

# Formal Specification and Verification of Programs

3rd Assignment Solutions  
Mohammad Hossein Khoshechin - 99210164  
Group 2

۲۴ بهمن ۱۳۹۹

## توصیف فرمال سیستم SnappDoctor

پروژه اینجانب توصیف رسمی سیستم اسنپ داکتر بود و توصیف صورت گرفته نشأت گرفته از امکانات این سیستم می باشد. در این سیستم هر پزشک می تواند ثبت نام کند و سرویس دهد. هر مشتری نیز می تواند ثبت نام کند و از پزشکان به صورت تماس آنلاین و یا چت آنلاین مشاوره بگیرد و یا یک نوبتی را رزرو کند و در آن زمان داکتر با او تماس بگیرد. علاوه بر آن هر داروخانه میتواند در این سیستم ثبت نام کند و سرویس دهد و هر مشتری ای که نسخه داکتر داشته باشد میتواند داروهایش را به صورت آنلاین تهیه کند.

در این بخش free type و given type های مورد نیاز برای توصیف نوشته شده اند.

[Name]  
[Major]  
[Description]  
[Prescription]  
[BusinessLicense]  
[Drug]  
Day == sunday | monday | tuesday | wednesday | thursday | friday | saturday  
Gender == male | female  
Response == DoctorRegistered | MedicalSystemIDisNotValid | PhoneNumberisNotValid |  
PatientRegistered | BalanceCharged | InvalidCharge | CallReserved | FaildReserve |  
CallMade | InvalidCall | TextMade | InvalidText | PharmacyRegistered | LicenseisNotValid |  
PurchaseMade | PurchaseFaield

در این قسمت یک سری شما به عنوان تایپ تعریف شده اند از قبیل Schedule که نشان میدهد در یک روز از هفته چه ساعتی و تا چه ساعتی مشخص شده است. کارت اعتباری که مشتری جهت پرداخت از آن استفاده می کند . تایپ Date که برای ثبت تاریخ یک روز مورد استفاده قرار میگیرد.

#### *Schedule*

*day* : *Day*  
*from* :  $\mathbb{N} \times \mathbb{N}$   
*to* :  $\mathbb{N} \times \mathbb{N}$

$from.1 \geq 0 \wedge from.1 \leq 23$   
 $from.2 \geq 0 \wedge from.2 \leq 59$   
 $to.1 \geq 0 \wedge to.1 \leq 23$   
 $to.2 \geq 0 \wedge to.2 \leq 59$   
 $(from.1 \leq to.1) \vee (from.1 = to.1 \wedge from.2 \leq to.2)$

#### *CreditCard*

*cardId* :  $\mathbb{N}$   
*pinCode* :  $\mathbb{N}$   
*cvv2* :  $\mathbb{N}$   
*expireDate* : *Date*

#### *Date*

*year* :  $\mathbb{N}$   
*mounth* :  $\mathbb{N}$   
*day* :  $\mathbb{N}$

$year \geq 1300 \wedge year \leq 1399$   
 $mounth \leq 12$   
 $day \leq 31$   
 $mounth \leq 6 \Rightarrow day \leq 31$   
 $mounth \geq 7 \wedge mounth \neq 12 \Rightarrow day \leq 30$   
 $mounth = 12 \Rightarrow day \leq 29$

در این قسمت تایپ دکتر نوشته شده است که دارای صفات نام و نام خانوادگی و شماره نظام پزشکی و تخصص اش است. صفت دیگری که دارد یک برنامه زمانی در هفته است که نشان میدهد در هر روز از هفته چه ساعاتی در دسترس است. دو صفت هم در نظر گرفته شده برای اینکه در هر دقیقه تماس تلفنی و یا تماس نوشتاری چه مبلغی دریافت می کند.

#### *Doctor*

*firstName* : Name  
*lastName* : Name  
*medicalSystemID* :  $\mathbb{N}$   
*major* : Major  
*schedules* :  $\mathbb{P}$  Schedule  
*textPerMin* :  $\mathbb{N}$   
*callPerMin* :  $\mathbb{N}$

در این قسمت تایپ بیمار توصیف شده است . که دارای نام و نام خانوادگی است . سن و جنسیت و آدرس و تلفنش را نیز در نظر میگیریم

#### *Patient*

*firstName* : Name  
*lastName* : Name  
*age* :  $\mathbb{N}$   
*gender* : Gender  
*address* : Description  
*phoneNum* :  $\mathbb{N}$

در این قسمت تایپ داروخانه را توصیف کرده ایم که دارای نام و جواز کسب و داروهایی است که دارد. اینطور فرض کردیم که قیمت تمام داروها یکسان است.

#### *Pharmacy*

*name* : Name  
*license* : BusinessLicense  
*drugs* :  $\mathbb{P}$  Drug  
*pricePerDrug* :  $\mathbb{N}$

در این بخش شمای حالت را مینویسم. سیستم اسنپ مشخصات همه بیماران و دکترها و داروخانه ها را ذخیره میکند. هر بیمار یک وضعیتی دارد نسبت به سلامت جسمانی و روانی خود که آن هم باید ذخیره شود و هر بار که بیمار درخواست پزشک میکند باید توسط خود بیمار آپدیت شود. در این سیستم هر شخص بر اساس پولی که به حساب اسنپ میریزد یک موجودی ای دارد که آن هم ذخیره می شود . هر بیمار زمانی که یک دکتر را برای تماس تلفنی رزرو میکند نیز باید اطلاعاتش ذخیره شود . علاوه بر این ها مشخصات تماس بیمار با دکتر و تماس تایپی بیمار با دکتر نیز باید ذخیره شود.

*SnappDoctor*

$doctors : \mathbb{P} \text{ Doctor}$   
 $patients : \mathbb{P} \text{ Patient}$   
 $pharmacies : \mathbb{P} \text{ Pharmacy}$   
 $callReserve : Patient \rightarrow \mathbb{P}(\text{Date} \times \text{Schedule} \times \text{Doctor})$   
 $calledDoctor : Patient \rightarrow \mathbb{P}(\text{Date} \times \text{Schedule} \times \text{Doctor})$   
 $textDoctor : Patient \rightarrow \mathbb{P}(\text{Date} \times \text{Schedule} \times \text{Doctor})$   
 $balance : Patient \rightarrow \mathbb{N}$   
 $status : Patient \rightarrow \text{Description}$

در این بخش مقداردهی اولیه شمای حالت را توصیف می کنیم.

*SnappDoctorInit*

*SnappDoctor'*

$doctors' = \emptyset$   
 $patients' = \emptyset$   
 $callReserve' = \emptyset$   
 $calledDoctor' = \emptyset$   
 $textDoctor' = \emptyset$   
 $balance' = \emptyset$   
 $status' = \emptyset$   
 $pharmacies' = \emptyset$

در ادامه شما های عملیات هایمان را می نویسیم.

۱

در این بخش ثبت نام پزشک در سیستم را توصیف میکنیم. ابتدا باید شماره نظام پزشکی اش چک شود که چون این عملیات توسط یک third party انجام می شود ما یک تابع به صورت پیشفرض در نظر گرفتیم به نام checkID که با دادن شماره نظام پزشکی دکتر به آن صحتش بررسی می شود اگر معتبر باشد مقدار true و اگر نامعتبر باشد مقدار false برمیگرداند. هیچ دو پزشکی نیز نباید شماره نظام پزشکی شان با هم برابر باشد.

<i>RegisterDoctor</i>
$\Delta SnappDoctor$
$d? : Doctor$
$checkID(d?.medicalSystemID)$ $\forall d : Doctor \mid d \in doctors \bullet d.medicalSystemID \neq d?.medicalSystemID$ $doctors' = doctors \cup d?$ $patients' = patients$ $callReserve' = callReserve$ $calledDoctor' = calledDoctor$ $textDoctor' = textDoctor$ $balance' = balance$ $status' = status$ $pharmacies' = pharmacies$

<i>Success<sub>0</sub></i>
$r! : Response$
$r! = DoctorRegistered$

<i>DoctorNotValid</i>
$\Xi SnappDoctor$
$d? : Doctor$
$\neg checkID(d?.medicalSystemID) \vee$ $\exists d : Doctor \mid d \in doctors \bullet d.medicalSystemID = d?.medicalSystemID$

<i>Failure<sub>0</sub></i>
$r! : Response$
$r! = MedicalSystemIDisNotValid$

$$SignUpDoctor == (RegisterDoctor \wedge Success_0) \vee (DoctorNotValid \wedge Failure_0)$$

۲

در این بخش ثبت نام بیمار را توصیف کرده ایم . شماره موبایل آن هم باید توسط یک پیامک چک شود که چون این کار توسط یک third party انجام میشود ما هم از یک تابع پیشفرض با نام

checkPhoneNum جهت اعتبار سنجی تلفن او استفاده کردیم. شماره تلفن هیچ دو بیماری نباید با هم برابر باشد.

$RegisterPatient$ $\Delta SnappDoctor$ $p? : Patient$ $s? : Description$	
$checkPhoneNum(p?.phoneNum)$ $\forall p : Patient \mid p \in patients \bullet p.phonNum \neq p?.phoneNum$ $patients' = patients \cup p?$ $doctors' = doctors$ $callReserve' = callReserve$ $calledDoctor' = calledDoctor$ $textDoctor' = textDoctor$ $balance' = balance \cup \{p? \mapsto 0\}$ $status' = status \cup \{p? \mapsto s?\}$ $pharmacies' = pharmacies$	

$Success_1$ $r! : Response$	
$r! = PatientRegistered$	

$PatientNotValid$ $\Xi SnappDoctor$ $p? : Patient$	
$\neg checkPhoneNum(p?.phoneNum) \vee$ $\exists p : Patient \mid p \in patients \bullet p.phoneNum = p?.phoneNum$	

$Failure_1$ $r! : Response$	
$r! = PhoneNumberisNotValid$	

$$SignUpPatient == (RegisterPatient \wedge Success_1) \vee (PatientNotValid \wedge Failure_1)$$

در این بخش عملیات افزایش اعتبار توسط بیمار را توصیف میکنیم. هر شخص توسط کارت اعتباری اش مبلغی را به اعتبارش اضافه میکند. چون عملیات برداشت پول توسط بانک انجام می شود بنابراین ما هم از یک تابع پیشفرض به نام transaction جهت انجام عملیات بانکی استفاده میکنیم. اگر عملیات موفقیت آمیز باشد مقدار true در غیر این صورت مقدار false را بر میگردداند.

<i>ChargeBalance</i>
$\Delta SnappDoctor$
$p? : Patient$
$m? : \mathbb{N}$
$c? : CreditCard$
$p? \in patients$
$transaction(c?, m?)$
$balance' = balance \oplus \{p? \mapsto (balance(p?) + m?)\}$
$patients' = patients$
$callReserve' = callReserve$
$calledDoctor' = calledDoctor$
$textDoctor' = textDoctor$
$doctors' = doctors$
$status' = status$
$pharmacies' = pharmacies$

<i>InvalidCharge</i>
$\exists SnappDoctor$
$p? : Patient$
$m? : \mathbb{N}$
$c? : CreditCard$
$\neg transaction(c?, m?)$
$p? \notin patients$

<i>Success<sub>2</sub></i>
$r! : Response$
$r! = BalanceCharged$

<i>Failure<sub>2</sub></i>
$r! : Response$
$r! = InvalidCharge$

$$Charge == (ChargeBalance \wedge Success_2) \vee (InvalidCharge \wedge Failure_2)$$

۴

در این بخش عملیات رزرو تماس با دکتر توسط بیمار را توصیف میکنیم. اولاً باید بیمار در اعتبارش برای هزینه تماسش با دکتر پول داشته باشد. دوماً نباید در آن تایم و تاریخی که یک فرد میخواهد با دکتر رزرو کند کسی دیگر رزرو کرده باشد. هزینه رزرو تماس هم باید در پایان از اعتبار بیمار کسر شود.

*SubmitReserve*

$\Delta SnappDoctor$

$p? : Patient$

$d? : Doctor$

$t? : Date$

$s? : Schedule$

$u? : Description$

$m : \mathbb{N}$

$p? \in patients$

$d? \in doctors$

$\exists x : Schedule \mid x \in d?.schedules \bullet x.day = s?.day \wedge x.from \leq s?.from \wedge x.to \geq s?.to$

$\forall p : Patient \mid p \in callReserve \bullet$

$\forall x : Date \times Schedule \times Doctor \mid x \in callReserve(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$callReserve' = callReserve \oplus \{p? \mapsto \{callReserve(p) \cup \{(t?, s?, d?)\}\}$

$patients' = patients$

$calledDoctor' = calledDoctor$

$textDoctor' = textDoctor$

$doctors' = doctors$

$m = (((s?.to).1 - (s?.from).1) * 60 + ((s?.to).2 - (s?.from).2)) * d?.callPerMin$

$m \leq balance(p?)$

$balance' = balance \oplus \{p? \mapsto balance(p?) - m\}$

$status' = status \oplus \{p? \mapsto u?\}$

$pharmacies' = pharmacies$



$\text{FaildReserve}$ $\exists \text{SnappDoctor}$ $p? : \text{Patient}$ $d? : \text{Doctor}$ $t? : \text{Date}$ $s? : \text{Schedule}$ $u? : \text{Description}$ $m : \mathbb{N}$
$m = (((s?.to).1 - (s?.from).1) * 60 + ((s?.to).2 - (s?.from).2)) * d?.callPerMin$ $m > \text{balance}(p?) \vee$ $p? \notin \text{patients} \vee$ $d? \notin \text{doctors} \vee$ $\neg(\exists x : \text{Schedule} \mid x \in d?.schedules \bullet x.day = s?.day \wedge x.from \leq s?.from \wedge x.to \geq s?.to) \vee$ $\neg(\forall p : \text{Patient} \mid p \in \text{callReserve} \bullet$ $\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{callReserve}(p) \bullet$ $\quad \quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset)$
$\text{Success}_3$ $r! : \text{Response}$
$r! = \text{CallReserved}$
$\text{Failure}_3$ $r! : \text{Response}$
$r! = \text{InvalidReserve}$

$$\text{MakeReserve} == (\text{SubmitReserve} \wedge \text{Success}_3) \vee (\text{FaildReserve} \wedge \text{Failure}_3)$$

۵

در این بخش عملیات تماس آنلاین با دکتر را توصیف میکنیم. اولاً باید بررسی کنیم که دکتر مورد نظر در حال تماس آنلاین و یا تماس تایپی در آن تاریخ و زمان با شخص دیگری نباشد دوماً کسی در آن تاریخ و زمان با دکتر رزرو نکرده باشد و اعتبار کافی داشته باشد و در انتهای تماس هزینه اش از اعتبارش کم شود.

*SubmitCall*

$\Delta \text{SnappDoctor}$

$p? : \text{Patient}$

$d? : \text{Doctor}$

$t? : \text{Date}$

$s? : \text{Schedule}$

$u? : \text{Description}$

$m : \mathbb{N}$

$p? \in \text{patients}$

$d? \in \text{doctors}$

$\exists x : \text{Schedule} \mid x \in d?.\text{schedules} \bullet x.\text{day} = s?.\text{day} \wedge x.\text{from} \leq s?.\text{from} \wedge x.\text{to} \geq s?.\text{to}$

$\forall p : \text{Patient} \mid p \in \text{callReserve} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{callReserve}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\forall p : \text{Patient} \mid p \in \text{calledDoctor} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{calledDoctor}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\forall p : \text{Patient} \mid p \in \text{textDoctor} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{textDoctor}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\text{calledDoctor}' = \text{calledDoctor} \oplus \{p? \mapsto \{\text{calledDoctor}(p) \cup \{(t?, s?, d?)\}\}\}$

$\text{patients}' = \text{patients}$

$\text{callReserve}' = \text{callReserve}$

$\text{textDoctor}' = \text{textDoctor}$

$\text{doctors}' = \text{doctors}$

$m = (((s?.\text{to}).1 - (s?.\text{from}).1) * 60 + ((s?.\text{to}).2 - (s?.\text{from}).2)) * d?.\text{callPerMin})$

$m \leq \text{balance}(p?)$

$\text{balance}' = \text{balance} \oplus \{p? \mapsto \text{balance}(p?) - m\}$

$\text{status}' = \text{status} \oplus \{p? \mapsto u?\}$

$\text{pharmacies}' = \text{pharmacies}$

<i>FaildCall</i>	
$\exists \text{SnappDoctor}$	
$p? : \text{Patient}$	
$d? : \text{Doctor}$	
$t? : \text{Date}$	
$s? : \text{Schedule}$	
$u? : \text{Description}$	
$m : \mathbb{N}$	
$m = (((s?.to).1 - (s?.from).1) * 60 + ((s?.to).2 - (s?.from).2)) * d?.callPerMin$	
$m > \text{balance}(p?) \vee$	
$p? \notin \text{patients} \vee$	
$d? \notin \text{doctors} \vee$	
$\neg(\exists x : \text{Schedule} \mid x \in d?.schedules \bullet x.day = s?.day \wedge x.from \leq s?.from \wedge x.to \geq s?.to) \vee$	
$\neg(\forall p : \text{Patient} \mid p \in \text{callReserve} \bullet$	
$\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{callReserve}(p) \bullet$	
$\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset) \vee$	
$\neg(\forall p : \text{Patient} \mid p \in \text{calledDoctor} \bullet$	
$\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{calledDoctor}(p) \bullet$	
$\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset) \vee$	
$\neg(\forall p : \text{Patient} \mid p \in \text{textDoctor} \bullet$	
$\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{textDoctor}(p) \bullet$	
$\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset)$	
<i>Success<sub>4</sub></i>	
$r! : \text{Response}$	
$r! = \text{CallMade}$	
<i>Failure<sub>4</sub></i>	
$r! : \text{Response}$	
$r! = \text{InvalidCall}$	

$$\text{MakeCall} == (\text{SubmitCall} \wedge \text{Success}_4) \vee (\text{FaildCall} \wedge \text{Failure}_4)$$

۶

در این بخش به مانند بخش قبل عملیات تماس تاییدی با دکتر را توصیف می کنیم.

*SubmitText*

$\Delta \text{SnappDoctor}$

$p? : \text{Patient}$

$d? : \text{Doctor}$

$t? : \text{Date}$

$s? : \text{Schedule}$

$u? : \text{Description}$

$m : \mathbb{N}$

$p? \in \text{patients}$

$d? \in \text{doctors}$

$\exists x : \text{Schedule} \mid x \in d?.\text{schedules} \bullet x.\text{day} = s?.\text{day} \wedge x.\text{from} \leq s?.\text{from} \wedge x.\text{to} \geq s?.\text{to}$

$\forall p : \text{Patient} \mid p \in \text{callReserve} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{callReserve}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\forall p : \text{Patient} \mid p \in \text{calledDoctor} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{calledDoctor}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\forall p : \text{Patient} \mid p \in \text{textDoctor} \bullet$

$\forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{textDoctor}(p) \bullet$

$x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset$

$\text{textDoctor}' = \text{textDoctor} \oplus \{p? \mapsto \{\text{textDoctor}(p) \cup \{(t?, s?, d?)\}\}\}$

$\text{patients}' = \text{patients}$

$\text{callReserve}' = \text{callReserve}$

$\text{calledDoctor}' = \text{calledDoctor}$

$\text{doctors}' = \text{doctors}$

$m = (((s?.\text{to}).1 - (s?.\text{from}).1) * 60 + ((s?.\text{to}).2 - (s?.\text{from}).2)) * d?.\text{textPerMin})$

$m \leq \text{balance}(p?)$

$\text{balance}' = \text{balance} \oplus \{p? \mapsto \text{balance}(p?) - m\}$

$\text{status}' = \text{status} \oplus \{p? \mapsto u?\}$

$\text{pharmacies}' = \text{pharmacies}$

<i>FaildText</i>
$\exists \text{SnappDoctor}$ $p? : \text{Patient}$ $d? : \text{Doctor}$ $t? : \text{Date}$ $s? : \text{Schedule}$ $u? : \text{Description}$ $m : \mathbb{N}$
$m = (((s?.to).1 - (s?.from).1) * 60 + ((s?.to).2 - (s?.from).2)) * d?.textPerMin$ $m > \text{balance}(p?) \vee$ $p? \notin \text{patients} \vee$ $d? \notin \text{doctors} \vee$ $\neg(\exists x : \text{Schedule} \mid x \in d?.schedules \bullet x.day = s?.day \wedge x.from \leq s?.from \wedge x.to \geq s?.to) \vee$ $\neg(\forall p : \text{Patient} \mid p \in \text{callReserve} \bullet$ $\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{callReserve}(p) \bullet$ $\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset) \vee$ $\neg(\forall p : \text{Patient} \mid p \in \text{calledDoctor} \bullet$ $\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{calledDoctor}(p) \bullet$ $\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset) \vee$ $\neg(\forall p : \text{Patient} \mid p \in \text{textDoctor} \bullet$ $\quad \forall x : \text{Date} \times \text{Schedule} \times \text{Doctor} \mid x \in \text{textDoctor}(p) \bullet$ $\quad x.1 = t? \wedge x.3 = p? \Rightarrow x.2 \cap s? = \emptyset)$

<i>Success<sub>5</sub></i>
$r! : \text{Response}$
$r! = \text{TextMade}$

<i>Failure<sub>5</sub></i>
$r! : \text{Response}$
$r! = \text{InvalidText}$

$$\text{MakeText} == (\text{SubmitText} \wedge \text{Success}_5) \vee (\text{FaildText} \wedge \text{Failure}_5)$$

✓

در این بخش عملیات جستجوی دکتر بر اساس نام و یا بر اساس تخصص اش را توصیف میکنیم.

<i>SearchByName</i>
$\exists \text{SnappDoctor}$
$fn? : \text{Name}$
$ln? : \text{Name}$
$d! : \mathbb{P} \text{ Doctor}$
$d! = \{x : \text{Doctor} \mid x \in \text{doctors} \wedge (x.\text{firstName} = fn? \vee x.\text{lastName} = ln?)\}$

<i>SearchByMajor</i>
$\exists \text{SnappDoctor}$
$m? : \text{Major}$
$d! : \mathbb{P} \text{ Doctor}$
$d! = \{x : \text{Doctor} \mid x \in \text{doctors} \wedge x.\text{major} = m?\}$

^

در این بخش عملیات ثبت نام داروخانه را توصیف میکنیم. در ابتدا باید چک کنیم که پروانه کسب داروخانه معتبر است و چون این کار توسط یک third party انجام میشود ما نیز از یک تابع پیشفرض برای اعتبار سنجی آن استفاده میکنیم به نام `checkLicense`. در ضمن باید بررسی کنیم که هیچ دو داروخانه ای پروانه کسب یکسان نداشته باشند.

<i>RegisterPharmacy</i>
$\Delta \text{SnappDoctor}$
$p? : \text{Pharmacy}$
$\text{checkLicense}(p?.\text{license})$
$\forall p : \text{Pharmacy} \mid p \in \text{pharmacies} \bullet p.\text{license} \neq p?.\text{license}$
$\text{doctors}' = \text{doctors}$
$\text{patients}' = \text{patients}$
$\text{callReserve}' = \text{callReserve}$
$\text{calledDoctor}' = \text{calledDoctor}$
$\text{textDoctor}' = \text{textDoctor}$
$\text{balance}' = \text{balance}$
$\text{status}' = \text{status}$
$\text{pharmacies}' = \text{pharmacies} \cup p?$

<i>Success<sub>6</sub></i>
$r! : \text{Response}$
$r! = \text{PharmacyRegistered}$

<i>PharmacyNotValid</i>
$\exists \text{SnappDoctor}$
$p? : \text{Pharmacy}$
$\neg \text{checkLicense}(p?.\text{license}) \vee$ $\exists p : \text{Pharmacy} \mid p \in \text{pharmacies} \bullet p.\text{license} = p.\text{license}$

<i>Failure<sub>6</sub></i>
$r! : \text{Response}$
$r! = \text{LicenseisNotValid}$

$$\text{SignUpPharmacy} == (\text{RegisterPharmacy} \wedge \text{Success}_6) \vee (\text{PharmacyNotValid} \wedge \text{Failure}_6)$$

۹

در این بخش عملیات خرید دارو را توصیف میکنیم. اولاً باید نسخه بیمار صحت سنجی شود که این کار توسط یک تابع پیشفرض انجام می شود به نام `checkPr`. داروخانه باید تمام داروهای مدنظر مشتری را داشته باشد. مشتری نیز باید پول دارو ها را با کارت اعتباری اش پرداخت کند که این عملیات با استفاده از تابع `transaction` صورت میگیرد.

<i>PurchaseDrugs</i>
$\Delta \text{SnappDoctor}$
$d? : \mathbb{P} \text{ Drug}$
$pr? : \text{Prescription}$
$p? : \text{Pharmacy}$
$c? : \text{CreditCard}$
$m : \mathbb{N}$
$\text{checkPr}(pr?)p? \in \text{pharmacies}$
$d? \subseteq p?.\text{drugs}$
$m = \#(d?) * p?.\text{pricePerDrug}$
$\text{transaction}(c?, m)$
$\text{doctors}' = \text{doctors}$
$\text{patients}' = \text{patients}$
$\text{callReserve}' = \text{callReserve}$
$\text{calledDoctor}' = \text{calledDoctor}$
$\text{textDoctor}' = \text{textDoctor}$
$\text{balance}' = \text{balance}$
$\text{status}' = \text{status}$
$\text{pharmacies}' = \text{pharmacies}$

<i>PurchaseFaild</i>
$\exists \text{SnappDoctor}$
$d? : \mathbb{P} \text{ Drug}$
$pr? : \text{Prescription}$
$p? : \text{Pharmacy}$
$c? : \text{CreditCard}$
$m : \mathbb{N}$
$m = \#(d?) * p?.\text{pricePerDrug}$
$\neg \text{checkPr}(pr?) \vee$
$p? \notin \text{pharmacies} \vee$
$\neg \text{transaction}(c?, m) \vee$
$\neg (d? \subseteq p?.\text{drugs})$

<i>Success<sub>7</sub></i>
$r! : \text{Response}$
$r! = \text{PurchaseMade}$

<i>Failure<sub>7</sub></i>
$r! : \text{Response}$
$r! = \text{PurchaseFaild}$

$$\text{BuyDrugs} == (\text{PurchaseDrugs} \wedge \text{Success}_7) \vee (\text{PurchaseFaild} \wedge \text{Failure}_7)$$

۱۰

در این بخش عملیات جستجوی داروخانه بر اساس اسم و یا بر اساس یک مجموعه دارو را توصیف میکنیم.

<i>SearchPharmacyByName</i>
$\exists \text{SnappDoctor}$
$n? : \text{Name}$
$p! : \mathbb{P} \text{ Pharmacy}$
$p! = \{x : \text{Pharmacy} \mid x \in \text{pharmacies} \wedge x.\text{name} = n?\}$



<i>SearchPharmacyByDrugs</i>	_____
$\exists SnappDoctor$	
$d? : \mathbb{P} \textit{ Drug}$	
$p! : \mathbb{P} \textit{ Doctor}$	
<hr/>	
$p! = \{x : \textit{Pharmacy} \mid x \in \textit{pharmacies} \wedge d? \subseteq x.\textit{drugs}\}$	