

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

Represents the current month

month,

Represents the status of the database. Design requirements require
 that all persistent application state be stored here

database,

Required by spec

events

vars \triangleq \langle *events, month, database* \rangle

Provides all the data models required by the spec

INSTANCE *specdatamodels*

Now \triangleq *Len(events)*

Months \triangleq 0 .. 10

Strong Typing

Month \triangleq *Nat*

Database Rows

UserRow \triangleq [
subscribed : BOOLEAN ,
 Forget *cancelled*
inTrial : BOOLEAN ,
trialStartTime : *Nat*,
billedForMonth : *Nat*,
hasHadTrialOrSubscription : BOOLEAN ,
hasCancelled : BOOLEAN ,

$$\begin{aligned}
& \text{cancelMonth} : \text{Nat} \\
&] \\
\text{BillQueueItem} & \triangleq [\\
& \text{user} : \text{USERS}, \\
& \text{fee} : \text{Fees} \\
&] \\
\text{TypeOk} & \triangleq \\
& \wedge \text{EventsOk} \\
& \wedge \text{month} \in \text{Month} \\
& \wedge \text{database.users} \in [\text{USERS} \rightarrow \text{UserRow}] \\
& \wedge \text{database.billQueue} \in \text{Seq}(\text{BillQueueItem})
\end{aligned}$$

API endpoints

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

$$\begin{aligned}
\text{CancelSubscription}(u) & \triangleq \\
& \text{Subscribed} \\
& \wedge \vee \text{database.users}[u].\text{subscribed} = \text{TRUE} \\
& \vee \wedge \text{database.users}[u].\text{inTrial} = \text{TRUE} \\
& \wedge \text{database.users}[u].\text{trialStartTime} < \text{month} \\
& \wedge \text{database}' = \\
& \quad [\text{database EXCEPT} \\
& \quad \quad !["users"][u].\text{subscribed} = \text{FALSE}, \\
& \quad \quad !["users"][u].\text{inTrial} = \text{FALSE}, \\
& \quad \quad !["users"][u].\text{hasCancelled} = \text{TRUE}, \\
& \quad \quad !["users"][u].\text{cancelMonth} = \text{month}, \\
& \quad \quad \text{Charge cancellation fee} \\
& \quad]
\end{aligned}$$

$$\begin{aligned} & ![\text{"billQueue"}] = \\ & \quad \text{Append}(\text{database.billQueue}, \\ & \quad \quad [\text{user} \mapsto u, \text{fee} \mapsto \text{CancellationFee}]) \\ & \end{aligned}$$

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

$$\begin{aligned} \text{StartTrial}(u) & \triangleq \\ & \wedge \text{database.users}[u].\text{inTrial} = \text{FALSE} \\ & \wedge \text{database.users}[u].\text{subscribed} = \text{FALSE} \\ & \wedge \text{database.users}[u].\text{hasHadTrialOrSubscription} = \text{FALSE} \\ & \wedge \text{database}' = [\text{database EXCEPT} \\ & \quad \quad ![\text{"users"}][u].\text{inTrial} = \text{TRUE}, \\ & \quad \quad ![\text{"users"}][u].\text{trialStartTime} = \text{month}, \\ & \quad \quad ![\text{"users"}][u].\text{hasHadTrialOrSubscription} = \text{TRUE} \\ & \quad \quad] \end{aligned}$$

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

$$\begin{aligned} \text{CancelTrial}(u) & \triangleq \\ & \text{In active trial} \\ & \wedge \text{database.users}[u].\text{inTrial} = \text{TRUE} \\ & \wedge \text{database.users}[u].\text{trialStartTime} = \text{month} \\ & \text{And not subscribed} \\ & \wedge \text{database.users}[u].\text{subscribed} = \text{FALSE} \\ & \wedge \text{database}' = [\text{database EXCEPT} \\ & \quad \quad ![\text{"users"}][u].\text{inTrial} = \text{FALSE} \\ & \quad \quad] \end{aligned}$$

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

$$\begin{aligned} \text{WatchVideo}(u) & \triangleq \\ & \wedge \vee \text{database.users}[u].\text{subscribed} = \text{TRUE} \\ & \quad \vee \text{database.users}[u].\text{inTrial} = \text{TRUE} \\ & \quad \text{Remove video access at the end of cancelled month} \\ & \quad \vee \wedge \text{database.users}[u].\text{hasCancelled} = \text{TRUE} \\ & \quad \quad \wedge \text{database.users}[u].\text{cancelMonth} = \text{month} \end{aligned}$$

Observability required by stub

$\wedge events' = Append(events, [type \mapsto \text{"watchvideo"}, user \mapsto u])$
 $\wedge UNCHANGED \langle month, database \rangle$

Stub method, do not changed

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

$PaymentFailed(u, fee) \triangleq$
 $\wedge database' = [database \text{ EXCEPT}$
 $!["users"][u].subscribed = FALSE,$
 $!["users"][u].hasCancelled = FALSE$
 $]$

Observability required by stub

$\wedge events' = Append(events, [type \mapsto \text{"paymentfailed"},$
 $user \mapsto u,$
 $fee \mapsto fee])$
 $\wedge UNCHANGED \langle month \rangle$

Recurring Operations

This the the state that calls the Payment Failed API

$ExistingBillFailed \triangleq$
 $\vee \exists i \in 1 .. Len(events) :$
 $\text{Only a past bill can fail}$
 $\wedge events[i] \in BillEvent$
 $\wedge PaymentFailed(events[i].user, events[i].fee)$

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

$$\begin{aligned}
&] \\
& \wedge \text{UNCHANGED } \langle \text{events}, \text{month} \rangle \\
\text{ProcessBills} & \triangleq \\
& \wedge \text{Len}(\text{database.billQueue}) > 0 \\
& \wedge \text{LET } \text{bill} \triangleq \text{Head}(\text{database.billQueue}) \text{ IN} \\
& \quad \text{Bills user} \\
& \quad \wedge \text{Bill}(\text{bill.user}, \text{bill.fee}) \\
& \quad \wedge \text{database}' = \\
& \quad \quad [\text{database EXCEPT} \\
& \quad \quad \quad \text{Removes head of queue} \\
& \quad \quad \quad ![\text{"billQueue"}] = \\
& \quad \quad \quad \quad \text{SubSeq}(\text{database.billQueue}, \\
& \quad \quad \quad \quad \quad 2, \text{Len}(\text{database.billQueue})) \\
& \quad \quad] \\
& \wedge \text{UNCHANGED } \langle \text{month} \rangle
\end{aligned}$$

Stub method that prevents the month from passing until all operations are complete. Represent worker methods, etc

$$\begin{aligned}
\text{HandledMonth} & \triangleq \\
& \wedge \neg \text{ENABLED } \text{BillSubscribedUsers} \\
& \wedge \neg \text{ENABLED } \text{ProcessBills}
\end{aligned}$$

DO NOT MODIFY

$$\begin{aligned}
\text{MonthPasses} & \triangleq \\
& \wedge \text{HandledMonth} \\
& \wedge \text{month}' = \text{month} + 1 \\
& \wedge \text{events}' = \text{Append}(\text{events}, [\text{type} \mapsto \text{"monthpass"}]) \\
& \wedge \text{UNCHANGED } \langle \text{database} \rangle
\end{aligned}$$

Specification

$$\begin{aligned}
\text{Init} & \triangleq \\
& \wedge \text{events} = \langle \rangle \quad \text{Events must be intialized empty, per stub} \\
& \wedge \text{month} = 0 \\
& \wedge \text{database} = [\\
& \quad \text{Users start with everything unset} \\
& \quad \text{users} \mapsto \\
& \quad \quad [\text{u} \in \text{USERS} \mapsto \\
& \quad \quad \quad [\\
& \quad \quad \quad \quad \text{subscribed} \mapsto \text{FALSE}, \\
& \quad \quad \quad \quad \text{inTrial} \mapsto \text{FALSE}, \\
& \quad \quad \quad \quad \text{trialStartTime} \mapsto 0, \\
& \quad \quad \quad \quad \text{billedForMonth} \mapsto 0, \\
& \quad \quad \quad \quad \text{hasHadTrialOrSubscription} \mapsto \text{FALSE}, \\
& \quad \quad \quad \quad \text{hasCancelled} \mapsto \text{FALSE}, \\
& \quad \quad \quad \quad \text{cancelMonth} \mapsto 0 \\
& \quad \quad \quad] \\
& \quad \quad] \\
& \quad]
\end{aligned}$$

```

    ]
  ],
  Bill queue starts empty
  billQueue  $\mapsto$   $\langle \rangle$ 
]

```

```

Next  $\triangleq$ 
  \* Required by stub
   $\vee$  MonthPasses
  \* State modified below
   $\vee \exists u \in USERS:$ 
     $\vee$  StartSubscription( $u$ )
     $\vee$  CancelSubscription( $u$ )
     $\vee$  StartTrial( $u$ )
     $\vee$  CancelTrial( $u$ )
     $\vee$  WatchVideo( $u$ )
  \* Add more user based states
  \* Payment failing behavior is part of spec not implementation
   $\vee$  ExistingBillFailed
   $\vee$  BillSubscribedUsers

```

```

\* Modification History
\* Last modified Sun Jun 19 17:44:01 MST 2022 by elliotswart
\* Created Sun Jun 19 16:56:29 MST 2022 by elliotswart

```