

بسمه تعالی

نرم افزار مدیریت داروخانه

گزارش کار و شرح فنی

استاد راهنما: دکتر محرم منصوری زاده

دانشکده ی فنی تویسرکان

ترم دوم

نیمه ی دوم سال تحصیلی ۹۵-۹۶

فهرست

2	طرح پروژه
3	چالش ها
3	ساختار داده ها
3	داده های آزمایشی (Test Data)
3	شمایل (Story Board)
6	الگوریتم
22	فرایند تست
23	شرح وظایف
24	نمودار کلاس
25	تأیید



طرح پروژه

ابتدا نرم افزار های متعددی بررسی شدند. از آنجا که استفاده و پیاده سازی classها، توابع، وراثت، اورلودینگ، اشاره گر ها، آرایه ها و ... توسط استاد راهنما مطرح گردیده و از اهمیت خاصی برخوردار بودند، سعی بر آن شد تا در طراحی برنامه به این موارد توجه ویژه ای شود.

نرم افزار مدیریت داروخانه به جهت شامل شدن بخش های زیر یکی از راهکارهای جامع توسعه ی نرم افزار به حساب می آمد:

- بارگزاری
- مشاهده ی دارو ها
- افزودن دارو ها
- ویرایش دارو ها
- حذف دارو ها
- مشاهده ی مقدار انبار
- ویرایش مقادیر موجود از هر دارو در انبار
- مشاهده ی فروش ها
- فروش دارو
- مشاهده ی گزارش فروش
- ذخیره و خروج



چالش ها

ساختار داده ها

برای ذخیره و بازیابی اطلاعات نیاز به ساختاری قابل اطمینان داشتیم که استفاده از پایگاه های داده امکان پذیر نبود، xml ساختاری پیچیده بود و همچنین دیگر ساختارها هم شامل همان پیچیدگی ها بودند.

```
<Drug>
  <id>15</id>
  <name>Abobotulinumdoxin</name>
  <amount>10000</amount>
</Drug>
```

به همین جهت از ساختار csv استفاده کردیم.

CSV یا همان Comma Separated Value، این صورت است که میتوان هر فایل را به عنوان یک جدول در نظر گرفت که سطر اول آن عنوان ستون هاست که البته ذکر عناوین ستون ها اختیاری است و این ساختار بر اساس تعداد و ترتیب قرارگیری خط ها و کاما ها استوار است.

Id, Name, Amount

15, Abobotulinumdoxin, 10000

داده های آزمایشی (Test Data)

برای تولید داده های آزمایشی از نرم افزار Microsoft Excel استفاده کردیم. برای تولید مقادیر از تابع *RANDBETWEEN(Bottom, Top)* و برای تولید سریال ها و آیدی ها از تابع *ROW()* استفاده کردیم. در نهایت فایل تولید شده را در فرمت CSV با جدا کننده (delimiter) ویرگول (comma) ذخیره سازی کرده و فایل آن ها را به فرمت دلخواه 3ms ذخیره سازی کردیم.

شمایل (Story Board)

ابتدا برنامه را روی کاغذ طراحی کردیم که اولین نسخه ی آن به شکل زیر است:



Drug Store App v1.0.0.0

1- *Drugs*

2- *Amount*

3- *Sales*

4- *Report*

1 -----

1.1 - *View*

1.2 - *Add*

1.3 - *Edit*

1.4 - *Delete*

2 -----

Please enter drug:

Please enter Amount:

3 -----

Enter Drug:

Enter Sale Amount:

Enter National Code:



4 -----

4 – View



الگورتیم

1	#include <cstdlib>	atoi
2	#include <conio.h>	Getch
3	#include <iostream>	std
4	#include <fstream>	File
5	#include <string>	String
6	#include <stdio.h>	Getline
7	#include <iomanip>	Setw
8	#include <stdlib.h>	
9		
10	using namespace std;	cin, cout
11	const string delimiter = ",";	ثابت جدا کننده
12		
13	class Drug {	تعریف کلاس
14	public:	متغیرهای عمومی
15	int Id;	شناسه ی عددی
16	string Name;	اسم دارو
17		
18	Drug()	تابع سازنده ی کلاس (constructor)
19	{	
20	Id = 0;	شناسه ی پیش فرض
21	Name = "";	نام پیش فرض
22	}	
23	Drug(int id, string name)	کانستراکتور با پارامتر های ورودی
24	{	
25	Id = id;	مقداردهی به شناسه
26	Name = name;	مقدار دهی به نام
27	}	
28	};	
29	class Amount {	تعریف کلاس
30	public:	متغیرهای عمومی
31	int DrugId;	تعریف شناسه دارو



32	int Value;	مقدار
33	Amount()	تابع سازنده کلاس (constructor)
34	{	
35	DrugId = 0;	شناسه پیش فرض دارو
36	Value = 0;	مقدار پیش فرض دارو
37	}	
38	Amount(int drugId, int value)	کانستراکتور با پارامتر های ورودی
39	{	
40	DrugId = drugId;	مقدار دهی به شناسه دارو
41	Value = value;	ذخیره به مقادیر دارو
42	}	
43	};	
44	class Sale {	تعریف کلاس
45	public:	متغیر های عمومی
46	int Id;	شناسه عددی
47	string NationalCode;	کد ملی
48	int DrugId;	شناسه دارو
49	int Amount;	مقدار
50	Sale()	تابع (constructor)
51	{	
52	Id = 0;	شناسه پیش فرض
53	Amount = 0;	مقدار اولیه
54	NationalCode = "";	مقدار دهی اولیه به کد ملی
55	DrugId = 0;	شناسه پیش فرض دارو
56	}	
57	Sale(int id, string nationalCode, int drugId, int amount)	کانستراکتور با پارامتر های ورودی
58	{	
59	Id = id;	مقدار دهی به شناسه
60	Amount = amount;	ثبت مقدار اولیه
61	NationalCode = nationalCode;	ثبت ک ملی
62	DrugId = drugId;	مقدار دهی به شناسه دارو
63	}	
64	};	



65		
66		
67	Amount * amountsList;	تعریف لیست مقادیر دارو (از کلاس Amount)
68	Drug * drugsList;	تعریف لیست دارو ها (از کلاس Drug)
69	Sale * salesList;	تعریف لیست فروش
70	unsigned int amountsCount, drugsCount, salesCount;	تعریف شمارنده ی دارو، شمارنده مقادیر دارو، شمارنده فروش ها
71		
72	void clear();	پاک کردن console
73	void _1();	تعریف تابع دارو
74	void _2();	تعریف تابع مقادیر
75	void _3();	تعریف تابع فروش
76	void _4();	تعریف تابع گزارش
77	void Save();	تابع ذخیره
78		
79	int main()	تابع اصلی
80	{	
81	system("Color 84");	تنظیم رنگ background کنسول
82		
83	clear();	فراخوانی تابع پاک کننده
84	cout << "Loading ..." << endl;	نمایش عبارت Loading...
85		
86	// Drugs Load	
87	drugsCount = 0;	شمارنده تعداد دارو
88	string unused;	متغیر موقت برای ذخیره ی خطوط فایل به جهت شمارش
89	ifstream drugsFile("Drugs.3ms");	باز کردن فایل دارو ها
90	while (getline(drugsFile, unused))	خواندن خط به خط فایل در حلقه و ریختن آن در unused
91	++drugsCount;	افزودن شمارنده
92	drugsFile.close();	بستن فایل داروها
93	drugsFile.open("Drugs.3ms");	باز کردن مجدد فایل (به جهت خواندن مجدد از خط اول)
94	drugsList = new Drug[drugsCount];	ایجاد لیست خالی داروها به تعداد شمارنده
95	string line = "";	متغیر موقت برای خواندن خط به خط فایل
96	int i = 0;	تعریف یک متغیر
97	while (i < drugsCount)	حلقه



98	{	
99	getline(drugsFile, line);	خواندن یک خط و ریختن آن در متغیر line و رفتن به خط بعد
100	int id = atoi(line.substr(0, line.find(delimiter)).c_str());	جدا کردن id از متن Atoi: تبدیل رشته به عدد C_str(): تبدیل به رشته Find(): پیدا کردن مکان پارامتر ورودی اش در رشته Substr(): جداسازی و انتخاب کردن حروف از پارامتر تا پارامتر
101	line = line.erase(0, line.find(delimiter) + delimiter.length());	Length(): طول رشته
102	string name = line;	تعریف متغیر line
103	(drugsList[i] = Drug(id, name);	مقدار دهی به اندیس نام لیست داروها
104	i++;	متغیر را اضافه کن
105	}	
106	drugsFile.close();	لیست داروها را ببند
107		
108		
109	// Sales Load	
110	salesCount = 0;	شمارنده فروش با مقدار پیش فرض 0
111	unused = "";	تعریف یک متغیر موقت
112	ifstream salesFile("Sales.3ms");	باز کردن فایل فروش
113	while (getline(salesFile, unused))	خواندن خط به خط فایل و ریختن آن در unused
114	++salesCount;	شمارنده را اضافه کن
115	salesFile.close();	فایل را ببند
116	salesFile.open("Sales.3ms");	باز کردن مجدد فایل
117	salesList = new Sale[salesCount];	ایجاد لیست خالی از فروش به تعداد شمارنده
118	line = "";	ایجاد متغیر line
119	i = 0;	ایجاد شمارنده با مقدار پیش فرض 0
120	while (i < salesCount)	تا زمانی که شمارنده از تعداد لیست کمتر بود دستورهای زیر را اجرا کن
121	{	
122	getline(salesFile, line);	خواندن خط به خط و ریختن آن در متغیر line
123	int id = atoi(line.substr(0, line.find(delimiter)).c_str());	مقدار دهی به id
124	line = line.erase(0, line.find(delimiter) + delimiter.length());	حذف کردن id از خط
125	string nationalCode = line.substr(0, line.find(delimiter));	مقدار دهی به کد ملی



126	line = line.erase(0, line.find(delimiter) + delimiter.length());	حذف کردن کد ملی از خط
127	int drugId = atoi(line.c_str());	مقدار دهی به شناسه دارو
128	line = line.erase(0, line.find(delimiter) + delimiter.length());	پاک کردن شناسه دارو از خط
129	int amount = atoi(line.c_str());	ثبت مقدار دارو
130	(salesList[i]) = Sale(id, nationalCode, drugId, amount);	مقدار دهی به سطر نام فروش ها
131	i++;	افزایش شمارنده
132	}	
133	line = "";	خالی کردن متغیر
134	salesFile.close();	بستن فایل فروش ها
135		
136		
137	// Amounts Load	
138	amountsCount = 0;	شمارنده ی موجودی انبار
139	unused = "";	متغیر موقت
140	ifstream amountsFile("Amounts.3ms");	باز کردن فایل موجودی
141	while (getline(amountsFile, unused))	خواندن خط به خط فایل در حلقه
142	++amountsCount;	افزایش شمارنده = تعداد خطوط فایل
143	amountsFile.close();	بستن فایل موجودی انبار
144	amountsFile.open("Amounts.3ms");	باز کردن "مجدد" فایل
145	amountsList = new Amount[amountsCount];	ایجاد آرایه ای خالی از موجودی انبار به عنوان لیست به طول شمارنده
146	line = "";	خالی کردن متغیر
147	i = 0;	صفر کردن شمارنده ی حلقه
148	while (i < amountsCount)	حلقه
149	{	
150	getline(amountsFile, line);	خواندن خط و ریختن آن در متغیر line
151	int id = atoi(line.substr(0, line.find(delimiter)).c_str());	مقدار دهی id
152	line = line.erase(0, line.find(delimiter) + delimiter.length());	پاک کردن شناسه از خط
153	int value = atoi(line.c_str());	ثبت مقدار موجودی
154	(amountsList[i]) = Amount(id, value);	مقدار دهی به سطر انبار موجودی
155	i++;	افزایش شمارنده
156	}	
157	amountsFile.close();	بستن فایل موجودی انبار
158	i = 0;	صفر کردن متغیر



159		
160	// App start	
161	cout << "Drug Store App v1.0.0.0" << endl;	چاپ عنوان برنامه
162	cout << "Press any key to continue..." << endl;	چاپ عبارت مقابل
163	bool AppClosed = false;	مشخص میکند آیا برنامه بسته شده است یا نه
164	while (!AppClosed)	تا زمانی متغیر فوق نشان میدهد که برنامه باز است برنامه دائماً به کار خود ادامه میدهد
165	{	
166	getch();	منتظر ورودی کاربر
167	clear();	پاک کردن صفحه
168	cout << "1- Drugs" << endl;	چاپ کن گزینه 1
169	cout << "2- Amount" << endl;	چاپ کن گزینه 2
170	cout << "3- Sales" << endl;	چاپ کن گزینه 3
171	cout << "4- Report" << endl;	چاپ کن گزینه 4
172	cout << "5- Exit" << endl;	چاپ کن گزینه 5
173		
174	int Choice;	
175	cin >> Choice;	گرفتن منوی انتخابی کاربر
176		
177	switch (Choice)	سوئیچ به ازای انتخاب کاربر
178	{	
179	case 1:	منوی 1 = دارو
180	_1();	
181	break;	
182	case 2:	منوی 2 = موجودی انبار
183	_2();	
184	break;	
185	case 3:	منوی 3 = فروش
186	_3();	
187	break;	
188	case 4:	منوی 4 = گزارش
189	_4();	
190	break;	



191	case 5:	منوی 5 = خروج
192	clear();	پاک کردن صفحه
193	Save();	ذخیره صفحه
194	cout << "Goodbye!" << endl;	چاپ کن عبارت مقابل
195	cout << "Press any key to exit";	چاپ کن عبارت مقابل
196	AppClosed = true;	آیا برنامه بسته شده = درست
197	break;	
198		
199	}	
200		
201	}	
202		
203	getch();	منتظر ورودی کاربر
204	return 0;	بستن برنامه با خاتمه موفقیت آمیز تابع اصلی
205	}	
206	void clear() {	
207	system("cls");	استفاده از سینتکس cls خط فرمان ویندوز که وظیفه آن پاک سازی کنسول است
208	}	
209	void _1() {	
210	clear();	
211	cout << "1- View" << endl;	
212	cout << "2- Add" << endl;	
213	cout << "3- Edit" << endl;	
214	cout << "4- Delete" << endl;	
215		
216	int Choice;	
217	cin >> Choice;	
218	clear();	
219	switch (Choice)	
220	{	
221	case 1:	
222	{	



223	// View Drugs	
224	cout << " " << left << setw(10) << "Drug Code" << " " << "Drug Name" << endl;	چاپ سطر عناوین جدول (Setw()) به اندازه ی پارامتر ورودی فاصله در متن ایجاد میکند.
225	for (int i = 0; i < drugsCount; i++)	حلقه با شمارنده i
226	{	
227	cout << " " << left << setw(10) << drugsList[i].Id << " " << drugsList[i].Name << endl;	چاپ داروی نام
228	}	
229	break;	
230	}	
231		
232	case 2:	
233	{	
234	// Add Drugs	دستورات این کیس در واقع لیستی از همان نوع دارو ها اما با تعداد یکی بیشتر از لیست قبلی ایجاد میکند (temp). سپس توسط حلقه ی اول محتویات لیست قبلی را وارد لیست موقت (temp) میکند. پس از آن (خط 248) مجدد لیست اولیه را خالی کرده، آن را با تعداد جدید و اعضای خالی باز سازی میکند. در نهایت با حلقه ای عکس حلقه ی اول عمل کرده و محتویات را از لیست موقت به لیست قبلی وارد میکند.
235	drugsCount++;	
236	Drug * temp = new Drug[drugsCount];	
237	for (int i = 0; i < drugsCount - 1; i++)	
238	{	
239	temp[i] = drugsList[i];	
240	}	
241	cout << "Id: ";	
242	int id;	
243	cin >> id;	
244	cout << "Name: ";	
245	string name;	
246	cin >> name;	
247	temp[drugsCount - 1] = Drug(id, name);	
248	drugsList = new Drug[drugsCount];	



249	for (int i = 0; i < drugsCount; i++)	
250	{	
251	drugsList[i] = temp[i];	
252	}	
253	// delete temp;	
254	cout << "Done!";	
255	break;	
256	}	
257	case 3:	
258	// Edit Drug	دستورات کیس حاضر ابتدا شناسه ی داروی مورد نظر را از کاربر دریافت میکند. سپس با استفاده از یک حلقه در لیست دارو ها جستجو میکند. اگر شناسه ی مورد نظر یافت شد، نام جدید را از کاربر دریافت کرده سپس داروی بازیابی شده را با مقادیر جدید بازسازی میکند.
259	cout << "Id: ";	
260	int id;	
261	cin >> id;	
262	for (int i = 0; i < drugsCount; i++)	
263	{	
264	if (drugsList[i].Id == id)	
265	{	
266	cout << "Name: ";	
267	string name;	
268	cin >> name;	
269	drugsList[i] = Drug(id, name);	
270	cout << "Done!";	
271	}	
272	}	
273	break;	
274	case 4:	
275	{	
276	// Delete Drug	دستورات کیس زیر در واقع لیستی با تعداد یکی کمتر از لیست اولیه ایجاد کرده و شروع به انتقال رکورد ها به جدول جدید میکند. در همین حین اگر شناسه ی مورد نظر در



		فایل پیدا شد (found == true)، از آن رکورد را با افزایش اندیس (خط 290) عبور میکند. در واقع آن رکورد را به جدول جدید انتقال نمی دهد. در نهایت با استفاده از حلقه ی دوم، محتویات لیست موقت را به لیست اولیه با اندازه ی جدید منتقل میکند.
277	cout << "Id: ";	
278	int id;	
279	cin >> id;	
280	drugsCount--;	
281	Drug * temp = new Drug[drugsCount];	
282	bool found = false;	
283		
284	for (int i = 0; i < drugsCount; i++)	
285	{	
286	if (drugsList[i].Id == id)	
287	{	
288	found = true;	
289	}	
290	temp[i] = drugsList[found ? i + 1 : i];	
291	}	
292		
293	drugsList = new Drug[drugsCount];	
294	for (int i = 0; i < drugsCount; i++)	
295	{	
296	drugsList[i] = temp[i];	
297	}	
298	cout << "Done!";	
299	break;	
300	}	
301	}	
302	}	
303	void _2() {	
304	clear();	
305	cout << "1- View" << endl << "2- Edit" << endl;	



306	int myCase = 0;	
307	cin >> myCase;	
308	clear();	
309	switch (myCase)	
310	{	
311	case 1:	
312	{	
313	// View amounts	
314	cout << " " << left << setw(10) << "Drug Code" << " " << "Amount" << endl;	
315	for (int i = 0; i < amountsCount; i++)	
316	{	
317	cout << " " << left << setw(10) << amountsList[i].DrugId << " " << amountsList[i].Value << endl;	
318	}	
319	break;	
320	}	
321	case 2: {	
322	// Edit amounts	دستورات کیس حاضر ابتدا شناسه ی مقدار موجودی انبار مورد نظر را از کاربر دریافت میکند. سپس با استفاده از یک حلقه در لیست مقدار موجودی انبار ها جستجو میکند. اگر شناسه ی مورد نظر یافت شد، مقدار جدید را از کاربر دریافت کرده سپس مقدار موجودی انبار بازبینی شده را با مقادیر جدید بازسازی میکند.
323		
324	cout << "Drug Code: ";	
325	int drugId;	
326	cin >> drugId;	
327	for (int i = 0; i < amountsCount; i++)	
328	{	
329	if (amountsList[i].DrugId == drugId)	
330	{	
331	cout << "Value: ";	
332	int value;	



333	cin >> value;	
334	amountsList[i] = Amount(drugId, value);	
335	cout << "Done!";	
336	}	
337	}	
338	break;	
339	}	
340	}	
341	}	
342	void _3() {	
343	clear();	
344	cout << "1- View" << endl << "2- Add" << endl;	
345	int myCase = 0;	
346	cin >> myCase;	
347	clear();	
348	switch (myCase)	
349	{	
350	case 1:	
351	{	
352	// View sales	
353	cout << " " << left << setw(15) << "Factor Number" << " " << left << setw(10) << "Drug Code" << " " << left << setw(20) << "NationalCode" << " " << "Amount" << endl;	
354	for (int i = 0; i < salesCount; i++)	
355	{	
356	cout << " " << left << setw(15) << salesList[i].Id << " " << left << setw(10) << salesList[i].DrugId << " " << left << setw(20) << salesList[i].NationalCode << " " << salesList[i].Amount << endl;	
357	}	
358	break;	
359	}	
360	case 2:	
361	{	



	// Add sale	دستورات این کیس در واقع لیستی از همان نوع فروش ها اما با تعداد یکی بیشتر از لیست قبلی ایجاد میکند (temp). سپس توسط حلقه ی اول محتویات لیست قبلی را وارد لیست موقت (temp) میکند. پس از آن مجدد لیست اولیه را خالی کرده، آن را با تعداد جدید و اعضای خالی باز سازی میکند. در نهایت با حلقه ای عکس حلقه ی اول عمل کرده و محتویات را از لیست موقت به لیست قبلی وارد میکند.
362		
363	salesCount++;	
364	Sale * temp = new Sale[salesCount];	
365	for (int i = 0; i < salesCount - 1; i++)	
366	{	
367	temp[i] = salesList[i];	
368	}	
369	cout << "Id: ";	
370	int id;	
371	cin >> id;	
372	cout << "National Code: ";	
373	string nationalCode;	
374	cin >> nationalCode;	
375	int drugId;	
376	cout << "Drug Code: ";	
377	cin >> drugId;	
378	int amount;	
379	cout << "Amount: ";	
380	cin >> amount;	
381	temp[salesCount - 1] = Sale(id, nationalCode, drugId, amount);	
382	salesList = new Sale[salesCount];	
383	for (int i = 0; i < salesCount; i++)	
384	{	
385	salesList[i] = temp[i];	
386	}	
387	cout << "Done!";	
388	break;	



389	}	
390	}	
391		
392	}	
393	void _4() {	
394	// Report	
395	cout << " " << left << setw(15) << "Factor Number" << " " << left << setw(85) << "Drug Name" << " " << left << setw(20) << "NationalCode" << " " << "Amount" << endl;	
396	for (int i = 0; i < salesCount; i++)	
397	{	
398	cout << " " << left << setw(15) << salesList[i].Id << " " << left << setw(85);	
399	// drug name	Join حلقه ی زیر بر اساس شناسه ی دارو در لیست داروها جستجو میکند و اسم دارویی را که با آن شناسه ثبت شده است را جایگزین میکند.
400	for (int j = 0; j < drugsCount; j++)	
401	{	
402	if (drugsList[j].Id == salesList[i].DrugId)	
403	{	
404	cout << drugsList[j].Name;	
405	}	
406	}	
407	cout << " " << left << setw(20) << salesList[i].NationalCode << " " << salesList[i].Amount << endl;	
408	}	
409	}	
410	void Save() {	تابع ذخیره سازی
411	// TODO	
412		
413	// Drugs Save	
414	ofstream drugsFile("Drugs.3ms");	باز کردن فایل برای نوشتار
415	for (int i = 0; i < drugsCount; i++)	حلقه
416	{	
417	drugsFile << drugsList[i].Id << "," << drugsList[i].Name << endl;	افزودن داروی سطر نام به فایل دارو



418	}	
419	drugsFile.close();	بستن فایل دارو
420	// Amount Save	
421	ofstream amountsFile("Amounts.3ms");	
422	for (int i = 0; i < amountsCount; i++)	
423	{	
424	amountsFile << amountsList[i].DrugId << "," << amountsList[i].Value << endl;	
425	}	
426	amountsFile.close();	
427	// Sales Save	
428	ofstream salesFile("Sales.3ms");	
429	for (int i = 0; i < salesCount; i++)	
430	{	
431	salesFile << salesList[i].Id << "," << salesList[i].NationalCode << "," << salesList[i].DrugId << "," << salesList[i].Amount << endl;	
432	}	
433	salesFile.close();	
434	// Reports Save	
435	ofstream reportFile("Report.html");	باز کردن فایل گزارش
436	reportFile << "<!doctype html><html><head><meta charset='utf-8'><title>Drug Store</title><style>";	اضافه کردن قالب استاندارد HTML به فایل گزارش
437	reportFile << "table { border-collapse: collapse; text-align: left; width: 100%; } {font: normal 12px/150% Arial, Helvetica, sans-serif; background: #fff; overflow: hidden; border: 1px solid #991821; -webkit-border-radius: 3px; -moz-border-radius: 3px; border-radius: 3px; }table td, table th { padding: 3px 10px; }table thead th {background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05, #991821), color-stop(1, #80141C));background:-moz-linear-gradient(center top, #991821 5%, #80141C 100%);filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#991821', endColorstr='#80141C');background-color:#991821; color:#FFFFFF; font-size: 15px; font-weight: bold; border-left: 1px solid #B01C26; } table thead th:first-child { border: none; }table tbody td { color: #80141C; border-left: 1px solid #F7CDCD;font-size: 12px;font-weight: normal; }table tbody .alt td { background: #F7CDCD; color: #80141C; }table tbody td:first-child { border-left: none; }table tbody tr:last-child td { border-bottom: none; }table tfoot td div { border-top: 1px solid #991821;background: #F7CDCD;} table tfoot td { padding: 0; font-size: 12px } table tfoot td div{ padding: 2px; }table tfoot td ul { margin: 0; padding:0; list-style:	اضافه کردن تم CSS به قالب



	<pre> none; text-align: right; }table tfoot li { display: inline; }table tfoot li a { text- decoration: none; display: inline-block; padding: 2px 8px; margin: 1px;color: #FFFFFF;border: 1px solid #991821;-webkit-border-radius: 3px; -moz-border- radius: 3px; border-radius: 3px; background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05, #991821), color-stop(1, #80141C));background:-moz- linear-gradient(center top, #991821 5%, #80141C 100%);filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#991821', endColorstr='#80141C');background-color:#991821; }table tfoot ul.active, table tfoot ul a:hover { text-decoration: none;border-color: #80141C; color: #FFFFFF; background: none; background-color:#991821;}div.dhtmlx_window_active, div.dhx_modal_cover_dv { position: fixed !important; }"; </pre>	
438	<pre> reportFile << "</style></head><body><table><tbody><tr><th>Factor Number</th><th>Drug Name</th><th>National Code</th><th>Amount</th></tr>" << endl; </pre>	اضافه کردن سر ستون به جدول
439	<pre> for (int i = 0; i < salesCount; i++) </pre>	اضافه کردن داروها به جدول
440	<pre> { </pre>	
441	<pre> reportFile << "<tr><td>" << salesList[i].Id << "</td><td>"; </pre>	
442	<pre> // drug name </pre>	Join حلقه ی زیر بر اساس شناسه ی دارو در لیست داروها جستجو میکند و اسم دارویی را که با آن شناسه ثبت شده است را جایگزین میکند.
443	<pre> for (int j = 0; j < drugsCount; j++) </pre>	
444	<pre> { </pre>	
445	<pre> if (drugsList[j].Id == salesList[i].DrugId) </pre>	
446	<pre> { </pre>	
447	<pre> reportFile << drugsList[j].Name; </pre>	
448	<pre> } </pre>	
449	<pre> } </pre>	
450	<pre> reportFile << "</td><td>" << salesList[i].NationalCode << "</td><td>" << salesList[i].Amount << "</td></tr>" << endl; </pre>	
451	<pre> } </pre>	
452	<pre> reportFile << "</tbody></table></body></html>"; </pre>	بستن جدول و خاتمه ی قالب
453	<pre> reportFile.close(); </pre>	بستن فایل جدول
454	<pre> } </pre>	



فرایند تست

برای تست این نرم افزار از 3 فایل ورودی تست که با استفاده از نرم افزار Microsoft Excel تولید شده بودند استفاده کردیم و پس از آن نرم افزار را با استفاده از داده های دستی مکررا اجرا نمودیم. لازم به ذکر است که در این بین نرم افزار را با ورودی های غیر قابل انتظار اجرا و تست کردیم که جزئیات آن به شرح زیر است.

عنوان	نتیجه	دفعات
تست با ورودی های استاندارد	موفقیت آمیز	100
تست نرم افزار بدون ورودی	ناموفق - عدم نمایش اطلاعات	10
تست بدون دیتای ورودی داروها	ناموفق - عدم تولید گزارش	10
تست با ورودی یک میلیون فاکتور فروش	نامعلوم - وابسته به منابع سیستم	50

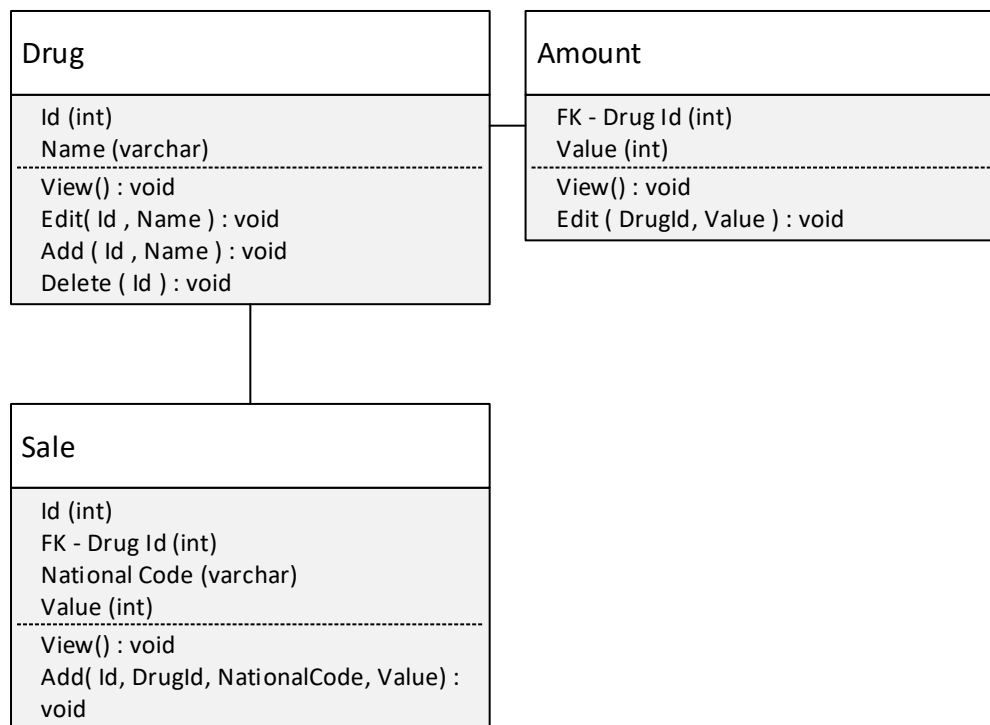


شرح وظایف

اعضای گروه	حوزه کاری	ساعت کاری
محمد رضا طیبی	مدیریت فایل و گزارش	10
محمد بهرامی	تست دیتا و ساختار فایل	10
مسعود طاهری	کلاس ها و تست کد	10
سامان سلیمانی	توابع و اجرا و تست جامع	10



نمودار کلاس



تأیید

این گزارش توسط تیم توسعه دهندگان تأیید میگردد.

محمد بهرامی mohaNadbahrami72@gmail.com +989380444649	محمدرضا طیبی reXa@gordarg.com +989388063351
	
سامان سلیمانی Soleymanisaman7575@gmail.com +989391378490	مسعود طاهری Masoudtaheri8498@gmail.com +989182168498

با تشکر و احترام

تیم توسعه دهندگان

دانشگاه بوعلی سینا همدان

دانشکده ی فنی توپسرکان

بهار ۹۶

