

بسمه تعالی



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

سند آزمون نرم افزار

سامانه‌ی مطب هوشمند - بخش سرپرست (Admin)

گروه شماره ۵

نیما سالم ۹۷۱۰۶۰۰۲

امیرمحمد قاسمی ۹۷۱۰۰۴۹۵

یگانه قره‌داغی ۹۷۱۰۶۲۱۶

حمیدرضا کامکاری ۹۷۱۱۰۱۷۷

سیدعلیرضا هاشمی ۹۷۱۰۲۶۲۲

3	مدل آزمون پروژه
3	آزمون عملکرد ارسال پیام
3	متغیرهای ورودی
3	خصوصیات تعریف شده
4	پوشش انتخاب شده
4	نیازمندی های آزمون
4	موارد آزمون
6	آزمون عملکرد حذف کاربر از نوبت
6	متغیر های ورودی
6	پوشش انتخاب شده
6	نیازمندی های آزمون
6	خصوصیات تعریف شده
8	آزمون برنامه ی یافتن اندیس عدد در یک آرایه ی مرتب
8	خواندن و review کد
8	تولید Unit-test

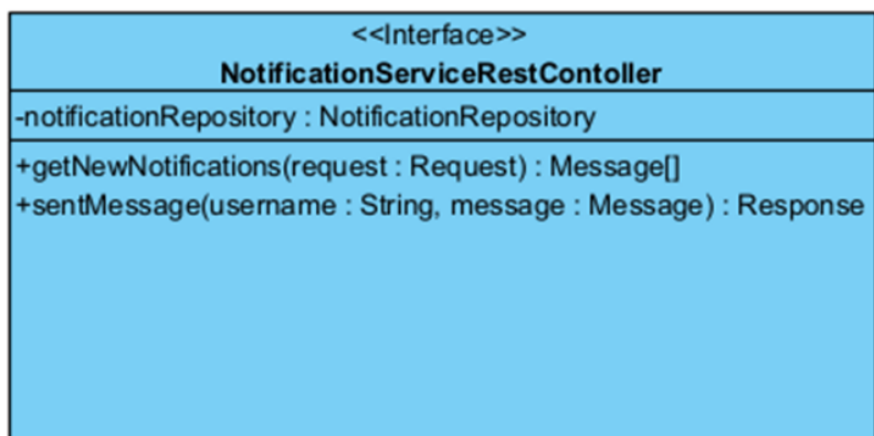
مدل آزمون پروژه

آزمون عملکرد ارسال پیام

توضیحات عملکرد «ارسال پیام» در بخش ۴ سند توصیف طراحی سامانه، قسمت واسطه‌های مربوط به Notification Management نوشته شده است. به صورت کلی، این عملکرد جهت ارسال پیام از جانب پزشک به بیمار یا بالعکس می‌باشد.

+ sendMessage(username: String: messageText: String)

NotificationServiceRestController



متغیرهای ورودی

متغیرهای ورودی این عملکرد، عبارتند از:

- نام کاربری گیرنده
- متن پیام

خصوصیات تعریف شده

در مورد نام کاربری:

- آیا کاربری با این نام کاربری وجود دارد یا ندارد.
- آیا کاربر پزشک است یا پزشک نیست.
- آیا کاربر بیمار است یا بیمار نیست.

در مورد متن پیام:

- آیا طول متن پیام بیشتر از ۳۰۰ کاراکتر است یا کمتر مساوی ۳۰۰ است.
- آیا در متن پیام از تگ‌های html استفاده شده است یا نه (جهت حفظ امنیت نرم‌افزار)

که این همه‌ی این خصوصیات، مبتنی بر عملکرد انتخاب شده‌اند.

پوشش انتخاب‌شده

از پوشش ECC (Each Choice Coverage) استفاده می‌کنیم.

نیازمندی‌های آزمون

با توجه به پوشش انتخاب شده، از هر دسته‌بندی باید یک مورد انتخاب شود و در نتیجه، نیازمندی‌های آزمون شامل مجموعه‌ای از زوج‌مرتب‌ها می‌شود که از هر کدام از دسته‌بندی‌ها، حداقل یک مورد بین زوج‌مرتب‌ها وجود دارد.

موارد آزمون

- آیدی کاربر پزشک: 81ea3204-236e-448c-87bb-19c2d13b85f3
- آیدی کاربر بیمار: 4a8b1b20-d86e-a08c-aa9b-a80c2dcdb8ea
- آیدی غیرواقعی: 81ea3204-236e-448c-87bb-19c2d13b85f4

TestcaseId: 1 Priority: کم Tracing: بررسی کاربر پزشک با متن پیام کمتر از ۳۰۰ کاراکتر که تگ اچ‌تی‌ام‌ال هم ندارد	
Input	username: 81ea3204-236e-448c-87bb-19c2d13b85f3 messageText: "سلام با عرض معذرت نوبت شما کنسل شده است"
Expected Result	پیام با موفقیت ارسال شود.
Actual Result	به توجه به این که سامانه به صورت واقعی پیاده‌سازی نشده است، در مورد جواب واقعی نمی‌توان نظری داد.

آزمون عملکرد حذف کاربر از نوبت

توضیحات عملکرد «حذف کاربر از نوبت» در بخش ۴ سند توصیف طراحی سامانه، قسمت واسط‌های مربوط به VisitSlot Controller نوشته شده است. به صورت کلی، این عملکرد جهت حذف یک کاربر از یک VisitSlot استفاده می‌شود.

+ RemovePatientFromVisitSlot(VisistSlitID: int, PatientID : int)

<<Interface>> VisitSlotController	
-visitSlotRepository : VisitSlotRepository	
+getDoctorsVisitSlots(docter username : String, startDate : DateTime, endDate : DateTime) : VisitSlot[]	
+CreateVisitSlot(docUsername : String, startDate : DateTime, endDate : DateTime) : Response	
+AddPatientToVisitSlot(visitSlotId : int, PatientId : int) : Response	
+RemovePatientToVisitSlot(visitSlotId : int, PatientId : int) : Response	

متغیر های ورودی

- شناسه ی مربوط به visitSlot
- شناسه ی مربوط به Patient

پوشش انتخاب‌شده

از پوشش (ECC (Each Choice Coverage استفاده می‌کنیم.

نیازمندی‌های آزمون

با توجه به پوشش انتخاب شده، از هر دسته‌بندی باید یک مورد انتخاب شود و در نتیجه، نیازمندی‌های آزمون شامل مجموعه‌ای از زوج مرتب‌ها می‌شود که از هر کدام از دسته‌بندی‌ها، حداقل یک مورد بین زوج‌مرتب‌ها وجود دارد.

خصوصیات تعریف شده

همگی از نوع عملکردی هستند.

Q1 = PatientdID is exists = [T, F]

Q2 = VisitSlotID is exists = [T, F]

Q3 = PatientID is registered for this visitSlot = [T, F]

برای تست تحت پوشش گفته شده نیاز است موارد زیر بررسی شوند.

$[Q1, Q2, Q3] = \{[T, T, T], [F, F, F]\}$

حالا فرض میکنیم داده ها به شکل زیر وجود دارند.

Patients = [1,2,3,4,5]

VisistSlots = [1,2,3,4,5]

(PatientID, VisitSlotid) = [(1,2), (3,5)]

Test Case ID : 4 Priority : زیاد Tracing : بیمار و نوبت هر دو وجود دارند و بیمار مرتبت با نوبت است	
Input	patientID = 1, VisitSlotID = 2
Expected Result	success
Actual Result	-

Test Case ID : 5 Priority : زیاد Tracing : بیمار و نوبت هیچکدام وجود ندارند	
Input	patientID = 6, VisitSlotID = 6
Expected Result	No patient and VisitSlot, this user is not registered for this VisitSlot
Actual Result	-

آزمون برنامه‌ی یافتن اندیس عدد در یک آرایه‌ی مرتب

خواندن و review کد

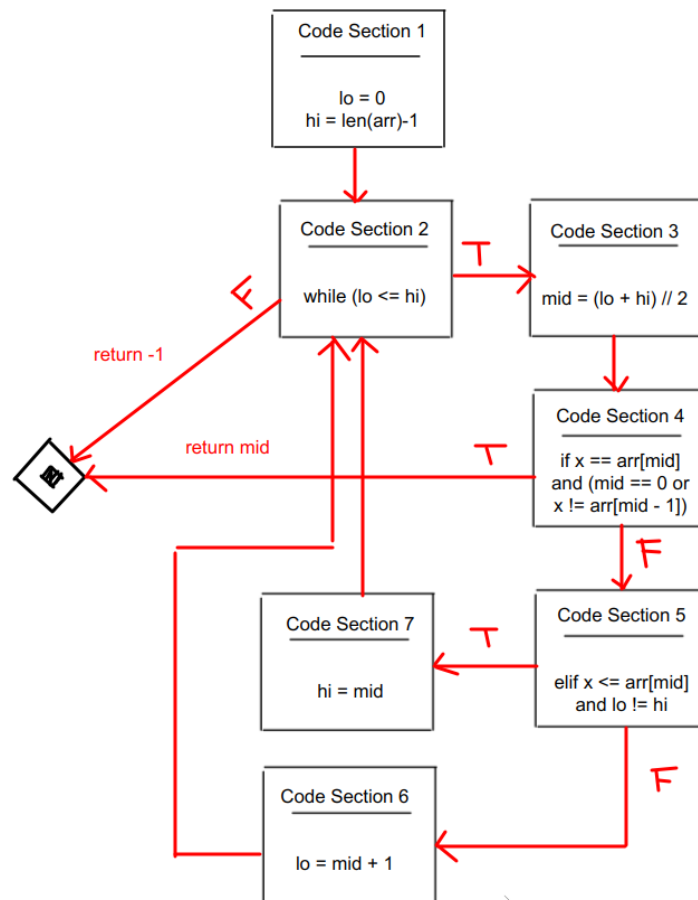
با توجه به کد می‌فهمیم مفهوم کلی هدف آن به دنبال پیدا کردن اولین نقطه ظاهر شدن x در دنباله مرتب شده‌است و چنانچه این عدد ظاهر نشده باشد مقدار -1 را برگردانده و در صورتی که ظاهر شده باشد اولین اندیس حضور آن را بر می‌گرداند.

برای اینکار ابتدا اولین اندیس آرایه که مقدارش بزرگتر یا مساوی x است پیدا می‌کند و در صورتیکه آن اندیس مقدارش برابر با x بود و اولین x در آرایه بود، یا به عبارت بهتر $arr_{mid} = x \wedge arr_{mid-1} \neq x$ برقرار باشد مقدار آن اندیس به عنوان خروجی تابع قرار می‌گیرد و در غیر اینصورت صفر خواهد بود.

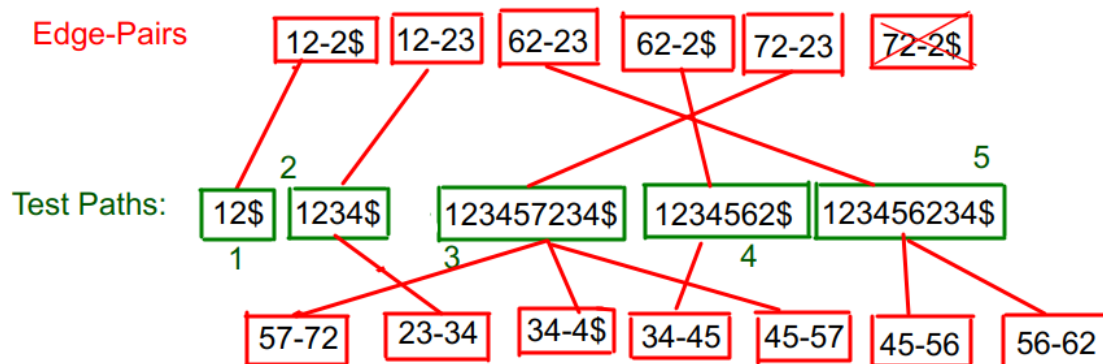
برای پیدا کردن این اندیس از یک پنجره بین lo و hi استفاده می‌شود و به صورت جست‌وجوی دودویی هر سری عضو وسط این پنجره با x مقایسه می‌شود، چنانچه مقدار آن اندیس بزرگتر یا مساوی x باشد یعنی اولین اندیس بزرگتر مساوی x در نیمه راست است و در اینصورت $hi = mid$ و در غیر اینصورت یعنی در نیمه راست نیست و $lo = mid + 1$.

تولید Unit-test

توجه کنید این یک توضیح سطح بالا از کد پیاده‌سازی شده است و کد پیاده‌سازی شده یکسری مشکلات دارد. با فرض یک کد ایده‌آل می‌توانیم به صورت زیر گراف محاسبات الگوریتم را به‌دست آوریم:



توجه کنید کد بالا کد درست شده مسئله است اما در ابتدا پیاده سازی را نادیده در نظر بگیرید و فرض کنید هر یک از رؤوس صرفاً یک رفتار طبق توضیحات ارائه شده به صورت ایده آل باید داشته باشد. حالا از پوشش جفت یال استفاده می کنیم و یکسری Test Path برای این مسئله ارائه می دهیم. در شکل زیر می توانید جفت یال ها و Test Path های متناظر که آن ها را پوشش می دهند ببینید:



اگر توجه کنید $2 - 72$ امکان‌پذیر نیست چرا که زمانی که $hi = mid$ می‌شود امکان ندارد مقدار hi از lo کوچکتر شود چرا که mid از اول بین hi و lo قرار داشته است. بنابراین تنها پوشش جفت‌های ممکن را در نظر می‌گیریم. حالا کافیه به ازای هر مسیر تست یک ورودی متناظر تولید کنیم. در جدول زیر ورودی متناظر با هر مسیر تست، خروجی کد داده شده و کد درست را می‌بینید:

شماره تست	ورودی مسئله	خروجی مورد انتظار	خروجی کد
۱	$arr = [], x = 10$	-1	Index error
۲	$arr = [1, 2, 3], x = 2$	1	1
۳	$arr = [2, 3, 4, 5, 6, 7], x = 3$	1	1
۴	$arr = [1], x = 10$	-1	Index error
۵	$arr = [2, 3, 4, 5, 6, 7], x = 6$	4	4
تست اضافه (برای چک کردن تمامی خطاهای کد)	$arr = [1], x = -1$	0	Time Limit Exceeded

برای اطلاعات بیشتر به فایل ضمیمه "find_first_in_sorted_unittest.py" مراجعه نمایید که اطلاعات تمامی اجراها را روی تمامی تست‌ها شامل می‌شود.

اسکرین‌شات اجرای همین کد روی اندیس شماره‌تست‌های مختلف در صفحه بعد ظاهر شده‌است. ما از پوشش جفت‌یالی استفاده کردیم چرا که ترکیب دوتایی‌های بلوک کدها اهمیت بالایی دارد و تعداد حالت‌های اجرای بیشتری را پوشش می‌دهد. برای مثال اگر بخواهیم روند جست‌وجوی دودویی تا بیشتر از یک‌لایه را بررسی کنیم باید جفت یال‌ها را پوشش دهیم تا حداقل دوبار حلقه اجرا تکرار شود.

نهایتاً می‌بینیم تست‌های ارائه شده به ازای کد پیشنهادی ما جواب درستی می‌دهد اما در کد ارائه شده مسئله در تست‌های مربوطه index error یا Time Limit Exceeded ایجاد می‌کند. یکی از موارد خطا این است که پنجره جست‌وجوی دودویی به صورت بسته-بسته در نظر گرفته شده‌است در حالی که در ابتدای کار مقدار $hi = len(arr)$ قرار داده شده. همچنین می‌بینیم که در صورت بسته بسته در نظر گرفتن حالت‌هایی که $hi = lo$ شود برنامه در یک لوپ نامتناهی قرار می‌گیرد (مشابه تست اضافه).

بنابراین با در نظر گرفتن تمامی تست‌ها و این حالات مشکلات برنامه را می‌توانیم حل کنیم.

```

(ma heinä-7 10:32:33)-(CPU 42,9%:0:Net 38)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 2: 0
-1
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependency
warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
Traceback (most recent call last):
  File "find_first_in_sorted.py", line 51, in <module>
    print(find_first_in_sorted_wa(tests[t][0], tests[t][1]))
  File "find_first_in_sorted.py", line 30, in find_first_in_sorted_wa
    if x == arr[mid] and (mid == 0 or x != arr[mid - 1]):
IndexError: list index out of range
(ERROR)-(Exit Code 1)-(General error)
(ma heinä-7 10:32:38)-(CPU 42,8%:0:Net 38)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 2: 1
1
1

```

```

(ma heinä-7 10:42:41)-(CPU 41,6%:0:Net 32)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 4: 0
-1
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependency
warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
Traceback (most recent call last):
  File "find_first_in_sorted.py", line 53, in <module>
    print(find_first_in_sorted_wa(tests[t][0], tests[t][1]))
  File "find_first_in_sorted.py", line 30, in find_first_in_sorted_wa
    if x == arr[mid] and (mid == 0 or x != arr[mid - 1]):
IndexError: list index out of range
(ERROR)-(Exit Code 1)-(General error)
(ma heinä-7 10:42:44)-(CPU 41,6%:0:Net 32)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 4: 1
1
1
(ma heinä-7 10:42:45)-(CPU 41,6%:0:Net 32)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 4: 2
1
1
(ma heinä-7 10:42:47)-(CPU 41,6%:0:Net 32)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 4: 3
-1
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependency
warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
Traceback (most recent call last):
  File "find_first_in_sorted.py", line 53, in <module>
    print(find_first_in_sorted_wa(tests[t][0], tests[t][1]))
  File "find_first_in_sorted.py", line 30, in find_first_in_sorted_wa
    if x == arr[mid] and (mid == 0 or x != arr[mid - 1]):
IndexError: list index out of range
(ERROR)-(Exit Code 1)-(General error)
(ma heinä-7 10:42:49)-(CPU 41,6%:0:Net 32)-(hamidreza:~/SWEng)-(12K:1)
> python3 find_first_in_sorted.py
enter test index from 0 to 4: 4
4
4

```