionFee
    THEN database.billQueue
    ELSE Append(database.billQueue,
      [user  $\mapsto$  u, fee  $\mapsto$  SubscriptionFee, when  $\mapsto$  month])
  IN

  Subscribed
 $\wedge \vee$  database.users[u].subscribed = TRUE
   $\vee \wedge$  database.users[u].inTrial = TRUE
   $\wedge$  database.users[u].trialStartTime < month
 $\wedge$  database' =
  [database EXCEPT
    !["users"][u].subscribed = FALSE,
    !["users"][u].inTrial = FALSE,
    !["users"][u].hasCancelled = TRUE,
    !["users"][u].cancelMonth = month,

    Charge cancellation fee
    !["billQueue"] =
      Append(updatedBillQueue,
        [user  $\mapsto$  u,
          fee  $\mapsto$  CancellationFee,
          when  $\mapsto$  month + 1])
  ]

  Observability required by stub
 $\wedge$  events' = Append(events, [type  $\mapsto$  "cancelsubscription", user  $\mapsto$  u])
 $\wedge$  UNCHANGED  $\langle$ month $\rangle$ 

```

Recurring Operations

```

BillSubscribedUsers  $\triangleq$ 
 $\wedge \exists u \in USERS :$ 
  That is subscribed
 $\wedge \vee$  database.users[u].subscribed = TRUE
  Subscribed from a trial so bill
 $\vee \wedge$  database.users[u].inTrial = TRUE
   $\wedge$  database.users[u].trialStartTime < month
  Ensure users are not double billed
 $\wedge$  database.users[u].billedForMonth < month
 $\wedge$  database' =
  [database EXCEPT
    Add subscription fee
    !["billQueue"] =
      Append(database.billQueue,

```

```

      [user ↦ u,
       fee ↦ SubscriptionFee,
       when ↦ month]),
    !["users"][u].billedForMonth = month
  ]
  ∧ UNCHANGED ⟨events, month⟩

ProcessBills ≜
  ∧ Len(database.billQueue) > 0
  ∧ ∃ i ∈ 1 .. Len(database.billQueue) :
    LET bill ≜ database.billQueue[i] IN
    ∧ bill.when = month
    Charge cancellation fees only a month after canceled
    and still canceled
    ∧ ∨ bill.fee ≠ CancellationFee
      ∨ ∧ bill.fee = CancellationFee
        ∧ ∨ database.users[bill.user].subscribed = FALSE
          Subscribed too late to cancel cancellation fee
          ∨ database.users[bill.user].subscribeMonth ≥ bill.when

    ∧ Bill(bill.user, bill.fee)
    ∧ database' =
      [database EXCEPT
        Removes head of queue
        !["billQueue"] =
          SubSeq(database.billQueue, 1, i - 1) ∘
          SubSeq(database.billQueue, i + 1, Len(database.billQueue))
      ]
    ∧ UNCHANGED ⟨month⟩

```

```

\ * Modification History
\ * Last modified Sun Jun 19 17:55:27 MST 2022 by elliotswart
\ * Created Sun Jun 19 16:56:29 MST 2022 by elliotswart

```