```
MODULE principal
Redeclaration of specdatamodels variables
EXTENDS Sequences, Naturals, FiniteSets

 Represents every potential user in the system
CONSTANT USERS

 Constants that should be set to single model values, to allow comparisons
 Only equality comparisons will be made.
CONSTANTS
    SubscriptionFee,
    CancellationFee,
    FailedPaymentFee

VARIABLES
    month,
     Represents the status of the database. Design requirements require
     that all persistant application state be stored here
    database,
     Required by spec
    events

vars ≜ ⟨events, month, database⟩

 Provides all the data models required by the spec
INSTANCE specdatamodels

Now ≜ Len(events)

Months ≜ 0 . . 10

Strong Typing

Month ≜ Nat

Database Rows
VideoAccessRow ≜
    [type : {"enabled", "disabled"}] ∪
    [
        type : {"disabledasoftimestamp"},
        disabledtime : Month
    ]

TrialStatusRow ≜
    [type : {"eligible", "ineligible", "cancelled"}] ∪
    [
        type : {"started"},
        endtime : Month
    $]$

$SubscriptionStatusRow \triangleq$
    $[type : \{\text{“notsubscribed”}, \text{“subscribed”}, \text{“tocancel”}\}] \cup$
    $[$
        $type : \{\text{“tocancel”}\},$
        $canceltime : Month$
    $]$

$UpcomingChargeItem \triangleq [fee : Fees, event : Nat]$


$TypeOk \triangleq$
    $\land \quad EventsOk$
    $\land \quad month \in Month$
    Represents a singleton record type tracking the billing month
    $\land database.billingMonth \in Month$
    Table with VideoAccessRow row definition
    $\land database.videoAccess \in [USERS \rightarrow VideoAccessRow]$
    Table with TrialStatusRow row definition
    $\land database.trial \in [USERS \rightarrow TrialStatusRow]$
    Table with SubscriptionStatusRow row definition
    $\land database.subscription \in [USERS \rightarrow SubscriptionStatusRow]$
    A row index by user and month, that holds UpcomingCharges
    $\land database.upcomingCharges \in [USERS \rightarrow [Months \rightarrow \text{SUBSET } UpcomingChargeItem]]$
    Table with PastDueStatus row definition
    $\land database.ingoodstanding \in [USERS \rightarrow \text{BOOLEAN}]$


API endpoints

 Database query:
$IsSubscribed(u) \triangleq$
    $\lor database.subscription[u].type = \text{“subscribed”}$
    A converted trial that hasn’t been processed in database
    $\lor \land database.trial[u].type = \text{“started”}$
      $\land database.trial[u].endtime \leq month$

$StartSubscription(u) \triangleq$
    LET $paymentFailedCharges \triangleq$
        IF $database.ingoodstanding[u] = \text{TRUE}$
         THEN $\{\}$
         ELSE $\{[event \mapsto Now, fee \mapsto FailedPaymentFee]\}$
    IN


    LET $thisMonthsCharges \triangleq$
        $database.upcomingCharges[u][month] \cup$
```

```
          Add payment failed fee if applicable
          $paymentFailedCharges$
    IN

     Remove cancellation fee for next month
    LET $nextMonthsCharges \triangleq \{$
                $c \in database.upcomingCharges[u][month + 1] :$
                        $c.fee \neq CancellationFee$
    $\}$
    IN
     Transaction precondition
     $\land \neg IsSubscribed(u)$
     Transaction
     $\land database' =$
        $[database$ EXCEPT
            $![$“ingoodstanding”$][u] =$ TRUE,
            $![$“subscription”$][u] = [type \mapsto$ “subscribed”$],$
            $![$“trial”$][u] = [type \mapsto$ “ineligible”$],$
            $![$“videoAccess”$][u] = [type \mapsto$ “enabled”$],$
             Add charges
            $![$“upcomingCharges”$][u][month] = thisMonthsCharges,$
            $![$“upcomingCharges”$][u][month + 1] = nextMonthsCharges$
        $]$
     Observability required by stub
     $\land events' = Append(events, [type \mapsto$ “startsubscription”$, user \mapsto u])$
     $\land$ UNCHANGED $month$


$CancelSubscription(u) \triangleq$
    LET $cancelTime \triangleq month + 1$IN
    LET $cancellationFee \triangleq [$
                                $event \mapsto Now,$
                                $fee \mapsto CancellationFee]$
    IN
    LET $subscriptionFee \triangleq$
          $[event \mapsto Now,$
           $fee \mapsto SubscriptionFee]$
    IN
     Adds subscription fee for this month if not already added
    LET $thisMonthsCharges \triangleq$ IF $\neg \exists\, charge \in database.upcomingCharges[u][month] :$
                                          $charge.fee = SubscriptionFee$
                                     THEN $database.upcomingCharges[u][month] \cup$
                                          $\{subscriptionFee\}$
                                     ELSE $database.upcomingCharges[u][month]$
    IN
     Transaction precondition
    ∧ IsSubscribed(u)
     Transaction
    ∧ database′ =
        [database EXCEPT
            ![“trial”][u] = [type ↦ “ineligible”],
            ![“subscription”][u] =  [type ↦ “tocancel”, canceltime ↦ cancelTime],
            ![“videoAccess”][u] = [type ↦ “disabledasoftimestamp”,
                                   disabledtime ↦ cancelTime],
            ![“upcomingCharges”][u][month] = thisMonthsCharges,
             Queue up cancellation fee for next month
            ![“upcomingCharges”][u][month + 1] =
                database.upcomingCharges[u][month] ∪
                {cancellationFee}
        ]
     Observability required by stub
    ∧ events′ = Append(events, [type ↦ “cancelsubscription”, user ↦ u])
    ∧ UNCHANGED ⟨month⟩

StartTrial(u) ≜
    LET endTime ≜ month + 1IN


     Transaction precondition
    ∧ database.trial[u].type = “eligible”
     Transaction
    ∧ database′ =
        [database EXCEPT
            ![“trial”][u] = [type ↦ “started”,
                             endtime ↦ endTime],
            ![“videoAccess”][u] =
                              [type ↦ “enabled”]
        ]
     Observability required by stub
    ∧ events′ = Append(events, [type ↦ “starttrial”, user ↦ u])
    ∧ UNCHANGED ⟨month⟩

CancelTrial(u) ≜
     Transaction precondition
    ∧ database.trial[u].type = “started”
     The trial has not already ended
    ∧ database.trial[u].endtime > month
     Transaction
    ∧ database′ =
        [database EXCEPT
            ![“trial”][u] = [type ↦ “cancelled”],
            $![\text{“videoAccess”}][u] =$
                                $[type \mapsto \text{“disabled”}]$
        $]$
      Observability required by stub
      $\land\ events' = Append(events,\ [type \mapsto \text{“canceltrial”},\ user \mapsto u])$
      $\land$ UNCHANGED $\langle month \rangle$

  Database query
$WatchVideoAuthorized(u) \triangleq$
      $\lor\ database.videoAccess[u].type = \text{“enabled”}$
      $\lor\ \land\ database.videoAccess[u].type = \text{“disabledasoftimestamp”}$
         $\land\ database.videoAccess[u].disabledtime > month$

$WatchVideo(u) \triangleq$
      $\land\ WatchVideoAuthorized(u)$
      Observability required by stub
      $\land\ events' = Append(events,\ [type \mapsto \text{“watchvideo”},\ user \mapsto u])$
      $\land$ UNCHANGED $\langle month,\ database \rangle$


$Bill(u,\ fee) \triangleq$
      $\land\ events' = Append(events,\ [type \mapsto \text{“bill”},$
                                      $user \mapsto u,$
                                      $fee \mapsto fee])$

$PaymentFailed(u,\ fee) \triangleq$
      $\land\ database' =$
        $[database$ EXCEPT
            $![\text{“ingoodstanding”}][u] =$ FALSE,
            $![\text{“subscription”}][u] = [type \mapsto \text{“notsubscribed”}],$
            $![\text{“trial”}][u] = [type \mapsto \text{“ineligible”}],$
            $![\text{“videoAccess”}][u] = [type \mapsto \text{“disabled”}],$
             Remove all upcoming changes
            $![\text{“upcomingCharges”}][u][month] = \{\},$
            $![\text{“upcomingCharges”}][u][month + 1] = \{\}$
        $]$
      $\land\ events' = Append(events,\ [type \mapsto \text{“paymentfailed”},$
                                      $user \mapsto u,$
                                      $fee \mapsto fee])$
      $\land$ UNCHANGED $\langle month \rangle$
```

Recurring Operations

```
  This the the state that calls the Payment Failed $API$
$ExistingBillFailed \triangleq$
      $\lor\ \exists\, i \in 1 \mathrel{..} Len(events) :$
           Only a past bill can fail
```

```
        $\land\ events[i] \in BillEvent$
        $\land\ PaymentFailed(events[i].user,\ events[i].fee)$

Trial users that have passed trial period are subscribed
$ConvertTrialUser(u)\ \triangleq$
    $\land\ database.trial[u].type = \text{“started”}$
    $\land\ database.trial[u].endtime \leq month$ The trial has ended
    Transaction
    $\land\ database' =$
        $[database\ \text{EXCEPT}$
            $![\text{“subscription”}][u] = [type \mapsto \text{“subscribed”}],$
            $![\text{“trial”}][u] = [type \mapsto \text{“ineligible”}],$
            $![\text{“videoAccess”}][u] = [type \mapsto \text{“enabled”}]$
        $]$
    $\land\ \text{UNCHANGED}\ \langle month,\ events \rangle$

Cancelled users that have passed cancellation period are unsubscribed
$ProcessCancelledUser(u)\ \triangleq$
    $\land\ database.subscription[u].type = \text{“tocancel”}$
    $\land\ database.subscription[u].canceltime \leq month$
    $\land\ database' = [database\ \text{EXCEPT}$
            unsubscribe
            $![\text{“subscription”}][u] = [type \mapsto \text{“notsubscribed”}]$
        $]$
    $\land\ \text{UNCHANGED}\ \langle month,\ events \rangle$

Any subscribed user is billed this month
$BillUserForSubscription(u)\ \triangleq$
    $\text{LET}\ subscriptionFee\ \triangleq$
        $[event \mapsto Now,$
         $fee \mapsto SubscriptionFee]$
    $\text{IN}$
    $\land\ database.subscription[u].type = \text{“subscribed”}$
    Add subscription fee for this month
    $\land\ database' =$
        $[database\ \text{EXCEPT}$
                $![\text{“upcomingCharges”}][u][month] =$
                    $database.upcomingCharges[u][month] \cup$
                    $\{subscriptionFee\}$
        $]$
    $\land\ \text{UNCHANGED}\ \langle month,\ events \rangle$

Bill users for their current month charges
$ProcessCharges\ \triangleq$
    $\land\ \exists\, u \in USERS :$
        $\text{LET}\ monthlyCharges\ \triangleq\ database.upcomingCharges[u][month]\text{IN}$
```

```
            If there are upcoming charges for this month
            $\land\ Cardinality(monthlyCharges) > 0$
            Deque a bill
            $\land\ \exists\ charge \in monthlyCharges :$
                Submit to payment processor
                $\land\ Bill(u,\ charge.fee)$
                Delete from queue
                $\land\ database' =$
                    $[database\ \textsc{except}$
                        $![\text{“upcomingCharges”}][u][month] = monthlyCharges \setminus \{charge\}$
                    $]$
        $\land\ \textsc{unchanged}\ month$

Stub method that prevents the month from passing until all operations are complete. Represent worker methods, etc
$HandledMonth\ \triangleq$
        $\land\ \neg\textsc{enabled}\ ProcessCharges$
        $\land\ \forall\ u \in USERS :$
            $\land\ \neg\textsc{enabled}\ ConvertTrialUser(u)$
            $\land\ \neg\textsc{enabled}\ ProcessCancelledUser(u)$
            $\land\ \neg\textsc{enabled}\ BillUserForSubscription(u)$


 DO NOT MODIFY
$MonthPasses\ \triangleq$
        $\land\ HandledMonth$
        $\land\ month' = month + 1$
        $\land\ events' = Append(events,\ [type \mapsto \text{“monthpass”}])$
        $\land\ \textsc{unchanged}\ \langle database \rangle$

Specification
$Init\ \triangleq$
        $\land\ events = \langle\rangle$  Events must be intialized empty, per stub
        $\land\ month = 0$
        $\land\ database = [$
                no months have been billed for yet
                $billingMonth \mapsto 0,$
                No user starts with access
                $videoAccess \mapsto [u \in USERS \mapsto [type \mapsto \text{“disabled”}]],$
                Every user starts eligible for trial
                $trial \mapsto [u \in USERS \mapsto [type \mapsto \text{“eligible”}]],$
                Every user starts not subscribed
                $subscription \mapsto [u \in USERS \mapsto [type \mapsto \text{“notsubscribed”}]],$
                All users start in good standing
                $ingoodstanding \mapsto [u \in USERS \mapsto \textsc{true}],$
                No bills to submit
            $upcomingCharges \mapsto [u \in USERS \mapsto [x \in Months \mapsto \{\}]]$
        ]
$Next \triangleq$
    \* Required by stub
    $\lor MonthPasses$
    \* State modified below
    $\lor \exists u \in USERS :$
            $\lor StartSubscription(u)$
            $\lor CancelSubscription(u)$
            $\lor StartTrial(u)$
            $\lor CancelTrial(u)$
            $\lor WatchVideo(u)$
            $\lor ConvertTrialUser(u)$
            $\lor ProcessCancelledUser(u)$
            $\lor ConvertTrialUser(u)$
            $\lor BillUserForSubscription(u)$
    \* Payment failing behavior is part of spec not implementation
    $\lor ExistingBillFailed$
    $\lor ProcessCharges$
```

```
\* Modification History
\* Last modified Sun Jun 19 17:45:52 MST 2022 by elliotswart
\* Created Fri Jun 17 00:28:26 MST 2022 by elliotswart
```