



# پروژه ربات فروشنده هوشمند

## G2S2R

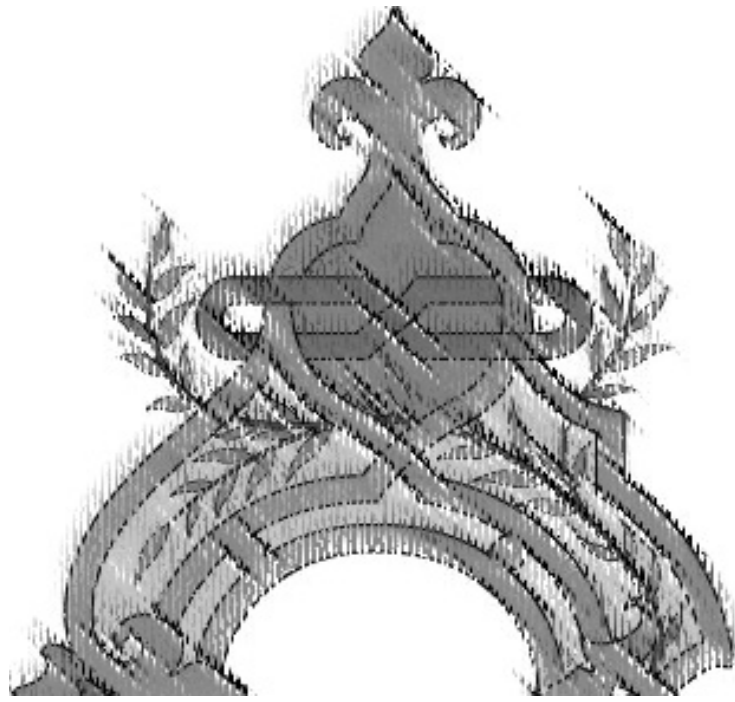
(Generous Greedy Smart Seller Robot)

مطالعه موردی: فروش تلفن همراه (موبایل)

استاد راهنما: دکتر محمد باقر منهج

### اعضای پروژه (به ترتیب حروف الفبا)

- ۱ محسن اسماعیلی
- ۲ عاطفه بابایی
- ۳ صادق حسینی
- ۴ حسن حکمی
- ۵ رضوان خاندانی
- ۶ پگاه خداکرمی
- ۷ حوری رضوی
- ۸ مهدی عزیزی
- ۹ مریم مجاور رضایی
- ۱۰ مجتبی هروی



گزارش اول  
**First Report**



## مقدمه

ربات‌های هوشمند را می‌توان حلقه اتصال تجارت و تکنولوژی در مکان‌های عمومی نامید. ربات‌ها جدیدترین مرحله تلاش انسان جهت صنایع اتوماتیک به شمار می‌روند. ربات‌ها آن دسته از ماشین‌های ساخت بشر هستند که لزوماً حرکت‌هایی شبیه انسان ندارند ولی توان تصمیم‌گیری و ایجاد و کنترل فعالیت‌های از پیش تعیین شده را دارند. از جمله ربات‌هایی که تاکنون ساخته شده‌اند عبارتند از:

- ربات پرستار هوشمند

- ربات هوشمند زیرآبی (AUV)

- جاروبرقی اتوماتیک فوق هوشمند رباتیک

- ربات هنرمند

- ربات دوچرخه سوار

- ربات انسان‌نمای «تیم شهید شهریار» که تاکنون هوشمندترین ربات ارائه شده در خاورمیانه با الگوریتم‌های پردازش تصویر و هوش مصنوعی می‌باشد که بیش از ۲۰ کارایی منحصر به فرد در آن تعریف شده است. این ربات قابلیت امدادگری دارد و می‌تواند در هر مسیر ناهموار حرکت کند. از دیگر قابلیت‌های آن عبارتند از: تشخیص بیش از هزار چهره با ذکر نام فرد، سخن گفتن و گفت‌وگوی هوشمند با افراد و شناسایی موقعیت‌های مختلف، شناسایی سریع افراد با الگوریتم جستجوی سریع، جستجو و ردیابی اجسام متحرک، انجام دستورات و فرامین پیچیده به زبان فارسی، مکالمه هوشمند، پاسخگویی به سوالات، دارا بودن منطق یادگیری و آموزش و تعلیم به دیگران، تشخیص مواد رادیواکتیو نیروگاه‌های هسته‌ای، دارا بودن احساسات و تشخیص نوازش که در مقابل تندخویی و نرمش نیز عکس‌العمل نشان می‌دهد.

...

ما نیز در این پروژه قصد داریم رباتی بسازیم که بتواند مانند یک فروشنده موبایل اعمال تصمیم‌گیری، مدیریت و بازاریابی را به صورت هوشمند انجام دهد و همواره با انتخاب بهترین استراتژی، سود خود و رضایت مشتری را حداکثر کند.

در ابتدا برای انجام کارها به طور موازی، افراد به سه گروه تقسیم شدند که تقسیم وظایف و کدنویسی سریعتر انجام شود:

گروه ۱	گروه ۲	گروه ۳
۱- حسن حکمی (سرگروه)	۱- پگاه خداکریمی (سرگروه)	۱- حوری رضوی (سرگروه)
۲- مجتبی هروی	۲- مریم رضایی	۲- رضوان خاندانی
۳- مهدی عزیزی	۳- محسن اسماعیلی	۳- عاطفه بابایی
۴- صادق حسینی		

در مراحل بعد با تشکیل جلسات متعدد و ارتباط از طریق سایت و Email با حضور همه اعضا به بررسی ابعاد مختلف پروژه، تعریف مسئله، تقسیم وظایف و ... پرداختیم. از آن جایی که یکی از مسائل مهم در پروژه‌ها مستندسازی «Documentation» می‌باشد، در این گزارش و گزارش‌های آینده هرآنچه که توسط افراد پروژه به صورت فردی و یا گروهی انجام شده است، بیان می‌شود.

## جلسه اول

در اولین جلسه که در تاریخ ۹۱/۷/۲۷ با حضور دکتر منهاج تشکیل شد، ابتدا بحث‌هایی در مورد یک نام مناسب برای پروژه، که به نوعی بیانگر هدف پروژه نیز باشد، انجام شد که در نهایت با راهنمایی دکتر، نام زیر انتخاب شد:

«پروژه ربات‌های فروشنده هوشمند (G2S2R)»

«Generous Greedy Smart Seller Robots»

سپس به بررسی ورودی‌ها (Inputs)، محدودیت‌ها (Constraints) و ماژول‌ها (Modules) پرداختیم. نتایج به صورت زیر می‌باشد:

- فضای فیزیکی و متراژ باید مشخص شود تا افراد داخل و بیرون فروشگاه از هم تمیز داده شوند.
- ساعت کاری مشخص باشد که بتوان برای سرویس دادن به مشتری‌ها برنامه‌زمان‌بندی در نظر گرفت.
- دخالت دادن ریسک‌پذیری، به عنوان مثال در نظر گرفتن شهرت افراد
- تشکیل یک پایگاه داده اولیه از کالاهای مورد نظر و به‌روز رسانی آن
- قوانین اگر- آنگاه (if-then rules)
- مشخصات ظاهری افراد به عنوان ورودی
- تعیین Module ها و subModule ها و استخراج ساختار درختی آن‌ها
- انیمیشن، شامل: نمایش فروشگاه، افراد و ورود و خروجشان، میزان درآمد و سود در هر مرحله و هر روز
- ...

و در نهایت نکته مهمی که باید در نظر گرفته شود این است که:

پیچیدگی‌های فرهنگی، اجتماعی و آداب و رسوم جوامع خصوصاً در وضع متغیرهای زبانی بسیار تأثیرگذارند. بنابراین در این پروژه نیز ما باید آداب و رسوم، قوانین عرفی و ساختار اجتماعی جامعه خود را در نظر بگیریم و توجه داشته باشیم که این پروژه باید بتواند پاسخگوی چنین جامعه‌ای باشد.

## جلسه دوم

جلسه دوم که در واقع اولین جلسه طوفان مغزی (Brain Storming) نیز بود در تاریخ ۹۱/۸/۳ تشکیل شد که قرار بود به منظور بررسی سناریوهای رفتاری یک ربات هوشمند و نزدیک شدن به مرحله پیاده‌سازی، یکسری تحقیقات میدانی از فروشگاه‌های مختلف انجام شود و اطلاعات به دست آمده به اشتراک گذاشته شود. در این مرحله یکسری سؤالات کلی مطرح شد:

- ۱- چقدر بودجه برای گوشی در نظر دارید؟
- ۲- آیا brand خاصی مورد نظرتان است؟
- ۳- از چه نوع Application‌هایی استفاده می‌کنید؟ (۵ نرم‌افزار به نمایندگی از فرد پرسیده می‌شود)
- ۴- آیا ظاهر گوشی برایتان مهم است؟

- ۵- چه نوع کاربری از گوشی مدنظرتان است؟
- ۶- آیا از گوشی برای اتصال به اینترنت استفاده می‌کنید؟
- ۷- آیا گرافیک گوشی برایتان مهم است؟ (مثلاً game زیاد بازی می‌کنید؟)
- ۸- آیا موزیک زیاد گوش می‌کنید؟
- ۹- OS (سیستم عامل) گوشی برایتان مهم است؟
- ۱۰- می‌خواهید گوشی مردانه داشته باشید یا زنانه؟
- ۱۱- آیا طول عمر گوشی برایتان مهم است؟
- ۱۲- آیا میزان کارایی باتری گوشی برایتان مهم است؟ (به‌عنوان مثال میزان Talk Time)

۱۳- در مورد مشخصات زیر کدام موارد برایتان مهم است و اولویت دارد:

- زمان ورود به بازار (چقدر از عرضه آن به بازار گذشته است)
  - اندازه صفحه نمایش
  - صفحه touch یا دکمه‌دار
  - size و وزن گوشی
  - تناسب بین وزن گوشی و اندازه صفحه نمایش
  - رنگ گوشی
  - کیفیت بدنه گوشی
  - کیفیت دوربین
  - keyboard از نوع ساده یا QWERTY
  - اندازه حافظه گوشی (داخلی یا حافظه افزوده)
  - میزان دریافت message (به طور کلی inbox و contacts)
  - قابلیت‌های جانبی
  - گارانتی
  - جنس keyboard و سهولت استفاده از آن (داشتن برچسب فارسی، بزرگ بودن دکمه‌ها و ...)
  - bluetooth
  - سهولت دسترسی به نرم‌افزار (از طریق دانلود رایگان و ...)
  - UserFriendly بودن گوشی
- همچنین در نظر گرفتن نکاتی مثل:

- در مواقعی که گوشی خاصی مورد نظر خریدار است که یک ویژگی منحصر به فرد (مثل کیفیت دوربین بالا) دارد، این آیتم را برای خریدار **bold** کنید و به او تبریک بگویید به خاطر حسن انتخابش.
- اگر کسی در هنگام سؤال پرسیدن بیش از k سؤال پرسید، ز گوشی را به او معرفی کنید. (هر چه سخت‌پسندتر، تعداد کمتر)

خانم عاطفه بابایی به عنوان گزارش کار اول (PaperWork#1)، فایلی را در این زمینه آماده کردند که به شرح زیر است:

## قوانین فروشندگی

### تعریف

- فروشندگی یعنی فراز و فرود چراکه اقتصاد کشور و تمام دنیا مدام در حال بالا و پایین شدن است.
- هم اکنون تعداد کارخانه‌هایی که محصولی را تولید می‌کنند یا شرکت‌هایی که خدماتی را ارائه می‌دهند، بسیار زیاد شده و کالایی را نمی‌توانید پیدا کنید که در انحصار یک شرکت باشد، بنابراین مزیت‌های رقابت در چیزهای دیگری تعریف می‌شود از جمله روش‌های فروش، فروشندگان و سیاست‌هایی که شرکت‌ها برای فروش در نظر می‌گیرند.
- «یک فروشنده خوب می‌تواند هر چیزی را بفروشد» زیرا که قدرت در فروش و فروشندگی، توانایی برانگیختن دیگران است نه در میزان دانش و سطح تحصیلات آکادمیک فروشنده. گرچه این دانش می‌تواند زمینه‌ساز مهارت ایجاد انگیزش در خریدار باشد(دنی، موفقیت در فروش).

### مراحل فروشندگی

- چیت وود در کتاب فروش حرفه‌ای فروش پیگیر را طی هفت مرحله مطرح می‌کند که عبارتند از:
  ۱. مرحله نزدیکی (تاثیر مثبت گذاشتن در مشتری)
  ۲. مرحله صلاحیت و شایستگی (که هدف ارزیابی مشتری، کشف مشکلات و نیازهای مشتری است)
  ۳. مرحله توافق در مورد نیاز (مرور اطلاعات دریافتی از مشتری و هماهنگی بین فروشنده و مشتری)
  ۴. مرحله فروش کمپانی (جا انداختن صلاحیت و صداقت شرکت در ذهن مشتری)
  ۵. مرحله برآورده کردن نیاز (ربط ویژگی‌های محصول با مزایا و منافع مشتری)
  ۶. مرحله گرفتن تعهد (مرحله خاتمه فروش)
  ۷. مرحله تثبیت فروش (تداوم رابطه با مشتری برای دوران بلندمدت)

### نحوه برخورد صحیح با مشتری (بایدها)

۱. سریع و به نوبت به مشتری‌های خود رسیدگی کنید.
۲. اگر تعداد مشتری‌ها بالا رفت از همکارانتان کمک بگیرید.
۳. باید اطلاع کافی از محصولات شرکت داشته باشید. همچنین باید بدانید که هر محصول کجا گذاشته شده است.
۴. همیشه با لبخند به مشتری‌ها خوشامدگویی کنید.
۵. یادتان باشد که حق همیشه با مشتری است حتی وقتی حق با آن‌ها نباشد.
۶. همیشه از مشتری سوال کنید که به کمک شما نیاز دارند یا خیر.
۷. همیشه بهترین محصولات خود را به مشتری پیشنهاد کنید.
۸. درمقابل مشتری‌های سالخورده یا ناتوان جسمی توجه بیشتری نشان دهید.
۹. اگر پاسخی برای سوال مشتری ندارید، یک پاسخ پیدا کنید. از اینکه پاسخ را از مدیرتان بپرسید واهمه نداشته باشید.
۱۰. سخت کار کنید و همیشه لبخند بر لب داشته باشید.

## نحوه برخورد صحیح با مشتری (نبایدها)

۱. هیچوقت با مشتری وارد بحث نشوید.
۲. هیچوقت مشتری را آزرده خاطر نکنید.
۳. هیچوقت به مشتری‌ها دروغ نگوئید. حتی اگر یک دروغ کوچک باشد.
۴. هیچوقت به مشتری‌ها نخندید و مسخره‌شان نکنید.
۵. با محصولات بازی نکنید.
۶. مشتری را زیاد معطل نگذارید چون ممکن است صبرش تمام شود. اگر نیاز به کمک داشتید، حتماً کمک بگیرید.
۷. در مقابل مشتری‌ها با همکارانتان غیبت نکنید و خنده راه نیندازید.
۸. با مشتریان سالخورده یا ناتوان جسمی بدرفتاری نکنید.
۹. مشکلات شخصیتان را وارد کار نکنید.
۱۰. به مشتری بی‌محلی نکنید.

## نحوه صحبت کردن با مشتری

آمار نشان می‌دهد که تاثیر اولیه که شخصی روی شما می‌گذارد عبارت است از:

- ۵۵ درصد غیرکلامی است: ظاهر و زبان بدن
- ۳۸ درصد تن صدا
- ۷ درصد کلامی است: کلماتی که به زبان می‌آورید.

## کلمات تاثیر عمیقی روی افراد دارد.

## نحوه صحبت کردن با مشتری

• «اجازه دارم؟»

از مشتری‌هایتان اجازه بگیرید. اینکار باعث می‌شود احساس خاص بودن کنند.

• «همانطور که می‌دانید...»

این چند کلمه اگر در ابتدای حرفی به زبان آید نشان می‌دهد که مشتری شما اطلاعات و دانش کافی در مورد محصول یا خدمات شما دارد و شما فقط آن نکات را برای او یادآور می‌شوید.

• «متشکر می‌شوم اگر...»

این کلمات را می‌توانید زمانی استفاده کنید که می‌خواهید مشتری برایتان کاری انجام دهد که می‌تواند مفید باشد مثلاً «متشکر می‌شوم اگر دفعه بعد که به ما سر زدید آگاهمان کنید که دخترخانمان این دستبند را دوست داشته است یا نه...»

• «خواهش می‌کنم»

• «باعث افتخار است که...»

اجازه بدهید مشتری‌تان بداند که حاضرید برای رفاه حال او فراتر از مسئولیتتان هم خدمت کنید.

• «متشکرم»

این کلمه را از ته قلب به زبان آورید نه طوطی‌وار.

- بن فلدمن، فروشنده‌ای که برای شرکت بیمه نیویورک کار می‌کرد، به عنوان بزرگ‌ترین فروشنده عصر حاضر شناخته شده است.
- او در کتابش «روش فلدمن» توضیح می‌دهد که موفقیت بزرگ وی عمدتاً به خاطر توانایی‌های او در پرسیدن سوالات درست، در زمان درست بوده است. فلدمن متوجه شده بود که پرسش‌هایی مناسب و به‌جا، می‌تواند یک فرد بی‌علاقه یا منفی را گاهی در عرض چند ثانیه به یک مشتری تبدیل کند.
- هینز گلدمن در کتاب خود با نام «مشتری شناسی» می‌گوید: «۹۵ درصد فروشندگان فقط با اصلاح جملات خود به هنگام ارایه مطالب می‌توانند نتایج بهتری به دست آورند. گزینش کلمات صحیح در یک پرسش یا پاسخ می‌تواند تاثیر عمیقی به جای بگذارد و تنها راه اطمینان از عالی بودن ساختار جملات، نوشتن آن‌ها روی کاغذ و تمرین مکرر آن‌هاست.»

## ۲۵ اشتباه فروشندگی

- |   |  |
|---|--|
| ۱۳. ارزش قایل نبودن برای کار خود                | ۱. نپرداختن با تمام وجود به کار          |
| ۱۴. مجبور کردن مشتری                            | ۲. گوش ندادن به حرف‌های مشتری            |
| ۱۵. کودن فرض کردن مشتری                         | ۳. درک نکردن احساس مشتری                 |
| ۱۶. در جریان اطلاعات قرار نداشتن                | ۴. مشتری را دشمن انگاشتن                 |
| ۱۷. شتابزدگی در فروش                            | ۵. حواس‌پرتی                             |
| ۱۸. استفاده نکردن از شاهد زنده                  | ۶. یادداشت نکردن گفته‌ها                 |
| ۱۹. برخورد تملق‌آمیز                            | ۷. قطع ارتباط با مشتری                   |
| ۲۰. فریفته طرح‌های بزرگ شدن                     | ۸. حفظ نکردن ارتباط با مشتری             |
| ۲۱. ناامیدی از جواب رد شنیدن                    | ۹. نداشتن برنامه روزانه                  |
| ۲۲. احساس مسؤلیت نکردن نسبت به کار              | ۱۰. بی‌توجهی به سر و وضع خود             |
| ۲۳. اهمیت ندادن به مشتری                        | ۱۱. منظم نبودن لوزام کار                 |
| ۲۴. منفی‌باف بودن                               | ۱۲. نسنجیدن کالا و خدمات از دیدگاه مشتری |
| ۲۵. روحیه رقابت و خطر کردن را در خود تقویت کنید |  |



در این جلسه، آموزش‌هایی نیز در رابطه با محیط کار Visual Studio و پروسهٔ TFS (Team Foundation Service) توسط آقای حسن حکمی داده شد و در نهایت ایشان سروری را به آدرس [khaste.visualstudio.com](https://khaste.visualstudio.com) برای این پروژه در نظر گرفتند که برای کار با آن باید از Visual Studio 2012 استفاده کنیم و به این ترتیب بستری فراهم شد تا به شیوهٔ کاملاً مهندسی فازهای مختلف پروژه را جلو ببریم و Task‌های خود و دیگر اعضا را در این قسمت قرار داده و گزارشات را مشاهده نماییم.

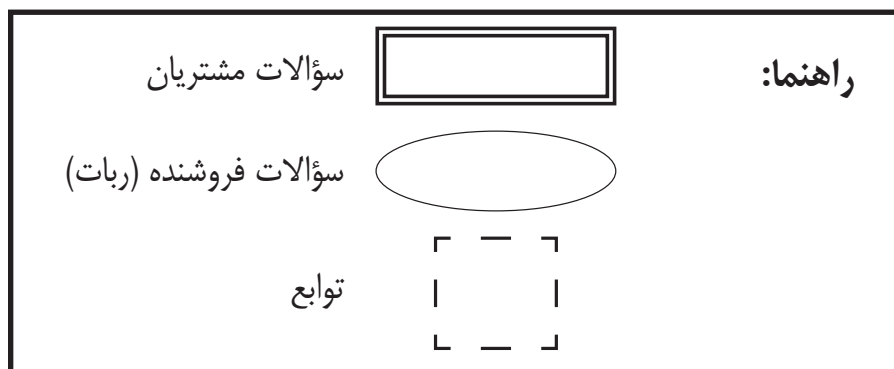
برای استفاده از این سرور ابتدا باید یک Windows Live ID بسازیم و با آن وارد شویم و بعد از آن نیز وارد پروژهٔ مورد نظر یعنی G2S2R شویم. محیط کار به شکل زیر می‌باشد:

The screenshot displays the TFS web interface for the G2S2R project. The main content area shows the 'overview' for 'Iteration 1' (October 21 - November 30) with a Burndown chart and 115 work items. Below this, there are 'TEAM FAVORITES' for Product Backlog, Active Bugs, and Feedback, each with a '0' count. On the right, there is a list of 'MEMBERS (10)' including a.babae66@yahoo.com, hourirazavi@gmail.com, Khaste, MahdiYar, mojaverezaei@gmail.com, mojtaba.heravi@gmail.com, Pegah Khodakarami, rezvankhandani@yahoo.com, sadeq, and حسن اشعائینی.

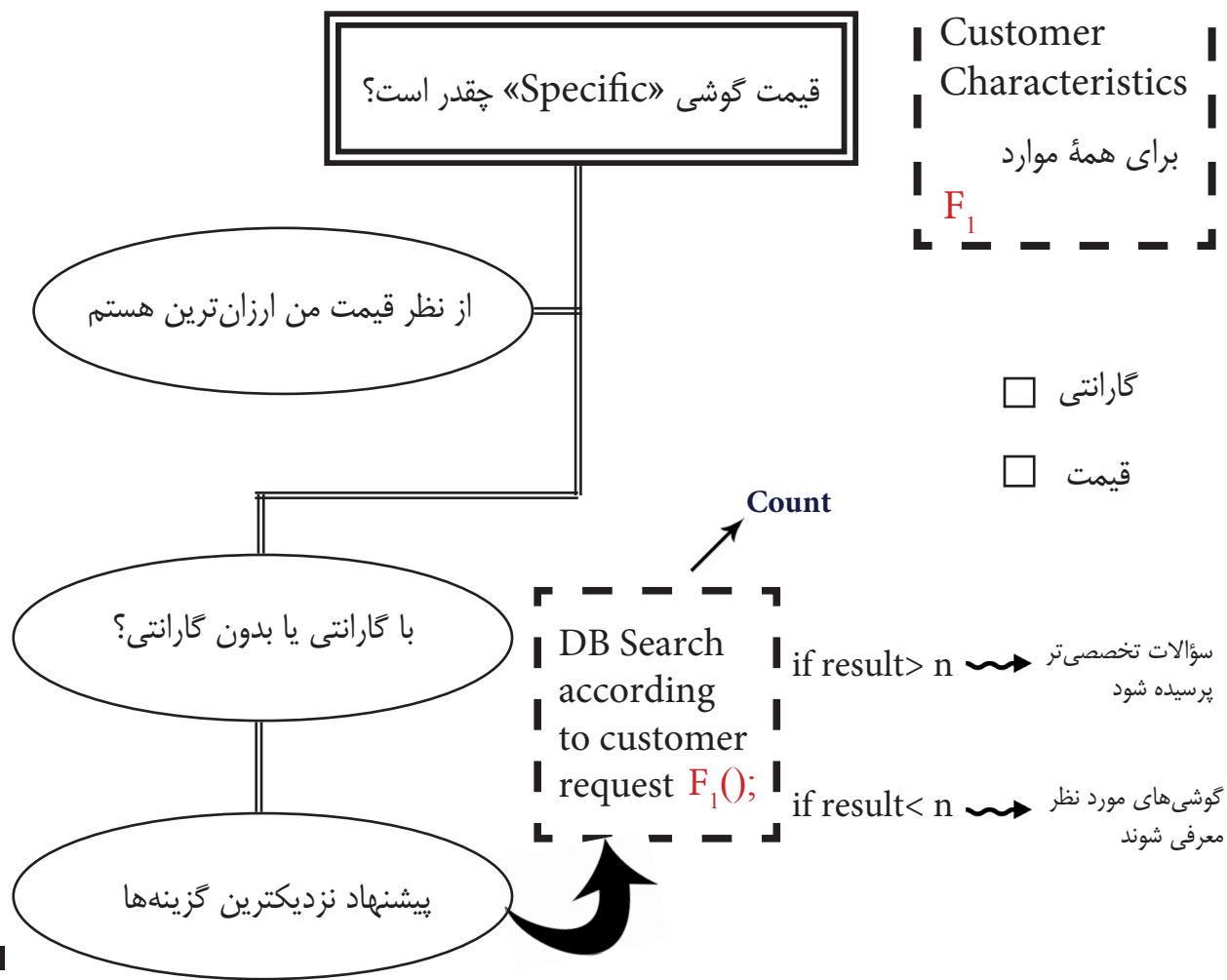
همچنین آقای حکمی سایتی را برای معرفی پروژه، معرفی اعضا، نمایش روند پیشرفت پروژه و همچنین Presentation نهایی طراحی کرد که آدرس آن [www.g2s2r.ir](http://www.g2s2r.ir) می‌باشد. شکل زیر نمایش اولیهٔ این سایت را نشان می‌دهد:

The screenshot shows the website [www.g2s2r.ir](http://www.g2s2r.ir). The main visual is a collection of smartphones displaying various mobile applications. Overlaid on this is the text 'MOBILE PHONE SELLER AGENT' in large white letters, and below it, 'AN INSTANCE OF G2S2R FRAMEWORK'. The website has a dark footer with the G2S2R logo and a navigation menu with links: HOME, ABOUT PROJECT, ABOUT TEAM, G2S2R IN MEDIA, TEAM MEMBERS, RECENT WORKS, GALLERY, CONTACT.

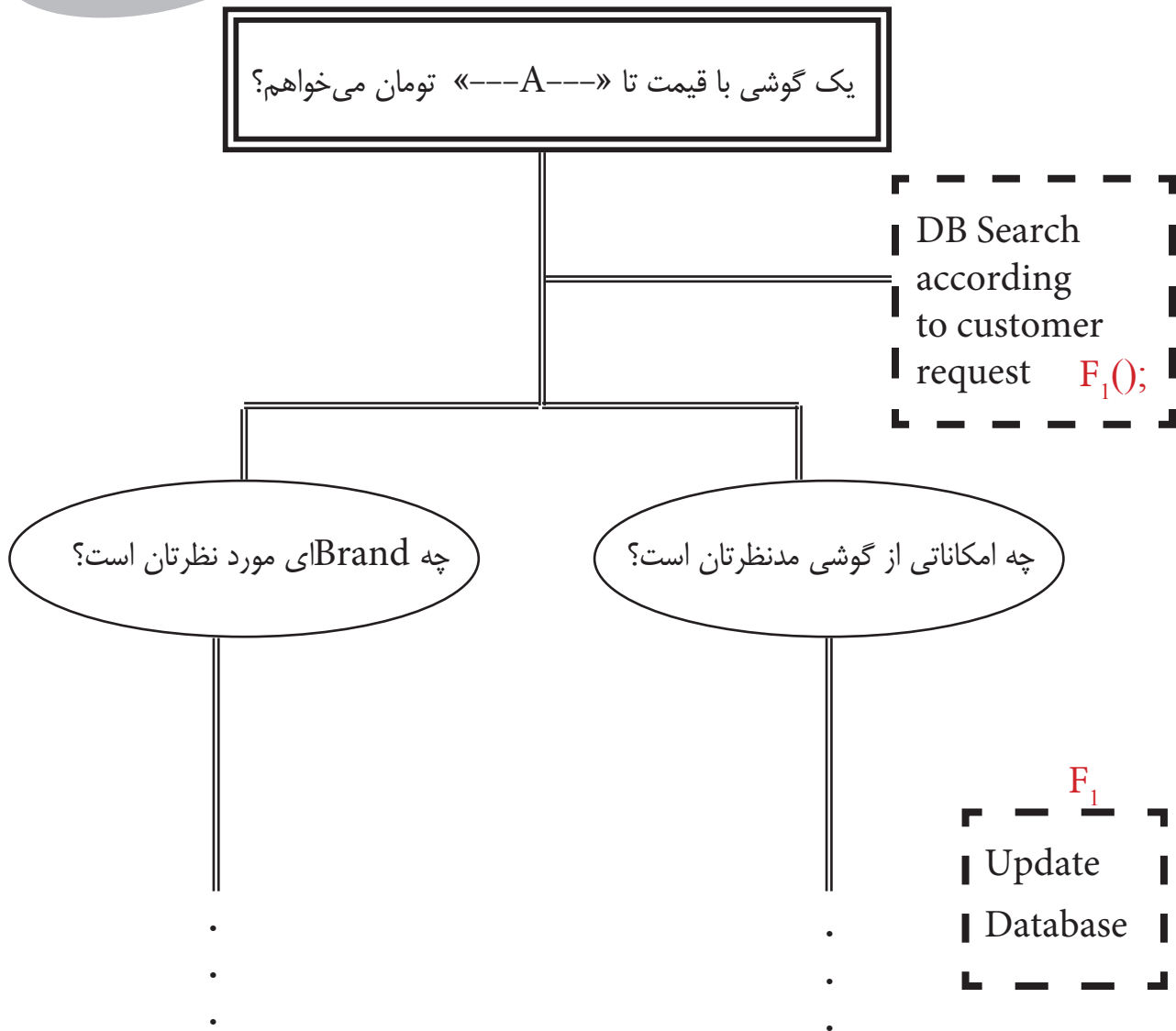
این جلسه در تاریخ ۹۱/۸/۲۲ به عنوان دومین جلسه طوفان مغزی (Brain Storming) تشکیل شد که در آن بر اساس سؤالات جلسه قبل و همچنین به منظور مشخص شدن ساختار سیستم، حالت‌های مختلف را در نظر گرفته و به بررسی درخت حالت آن‌ها پرداختیم:



Case 1



## Case 2



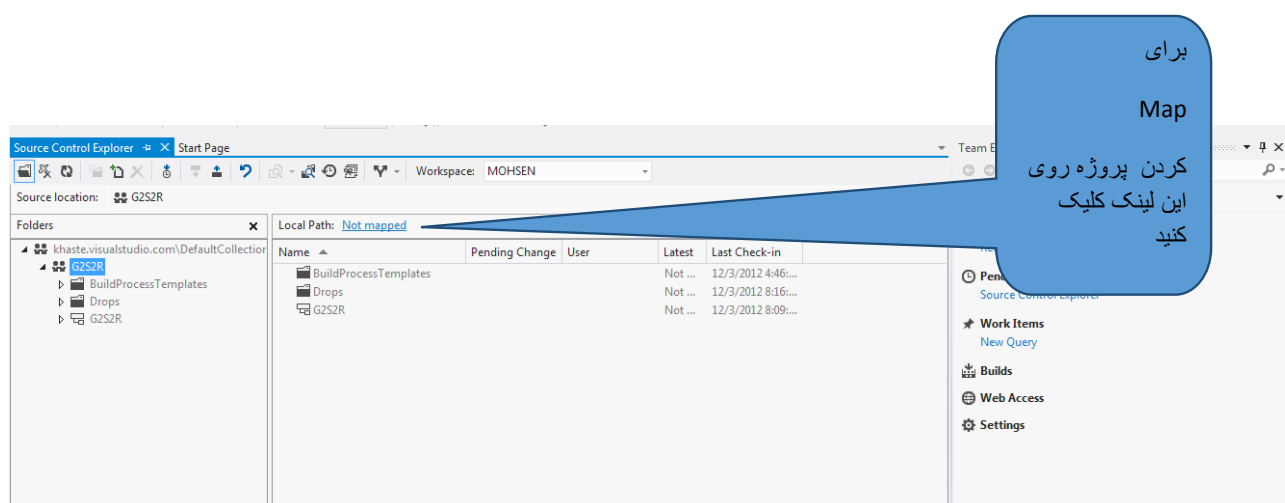
در جلسات آینده حالات دیگر مورد بررسی قرار خواهند گرفت.

در پایان این جلسه، با توجه به این نکته که ما با زبان C# و پایگاه داده SQL SERVER و در محیط Visual Studio 2013 کار خواهیم کرد، قرار شد افراد به تحقیق در مورد نرم‌افزارها و Framework های مورد نیاز و نحوه کار با آنها بپردازند. در این راستا، آقای محسن اسماعیلی فایل را در رابطه با نحوه کار با TFS که روند کاری ما در پروژه می‌باشد، تهیه کردند که در صفحه بعدی آورده شده است.

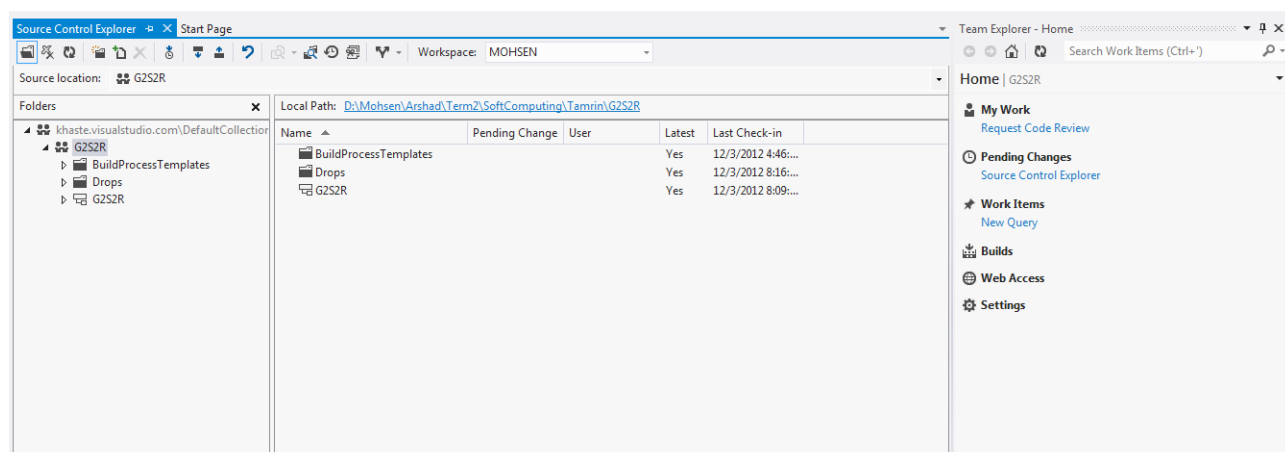
## TFS

در این سند قصد دارم خیلی خلاصه مفاهیمی که برای کار با TFS برای پیش بردن پروژه به صورت عمومی دوستان لازم دارند را ارائه دهم، امیدوارم مفید واقع شود مابقی اگر جایی لازم به توضیح بیشتر بود در حد توان در خدمت دوستان هستم.

پس از اینکه در Source Control از طریق Visual Studio وارد شدید لازم است با کلیک بر روی گزینه Source Control در این قسمت ابتدا مسیری را روی سیستم خود برای پروژه Map کنید تا فایل‌هایی که از روی سرور بر روی System شما Get می‌شود در آنجا ذخیره شود. با کلیک روی گزینه Source Control در صفحه باز شده در صورتی که پروژه در سیستم شما Map نباشد لینک Not Map را کلیک کرده و مسیری را برای آن انتخاب می‌کنیم و اگر قبلاً روی سیستم شما Map شده باشد، لینک Not Map به لینک مسیر Map شده تبدیل می‌شود. اگر برای اولین بار پروژه را Map کنیم پیغامی مبنی بر Get Latest Version نمایش داده می‌شود و با پذیرفتن آن پروژه از سرور روی سیستم شما Get می‌شود.

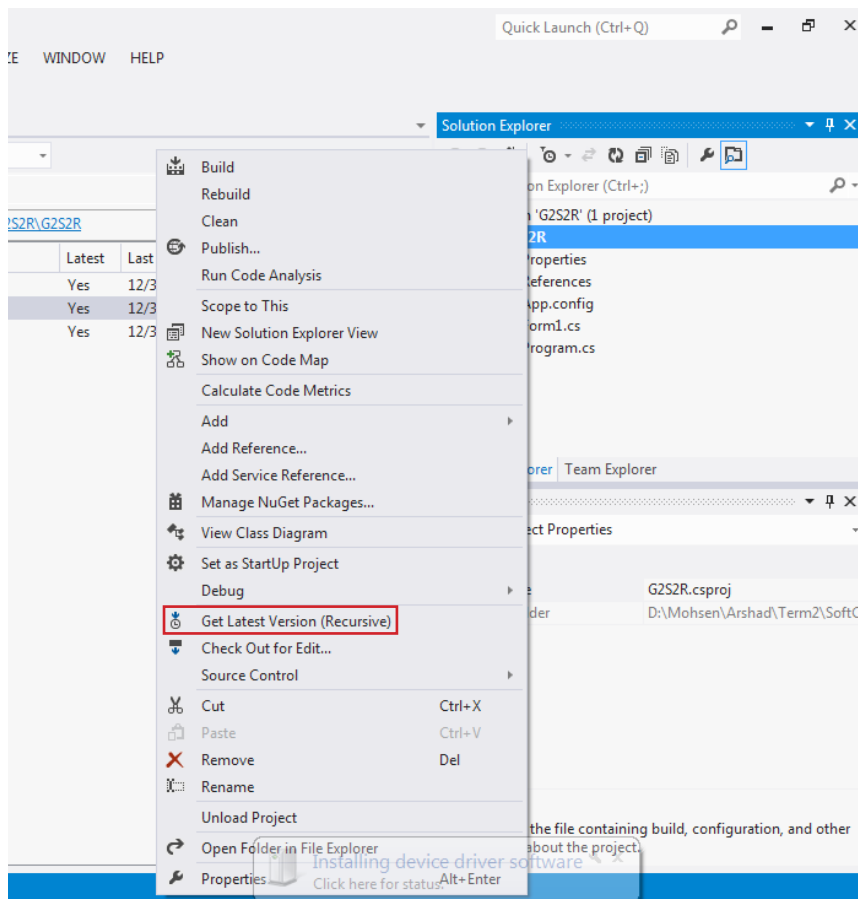


در صورتی که پروژه را روی کامپیوتر خود Map کرده باشیم:



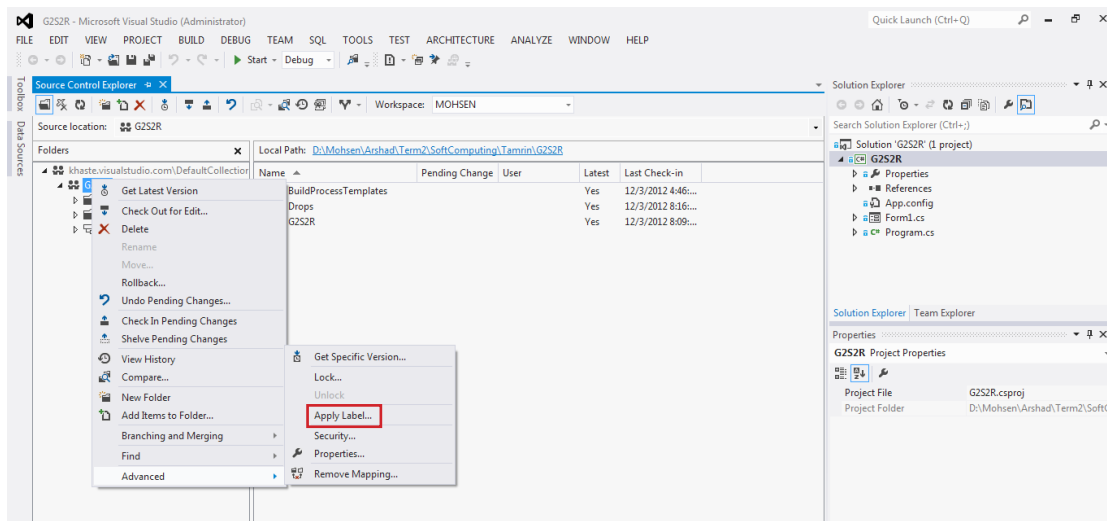
### Get Latest Version:

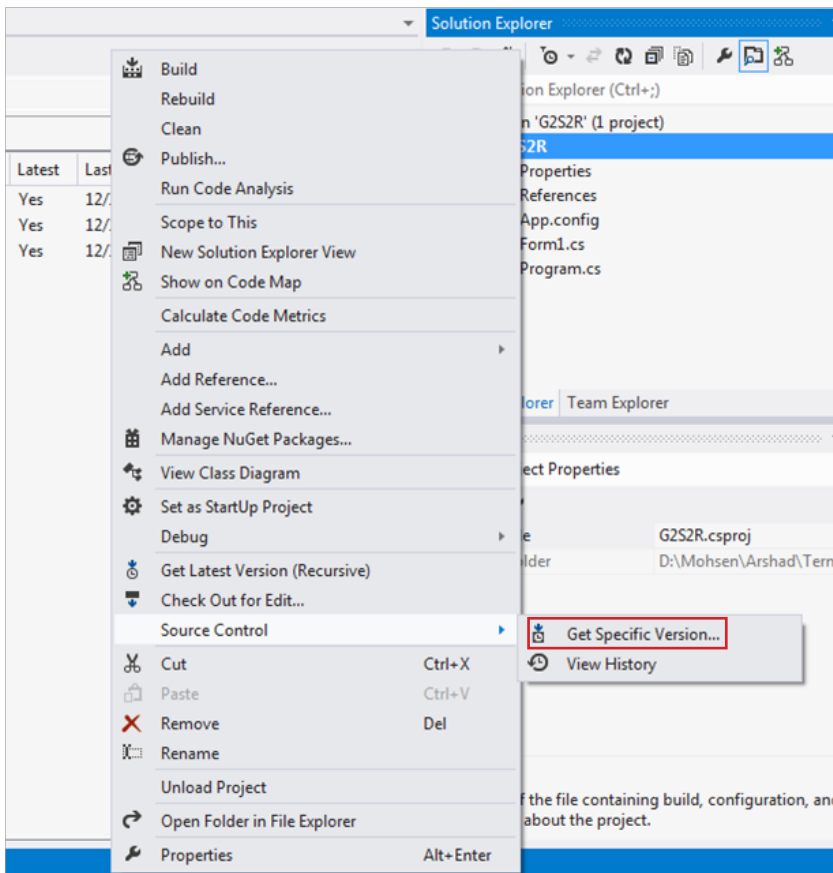
از این فرایند برای انتقال آخرین ورژن از پروژه از روی سرور بر روی کامپیوتر کاربر استفاده می‌شود که می‌تواند برای گرفتن یک یا چند فایل یا کل پروژه استفاده شود.



## Apply Label:

گاهی در کدنویسی لازم است قبل از شروع به تغییر یک قسمت یک برچسب یا نشانه بر روی پروژه تنظیم کنیم و اگر بعد از تغییر کد مشکل خاصی پیش آمد بتوانیم پروژه را دقیقاً به حالتی برگردانیم که روی آن برچسب گذاری شده است، توصیه می‌شود قبل از شروع به تغییر یک برچسب برای کد ایجاد شود تا در صورت بروز مشکلات احتمالی امکان بازگشت باشد. برای این کار روی Source Control از پنجره Team Explorer کلیک کرده و روی نام پروژه ای از TFS که می‌خواهیم روی آن برچسب تعیین کنیم کلیک راست کرده و Apply Model را انتخاب می‌کنیم و با تعیین یک نام برای آن برچسب را ایجاد می‌کنیم (ترجیحاً در انتخاب این نام از نام‌های بی‌ربط استفاده نشود).



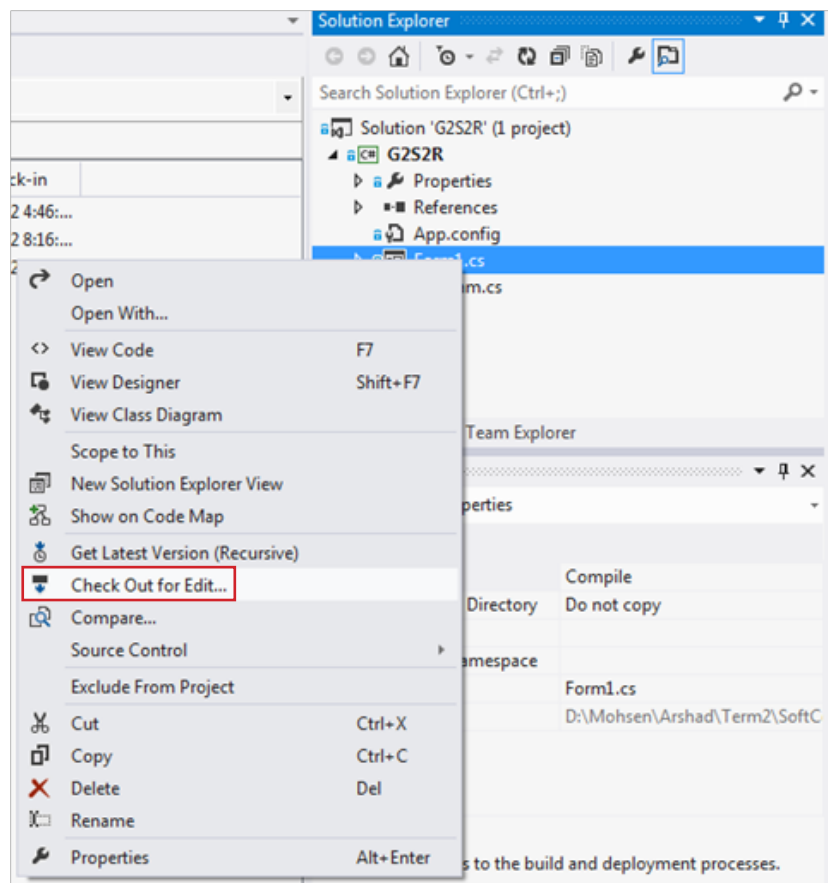


### Get Specific Version:

برای انتقال ورژن خاصی از پروژه روی سیستم از آن استفاده می شود یکی از کاربردهای مهم این آیتم برای مواقعی است که قصد داریم ورژن خاصی از پروژه را بر اساس برجسب گذاری هایی که قبلاً انجام داده ایم برگردانیم همچنین می توانیم پروژه را به تاریخ Check in خاصی برگردانیم.

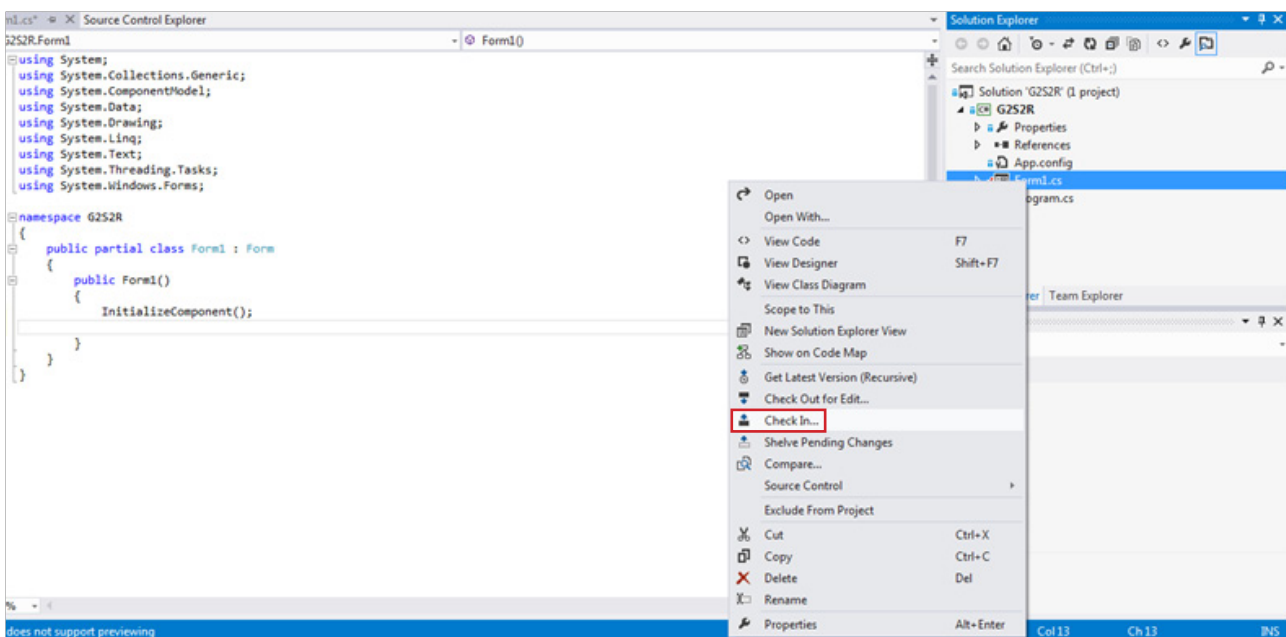
### Check Out For Edit:

این فرایند برای قفل کردن یک فایل برای کاربر استفاده می شود چون حالت کدنویسی مطمئن در تیم هایی که از اینگونه Source Control ها استفاده می کنند این است که در آن واحد یک فایل را چند نفر نتوانند تغییر دهند. شایان ذکر است که وقتی کاربر یک فایل را باز می کند و شروع به تغییر آن می کند TFS خود به خود فایل را برای آن کاربر Check out می کند و اگر فایل در دست کاربر دیگری باشد از Check Out شدن فایل جلوگیری می کند.

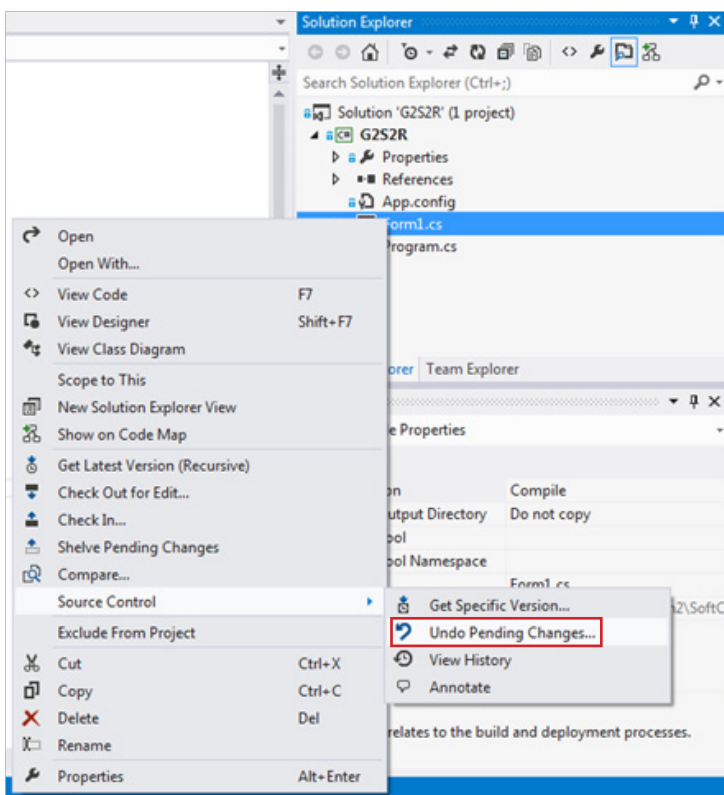


## Check In:

در این فرایند که می توانیم با راست کلیک و انتخاب گزینه Check in آن را انجام دهیم تغییرات پروژه از کامپیوتر به سرور انتقال پیدا می کند که اعضای دیگر تیم با فرایند دیگری به نام Get Latest Version می توانند تغییرات Check in شده ما را روی کامپیوتر خود مشاهده کنند، در این فرایند بخشی را برای Comment تنظیم کردن برای Check in مورد نظر قرار داده اند که پیشنهاد می شود حتی الامکان(همیشه این فیلد وارد شود چون در صورت وارد شدن خلی در پروژه به واسطه این Check in این فیلد کمک بسیاری در برگرداندن پروژه به حالت قبل از Check in بازی می کند). باید توجه داشت که در این فرایند فقط تغییرات انجام شده به سرور انتقال می یابد.



۱۵

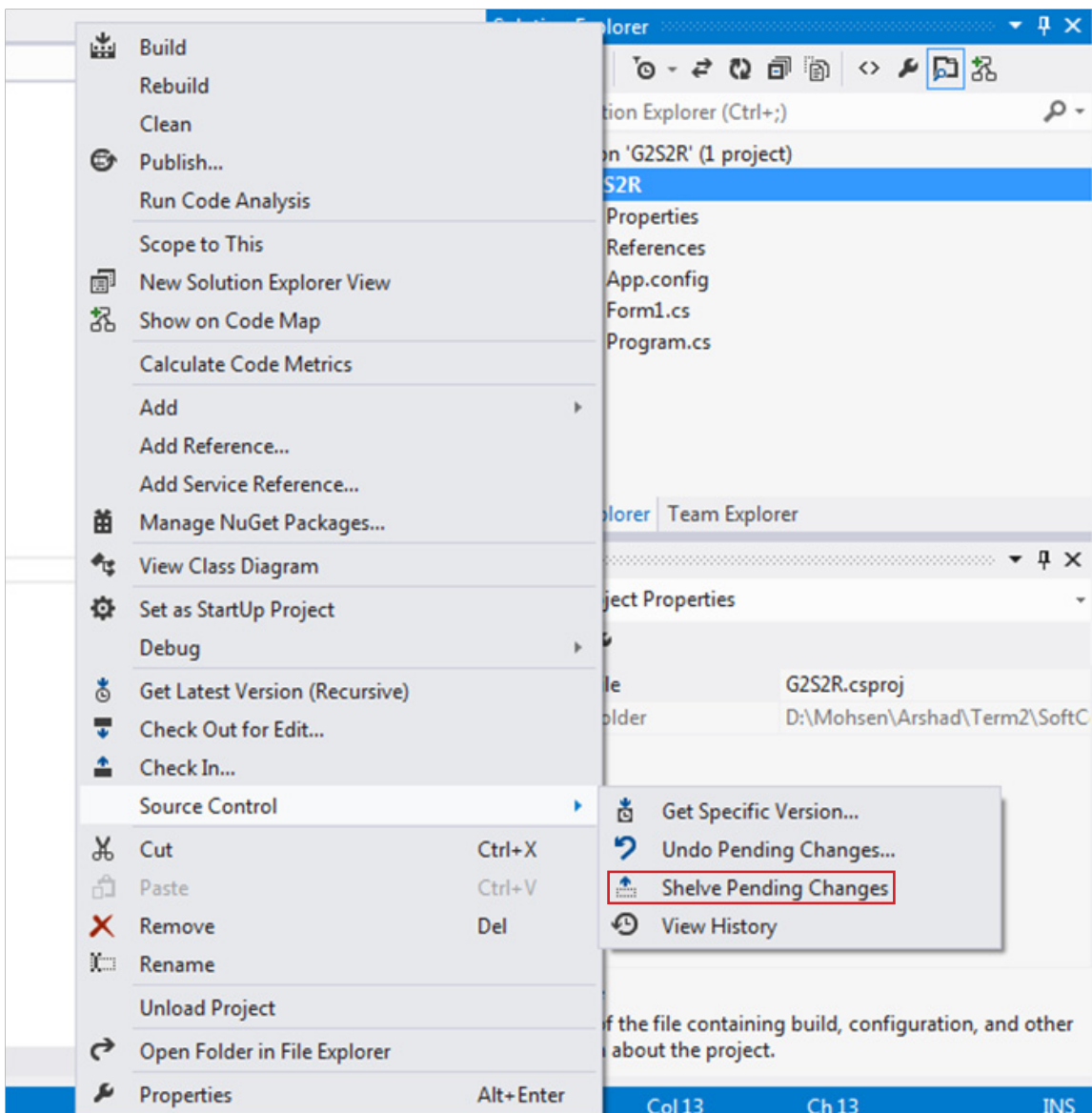


## Undo Pending Changes:

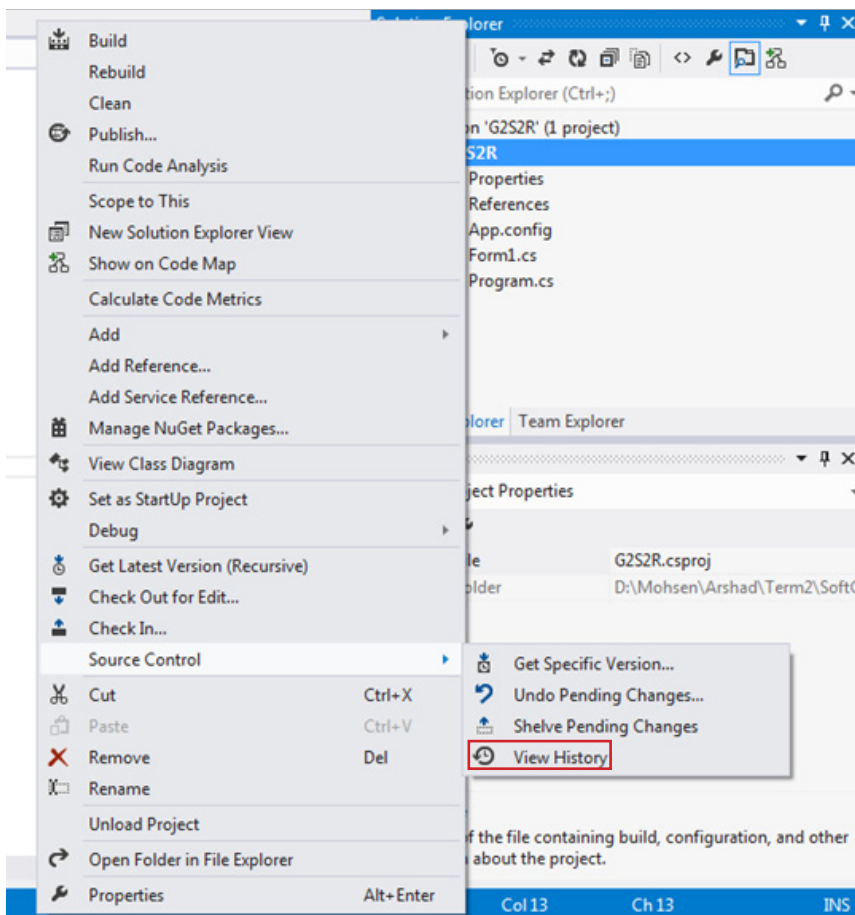
برای برگرداندن تغییرات انجام شده به حالت سرور یا به عبارتی لغو تمام تغییرات از این گزینه می توان استفاده کرد.

## Shelve:

باید دقت داشت که هیچ وقت پروژه ای را که خطا دارد Check in نباید کرد چون در اینصورت کدی که روی سرور قرار می گیرد با خطا مواجه شده و با Get Latest Version که توسط کاربران دیگر انجام می شود کلیه کاربران هم با خطا مواجه می شوند یا زمانی که کاری که قصد انجام آن را داریم کامل نشده است و قصد Check in آن را نداریم و از طرفی برای بالا رفتن امنیت کد قصد داریم آن را روی سرور انتقال دهیم در اینجور مواقع از گزینه Shelve استفاده می شود که با کلیک راست روی پروژه یا هر فایل دیگری از پروژه می توانیم این کار را انجام دهیم در اینصورت تغییرات ما روی سرور قرار می گیرد با این تفاوت که با Get Latest Version کردن کاربران این تغییرات برای آنها انتقال پیدا نخواهد کرد و اگر مشکل خاصی برای کد شما پیش آید با Un Shelve می توانید قسمت Shelve شده را روی سیستم خود مشاهده کنید توصیه می شود هر جایی نیاز دیدید حتماً از این مورد استفاده کنید. برای انجام Shelve می توانیم روی پروژه کلیک راست کرده و گزینه Shelve را انتخاب کنیم یا از منوی Shelve => Source Control => File استفاده کنیم و برای انجام Un Shelve که برای مواقعی بکار می رود که نسخه ای که روی سیستم موجود است مشکل داشته باشد و نیاز داشته باشیم آن نسخه ای که Shelve کردیم را برگردانیم برای این منظور از منوی Shelve => Source Control => Find Shelve استفاده می کنیم و پس از یافتن Shelve مورد نظر روی آن کلیک راست کرده و Un Shelve را کلیک می کنیم





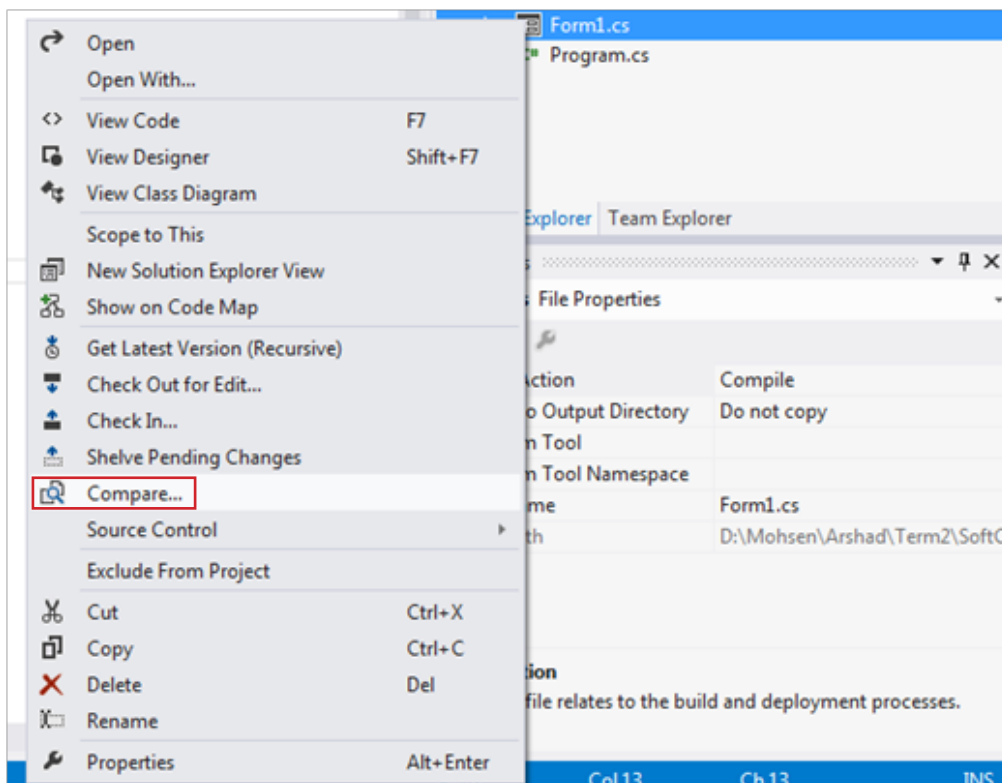


## History:

کلیه اطلاعات تغییرات روی پروژه و کاربرانی که این تغییرات را اعمال کرده اند در View History روی تک تک فایل های پروژه موجود می باشد که در صورت نیاز به آن ها می توانیم آن ورژن را روی سیستم خود برگردانیم.

## Compare:

در صورتی که یک فایل را تغییر داده بودیم می توانیم با این گزینه تغییرات را با ورژنی که روی سرور است یا ورژن خاصی مقایسه کرده و جاهایی که تغییرات دادیم را ملاحظه نماییم.



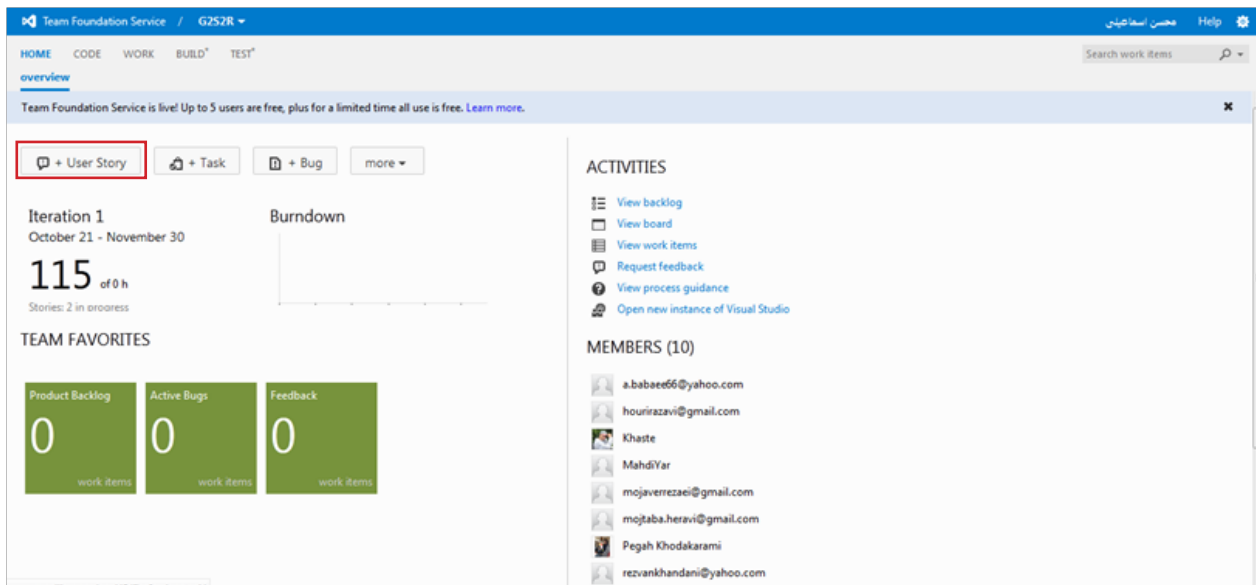
## Work Items Flow:

از Work Item ها برای دنبال کردن، مانیتور کردن و گزارش بر روی توسعه محصول و آینده آن استفاده می شود. یک Work Item در دیتابیس TFS بر روی آن تعریف می شود ذخیره می شود، در متودولوژی MSF انواع Work Item هایی که استفاده می شوند عبارتند از:

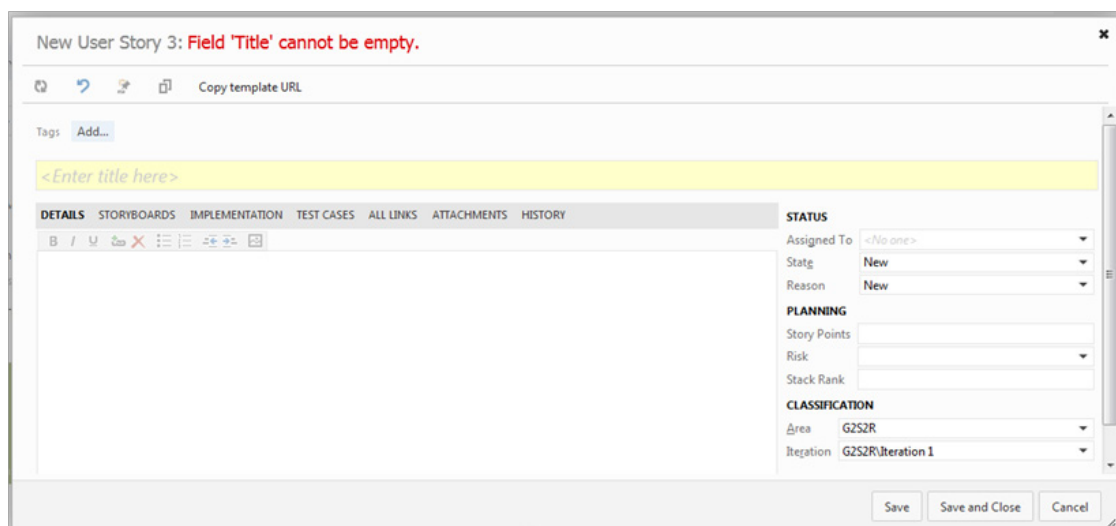
User Stories, Task, Test Case, Bug, Issue

## User Story:

تیم برنامه نویسی از User Story برای تعریف امکانات، توابع و نیازمندی ها برای پیاده سازی استفاده می کنند. یک User Story اهداف مشتری را در بالاترین سطح توصیف می کند .  
برای استفاده از این امکان TFS در وب سایت مربوطه پس از وارد شدن در سایت می توانیم با توجه به تصویر زیر مراحل را طی کرده و User Story مورد نظر را وارد کنیم.



پس از کلیک User Story از پنجره زیر اطلاعات لازم را برای مورد مورد نظر وارد می کنیم.



خانم حوری رضوی نیز در زمینه متدولوژی MSF for Agile Software development تحقیقاتی انجام دادند که نتایج آن به صورت زیر می باشد:

## MSF

MSF ابزاری است که از یک استراتژی مهندسی نرم افزار پویا حمایت می کند و از فرایندهای تکرار چندگانه و روش های مبتنی بر سناریو برای پیاده سازی پروژه ها به صورت تیمی بهره می برد.

### Key Items

ابتدا یک پروژه تیمی با استفاده از MSF در محیط visual studio تعریف می شود. پیکره اصلی این فرایند شامل آیتم های کلیدی زیر می شود که هر کدام را به تفصیل در ادامه توضیح می دهیم.

\* Work item آیتم های کاری

\* Groups and permissions گروه ها و اختیارات آنها

\* Source control کنترل منابع

\* Areas and iterations نواحی و تکرارها

\* Reports گزارش ها

### Work Items

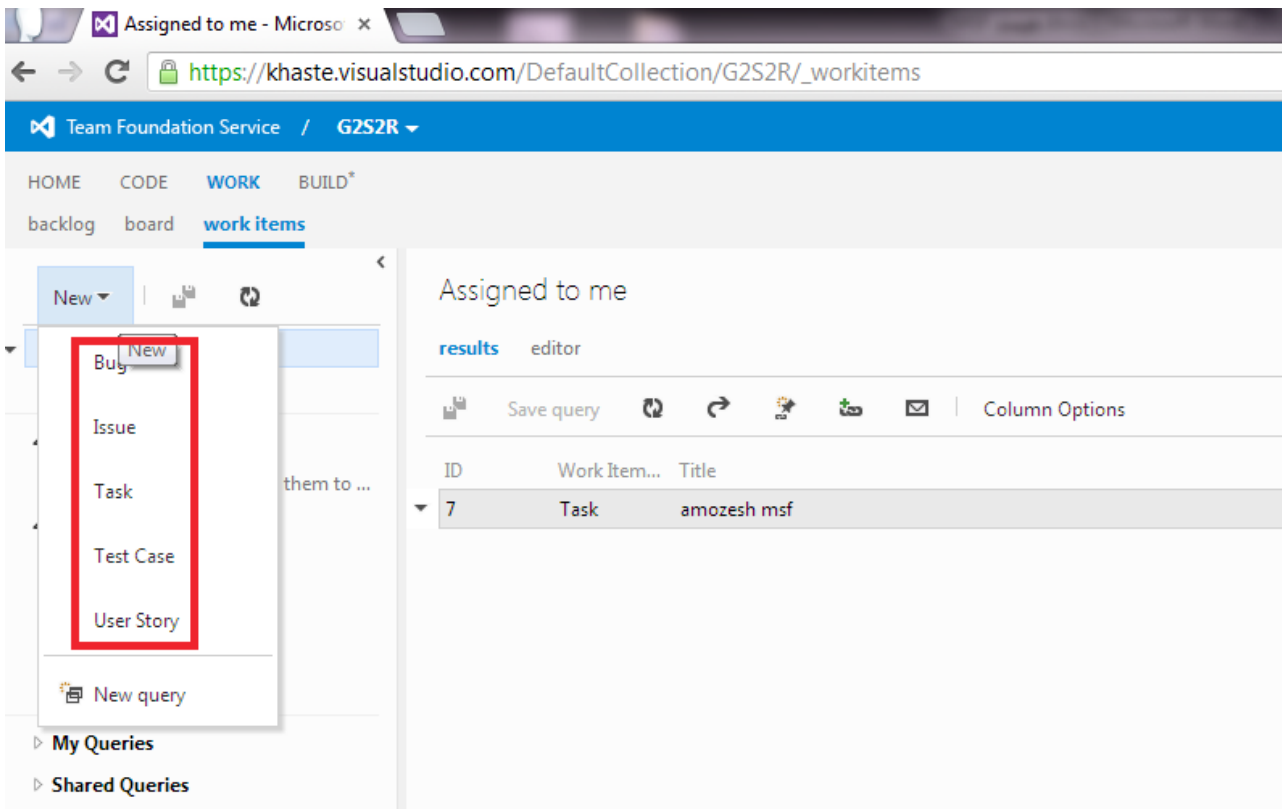
MSF شامل آیتم های کاری زیر می شود:

\* Bug بیانگر یک مشکل یا یک مشکل بالقوه در پروژه است. ایجاد باگ این امکان را به ما می دهد که نقص را به دقت برای اعضای تیم گزارش کنیم تا بتوانند از تاثیرات آن آگاه شوند. در این قسمت باید مراحل که منجر به بروز مشکل شده کاملاً توضیح داده شود.

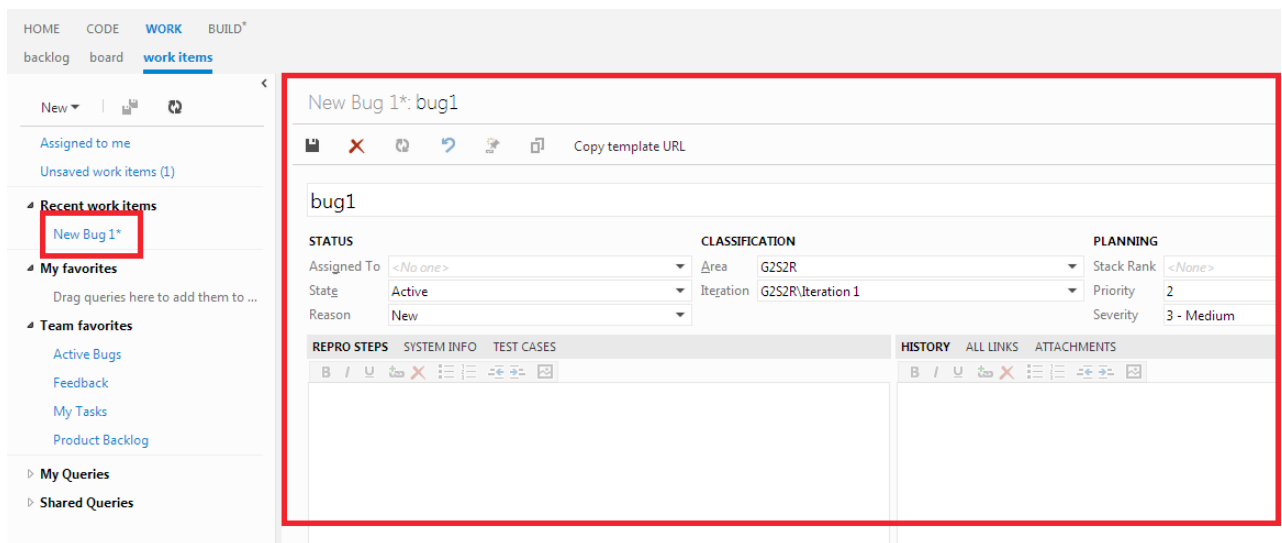
\* Issue بیانگر موانع و ریسک هایی است که امکان وقوع آنها وجود دارد و در صورت بروز ممکن است تاثیرات منفی بر پروژه بگذارد. افراد تیم نباید از بیان ریسک ها و موانع بترسند زیرا در این صورت است که می توانند پروژه را با سرعت و کیفیت بهتر پیش ببرند.

\* User Story بیانگر ویژگی ها، توابع و نیازها برای پیاده سازی پروژه است. یک user story اهداف مشتری را به طور مفهومی مشخص می کند و در نتیجه به تیم در تعریف، اولویت بندی و برنامه ریزی کارها کمک می کند. Task آیتم یا کاری خاص است که به یک عضو تیم برای انجام واگذار می شود. و معمولاً زمان آن یک تا دو روز است.

\* Test Case آزمایش هایی برای تست user story ها هستند.



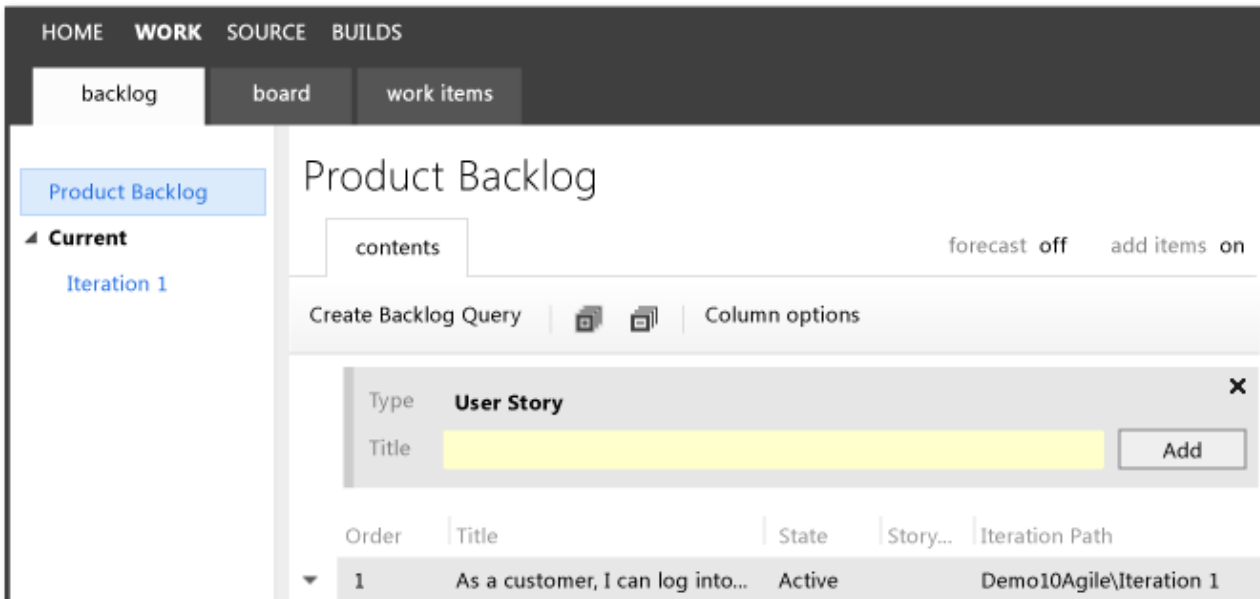
همان‌طور که در تصویر بالا مشاهده می‌کنید برای ایجاد یک work item وارد تب work items می‌شویم و سپس تب new را انتخاب می‌کنیم و پس از آن یکی از موارد مورد نظر را انتخاب می‌کنیم. مثلاً اگر باگ را انتخاب کنیم پنجره زیر ظاهر خواهد شد.



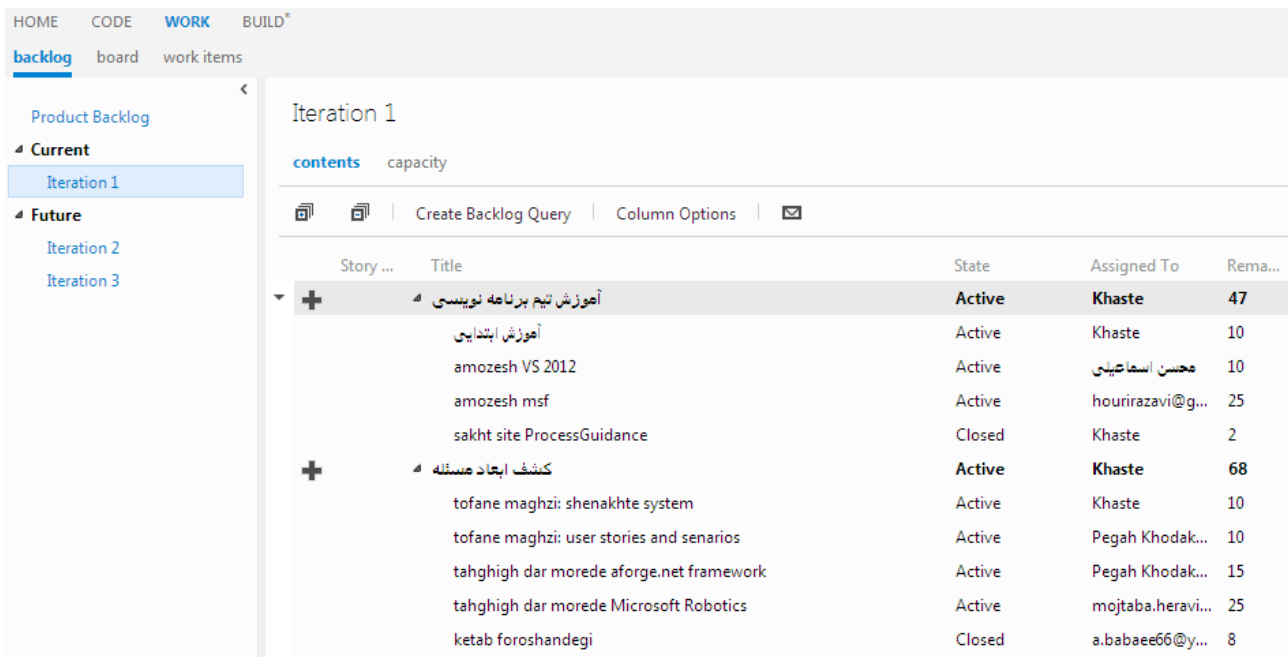
## ایجاد یک back log

در توسعه نرم‌افزار پویا، لیست نیازهای یک محصول را که قصد ساخت یا گسترش آن را داریم در قالب یک back log نشان می‌دهیم. به طور ساده‌تر، back log پاسخی به این سوالات است «چه می‌خواهیم بسازیم؟ و چگونه می‌خواهیم بسازیم؟». پاسخ این سوالات در قالب work item ذخیره می‌شوند. نمونه یک back log در شکل صفحه بعد آمده است. بعد از ساخت back log زمان طراحی iteration یا sprintهاست. در اینجا می‌توانیم آیتم‌ها را از صفحه back log به sprint منتقل کنیم. همچنین از طریق صفحه sprint back log می‌توانیم taskهایی برای پیاده‌سازی یک back log item اضافه کنیم. همچنین افراد تیم می‌توانند مراحل

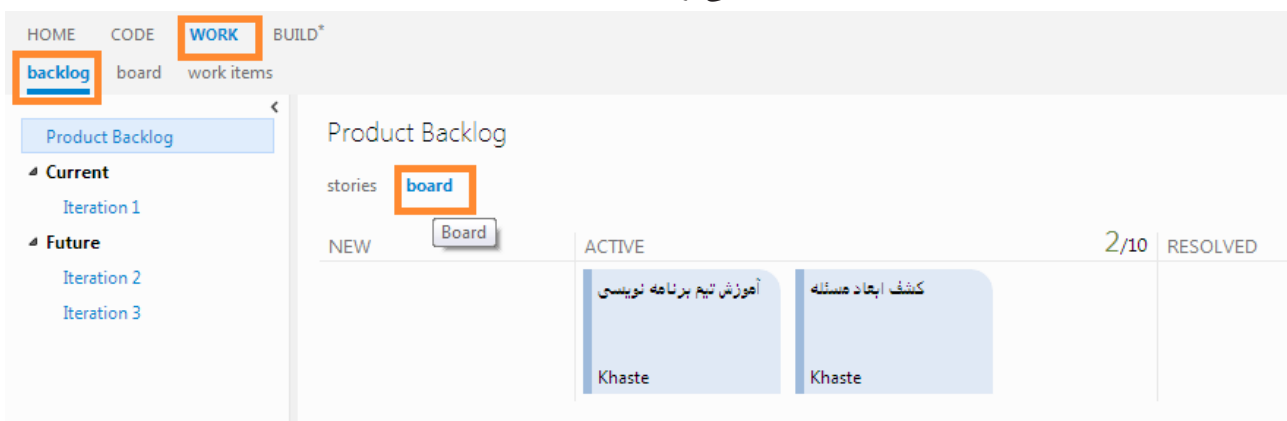
پیشرفت پروژه را در task board ببینند.



مثالی از sprint black log



مثالی از task board



## تفاوت‌های Product Back log و Iteration or Sprint back log

Product Back log مربوط به آیتم‌هایی می‌شود که فعال هستند و هنوز کامل و بسته نشده‌اند. در مقابل Iteration back log به آیتم‌ها و وظایفی مربوط می‌شود که تیم قرار است آنها را در یک iteration خاص کامل کند. در Product Back log تیم user storyها را با یک واحد نسبی تخمین می‌زند ولی در Iteration back log تیم این وظایف را با تخصیص ساعت مشخص می‌کند.

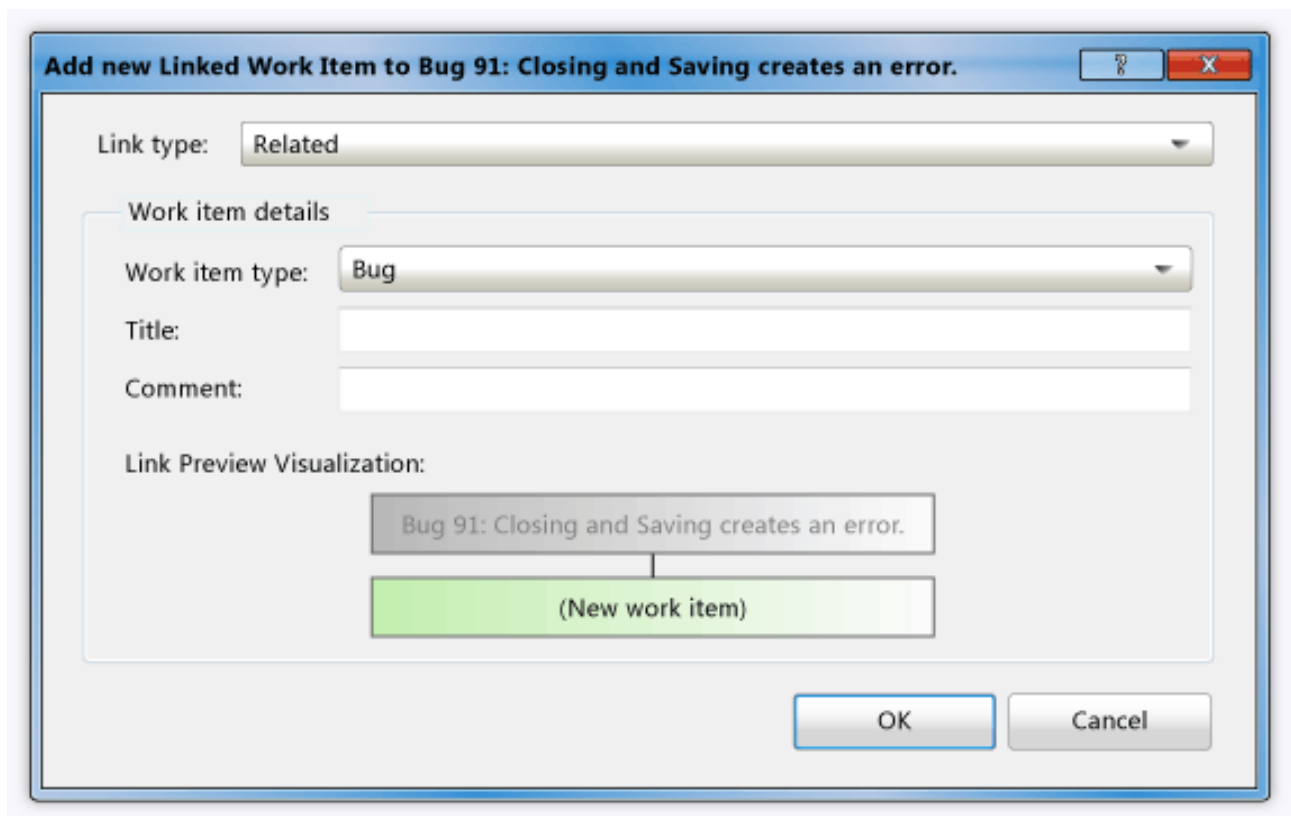
### اتصال یک آیتم کاری به آیتم مرتبط به آن

ما می‌توانیم یک آیتم کاری را به user story موجود یا آیتم‌های کاری دیگر مرتبط کنیم. این کار را می‌توان از طریق یک آیتم کاری باز یا از لیست نتایج یک work item query انجام داد.

۱- ابتدا پروژه خود را از طریق team explorer انتخاب کنید.

۲- روی Work Items کلیک راست کرده و open را بزنید.

۳- در team explorer روی آیتم کاری جاری که می‌خواهید آن را با دیگری لینک کنید کلیک راست کرده و Add New Linked Work Items را انتخاب کنید. پنجره مربوطه به شکل زیر است.



۴. موارد زیر را در پنجره بالا به این صورت تنظیم کنید:

- در قسمت link type اگر می‌خواهید یک آیتم کاری جدید به user story یا test case جاری اضافه کنید، tests را انتخاب کنید ولی اگر می‌خواهید یک آیتم کاری جدید به task, issue, یا باگ دیگری بیفزایید گزینه related را انتخاب کنید.
- در قسمت work item type نوع آیتم مورد نظر خود را مشخص کنید.

• در title نامی را که توصیف‌کننده آیتم مورد نظر باشد بنویسید.

• در comment اطلاعات مورد نیاز را وارد کنید.

۵. سپس ok را کلیک کنید تا یک آیتم کاری با مشخصات مورد نظر شما باز شود.

۶. روی toolbar آیتم کاری، Save Work Item را کلیک کنید.

## نکاتی در مورد bug:

۱. زمانی که با رفتار غیرعادی نرم‌افزار در حین تست مواجه می‌شوید، بهتر است سریعاً یک باگ بسازید.

۲. با استفاده از test runner شما می‌توانید داده‌های مناسب برای trace را جمع‌آوری کرده به طوری که توسعه‌دهنده سیستم بتواند در آینده برای حل خطاها از آن استفاده کند.

۳. شما می‌توانید مراحل انجام تست دستی را به صورت متنی در log file ذخیره کنید و سپس آن را به باگ اضافه کنید.

۴. شما می‌توانید با استفاده از action log از یک باگ یا نتیجه تست، یک test case دستی (manual) بسازید.

۵. در حالتی که باگ بر اساس test case ساخته می‌شود به راحتی می‌توان state آن را به‌روز کرد.

---

زمانی که یک پروژه تیمی با استفاده از MSF ایجاد می‌شود، آیتم‌های کاری زیر برای ما ساخته می‌شود. در واقع این لیستی از وظایفی است که باید برای شروع کار انجام شوند:

\* Create Vision Statement : هدف نهایی و نتایج مورد انتظار مشخص می‌شوند.

\* Create Persons : افرادی که قرار است به عنوان کاربر با این سیستم در ارتباط باشند مشخص می‌شوند.

\* Set Permissions : در اینجا هر عضو تیم را در یکی از گروه‌های administrators, build services, contributors, readers قرار می‌دهیم.

\* Migration of Source Code and work items : اگر قصد استفاده از TSF را داریم باید انتقال کدها و آیتم‌های کاری را قبل از دسترسی اعضا به درستی انجام دهیم.

\* Set Check-in Policies : در اینجا قوانین و ضوابط مربوط به Check-in (تغییرات) را مشخص می‌کنیم.

\* Configure Build : در این قسمت Source Tree اولیه را می‌سازیم و Build را به صورت دوره‌ای (معمولاً روزانه) اجرا می‌کنیم.

\* Send Mail to Users : ایمیلی برای اعضا فرستاده می‌شود تا به آنها بگوید که با چه TSF‌ای وصل شوند و با چه پروژه تیمی شروع به کار کنند.

\* Define Iteration Length : چرخه تکرار پروژه مشخص می‌شود. تعداد تکرارها و زمان هر تکرار را تعیین می‌کنیم. زمان iteration به اندازه و پیچیدگی پروژه بستگی دارد.

\* Create Test Approach Worksheet including Test Thresholds : هدف از این کار مشخص کردن استراتژی تست از اولین گام‌های پروژه است. این کار به برنامه‌ریزی test task و کارایی بیشتر کمک خواهد کرد.

\* Brainstorm and Prioritize Scenarios List : در اینجا سناریوها و اولویت آنها را مشخص می‌کنیم.

\* Prioritize Quality of Service Requirements List: در اینجا نیازهای غیرعملیاتی مثل امنیت، کارآمدی و ... را مشخص می‌کنیم.

## Groups and permissions

گروه‌های زیر به طور پیش فرض در MSF وجود دارند:

- \* Readers اعضای هستند که فقط امکان خواندن پروژه را دارند و اجازه اعمال هیچگونه تغییر یا کار دیگری را ندارند. (read-only)
- \* Contributors اعضای هستند که امکان اضافه کردن، تغییر دادن و حذف آیتم‌های پروژه تیمی برای آنها وجود دارد. (read, change, lable check-in, check-out, start build)
- \* Build Services این افراد مسئول انجام build service پروژه هستند. (change,check-in, lable, start build, edit)
- \* Project Administrators امکان انجام هر کاری در پروژه را دارند. (build, read)

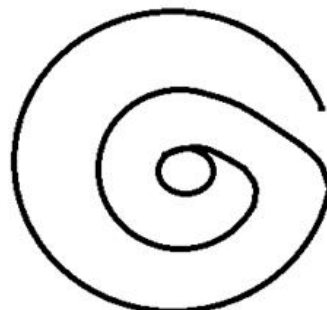
## Source Control

MSF به طور پیش فرض از تنظیمات کنترل منابع زیر استفاده می‌کند:

- \* Multiple Checkout به طور پیش فرض این امکان وجود دارد که چند عضو تیم بتوانند به طور همزمان روی یک فایل در آن واحد کار کنند.
- \* Permissions اختیارات به همان شکلی است که در صفحه قبل آمده است.

## Areas and Iterations

در MSF نواحی و Iterationها بصورت اجزا جدا از هم نیستند. در اینجا معمولاً از روش‌های چرخه‌ای که بر اساس بهبود ویژگی‌ها و اجزاء پیش می‌روند، استفاده می‌شود.



## Reports

گزارش‌هایی که به طور پیش فرض در MSF وجود دارد عبارتند از:

- \* Bugs by priority: این گزارش باگ‌ها را به ترتیب اهمیتشان از مهم‌ترین به کم اهمیت‌ترین مرتب می‌کند.



\* Bugs Rates: میزان تأثیر تشخیص به موقع باگ‌ها، و برطرف کردن آنها را مشخص می‌کند.

\* Builds: این گزارش کیفیت و سایر ویژگی‌های یک Build را مشخص می‌کند.

\* Project velocity: سرعت تیم در انجام کارها را نشان می‌دهد.

\* Quality Indicators: این گزارش کیفیت نرم‌افزار را با جمع‌آوری نتایج تست‌ها، باگ‌ها، کدها و غیره بیان می‌کند.

\* Load Test Summary: نتایج آزمایشات را گزارش می‌دهد.

\* Regression: لیست تست‌هایی را نشان می‌دهد که در گذشته با موفقیت انجام شدند ولی حالا با شکست مواجه شده‌اند.

\* Reactiveness: این گزارش work item‌هایی را که زودتر از موعد مقرر حل شده‌اند یا بسته شده‌اند، لیست می‌کند.

\* Related Work Items: در اینجا work item‌هایی که به هم وابسته هستند مشخص می‌شوند.

\* Remaining work: این گزارش کارهای انجام شده و زمان اتمام آنها را نشان می‌دهد.

\* Unplanned Works: این گزارش حجم کارهای برنامه‌ریزی نشده را مشخص می‌کند.

\* Work Items: آیتم‌های کاری فعال را نشان می‌دهد.

\* Work Item by Owner: میزان کارهای اختصاص داده شده به هر عضو تیم را نشان می‌دهد.

\* Work Item: حالت هر آیتم کاری که فعال یا حل شده یا تمام شده است را نشان می‌دهد.

در قسمت بعدی تحقیق، خانم پگاه خداکرمی مطالبی را در مورد AForge.Net FrameWrok آماده کردند که به شرح زیر می‌باشد:

## Framework های برنامه نویسی محاسبات نرم و هوش مصنوعی

در این گزارش، جهت استفاده از محاسبات نرم در پروژه‌های برنامه نویسی، از جمله پروژه‌های مربوط به درس محاسبات نرم (فروشنده‌ی هوشمند)، به بررسی Framework های موجود می‌پردازم. در این نوشته، عناوین Framework های مرتبط با محاسبات نرم را ذکر کرده و سپس توضیحاتی درباره‌ی هر Framework ارائه می‌دهم. و در آخر، برخی از نظرات برنامه‌نویسان از سایت‌های مختلف را اضافه کردم که با مقایسه‌ی آنها تصمیم‌گیری درباره‌ی انتخاب Framework مورد استفاده در پروژه‌ی، ساده‌تر شود.

### انواع Framework های برنامه نویسی

در حال حاضر Framework های متعددی در ارتباط با هوش مصنوعی در زبان‌های مختلف وجود دارد که طی بررسی‌های انجام شده، سه Framework اصلی در این زمینه عبارتند از:

- **MATLAB**
- **AForge.NET**
- **OpenCV(Open Source Computer Vision Library)**

• **MATLAB**: برنامه متلب طیف وسیعی از توابع برای استفاده در محاسبات نرم را دارا می‌باشد. از آن جمله می‌توان کاربردهای شبکه‌های عصبی (با دستور `nnTool`)، کاربردهای منطق فازی (با دستور `fuzzy`) برای فازی و `anfisedit` برای سیستم استنتاج عصبی-فازی) و الگوریتم ژنتیک (با دستور `optimtool('ga')`) را نام برد. اما از اشکالات عمده‌ای که در اکثر وبلاگ‌ها نیز به آن اشاره شده بود، سرعت پایین اجرای برنامه‌های متلب، در محاسبات با تعداد زیاد و داشتن GUI ساده است که اجازه `design` مناسب را به ما نمی‌دهد. همچنین برای اجرای پروژه لازم است که برنامه متلب روی سیستم نصب باشد که انصافاً برنامه سنگینی هم هست! همچنین اگر بخواهیم در این پروژه به زبان `C#` برنامه‌نویسی کنیم، ارتباط دادن آن با متلب کار راحتی به نظر نمی‌رسد!

• **AForge.NET**: چارچوب کاری `AForge.NET`، مجموعه‌ای از کلاس‌ها و توابع در زمینه‌های بینایی ماشین (`Computer Vision`)، هوش مصنوعی (`Artificial Intelligence`)، پردازش تصویر (`Image Processing`)، شبکه‌های عصبی (`Neural Network`)، الگوریتم ژنتیک (`Genetic Algorithm`)، یادگیری ماشین (`Machine Learning`)، منطق فازی (`Fuzzy Logic`)، رباتیک (`Robotic`) و... به زبان `C#` و تحت دات نت ۲ می‌باشد. این کتابخانه قدرتمند، متن باز (`open source`) بوده و محققین و توسعه‌گران می‌توانند به راحتی از آن استفاده نمایند. کتابخانه `AForge.NET` تحت حمایت گوگل در بخش گوگل کد (`Google Code`) در حال توسعه می‌باشد.

### برخی از مزایای چارچوب کاری `AForge.NET`:

- رایگان (`Freeware`) و متن باز (`open source`)
- وجود مستندات مناسب (`Document`) به صورت فایل `CHM` و نیز مثال برای هر بخش (برنامه‌های نمونه)
- وجود انجمن `www.aforgenet.com/forum` برای طرح مسائل و پرسش و پاسخ
- وجود `DLL` برای هر کدام از بخش‌های: شبکه‌های عصبی، منطق فازی، پردازش تصاویر و... به منظور اضافه کردن به پروژه برای استفاده از کلاس‌ها و متدها
- ساده کردن برنامه نویسی در مفاهیم مختلف هوش مصنوعی با نوشتن حداقل کد ممکن
- وجود کد پیاده‌سازی کلاس‌ها و متدها و امکان تغییر آن‌ها در صورت نیاز

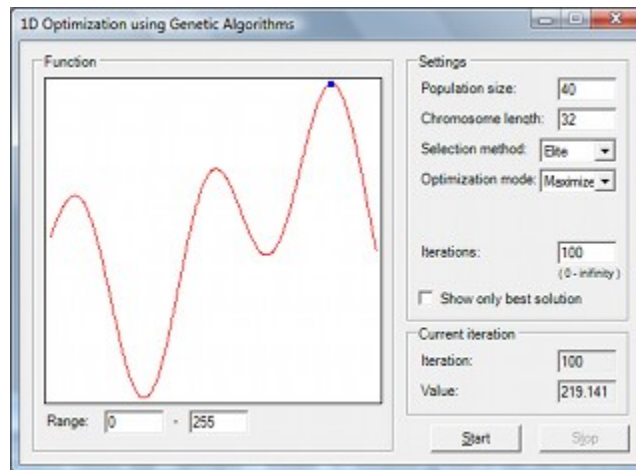
برای اطلاعات بیشتر درباره ی `AForge.NET` می‌توانید به لینک‌های زیر مراجعه کنید:

- <http://www.aforgenet.com/framework/>
- <http://www.aforgenet.com/framework/features/>
- <http://madmed88.tumblr.com/post/7999376854/tutorial-aforge-net-the-computer-vision-framework>

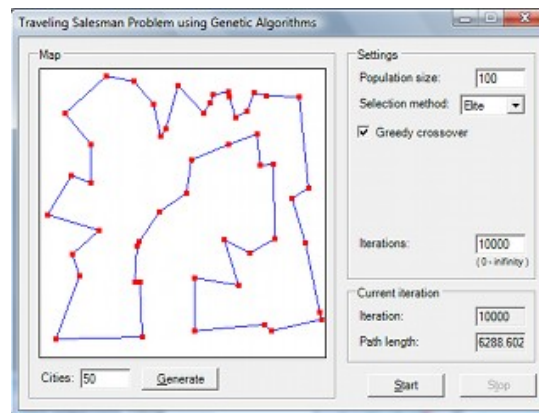
برای استفاده از AForge.NET در Visual Studio به لینک زیر مراجعه کنید:  
 (همچنین در بخش ضمیمه، راهنمای کار با AForge.NET در Visual Studio آورده شده است)  
<http://irslab.blogspot.nl/2010/07/aforgenet-visual-studio-2010-setup.html>

تصویر محیط برخی از برنامه‌هایی که با استفاده از AForge.net ایجاد شده‌اند، در ادامه آورده شده است:

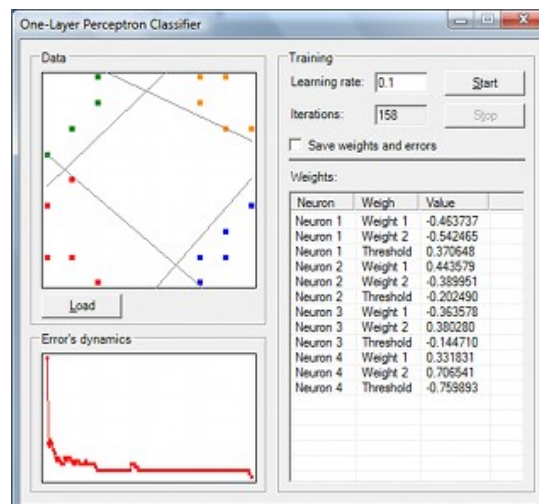
بهینه سازی با استفاده از الگوریتم ژنتیک



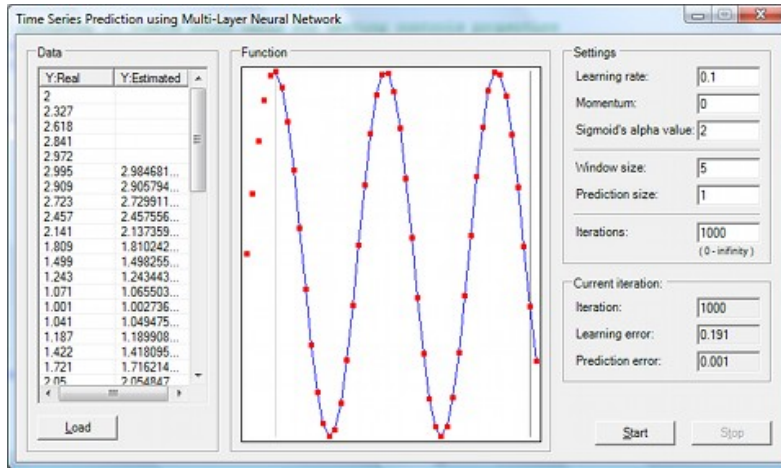
حل مسأله فروشنده دوره گرد با استفاده از الگوریتم ژنتیک



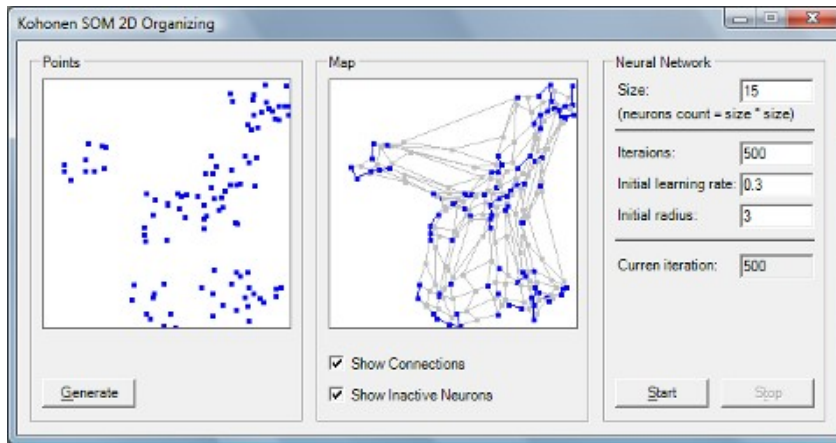
شبکه عصبی پرسپترون تک لایه برای دسته بندی داده ها (Classification)



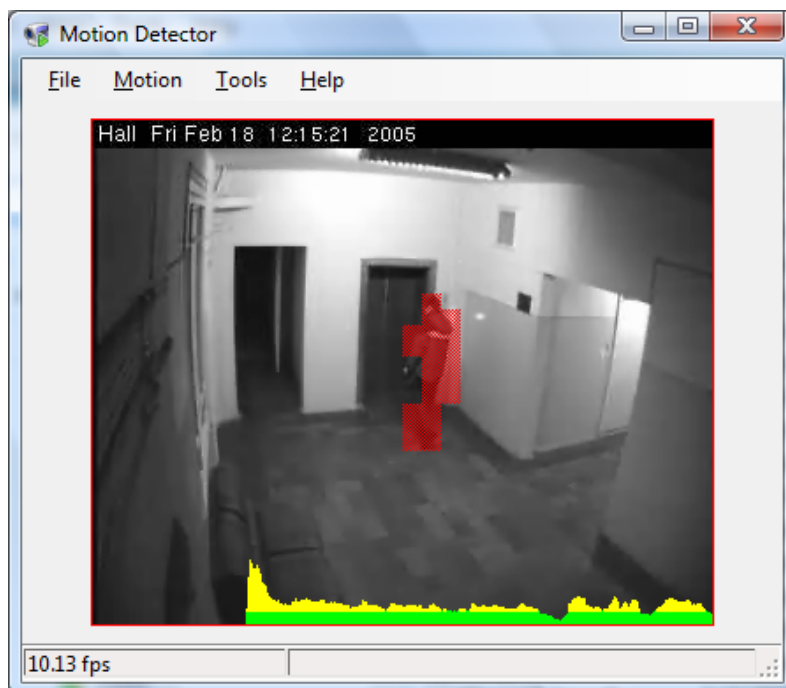
پیش بینی سری زمانی با استفاده از شبکه عصبی چند لایه



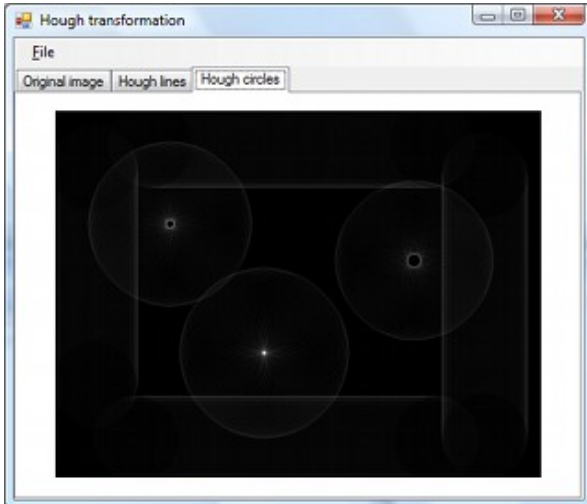
نگاشت های خود تنظیم (SOM) کوهونن



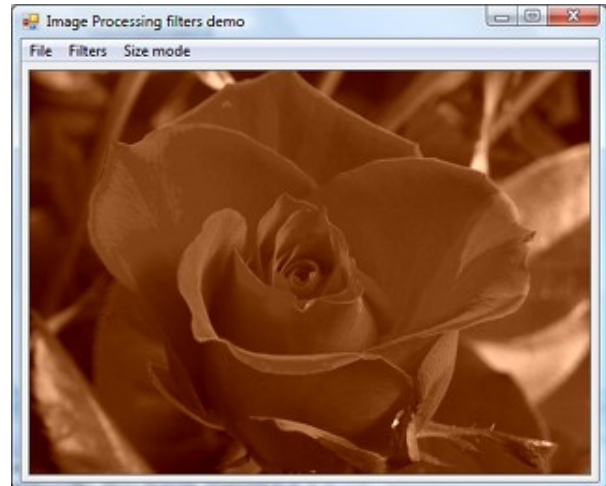
بینایی ماشینی برای تشخیص حرکت با استفاده از دوربین مدار بسته



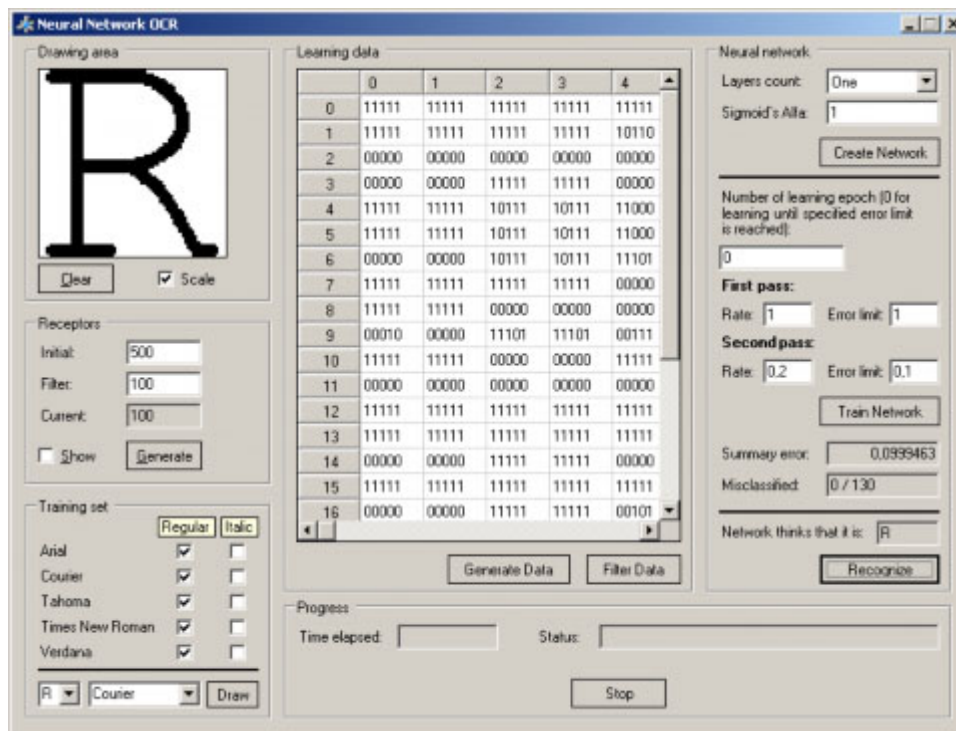
پیاده سازی نگاشت هاف (Hough Transformation)



پیاده سازی فیلترهای تصویری



تشخیص کاراکتر نوری یا تشخیص حروف (OCR) در زبان C# و با استفاده از شبکه عصبی



• **OpenCV (Open Source Computer Vision Library)**: کتابخانه‌ای از توابع برنامه‌نویسی است که عمدتاً برای بینایی کامپیوتر بلادرنگ<sup>۱</sup>، توسط Intel توسعه داده شده است. این یک کتابخانه‌ی رایگان است که به صورت چند سکویی<sup>۲</sup> می‌باشد. ویژگی‌های شاخص OpenCV را می‌توان به صورت زیر بیان کرد:

- کتابخانه‌ی بینایی کامپیوتر منبع باز (Open source) در زبان های C/C++
- بهینه‌سازی شده برای کاربردهای بلادرنگ
- عدم وابستگی به سیستم عامل، سخت‌افزار و مدیریت ویندوز

- ذخیره و بازیابی و استفاده از تصاویر و ویدیوها
- دارا بودن هر دو رابط کاربری (API) سطح بالا و پایین
- Provides interface to Intel's Integrated Performance Primitives (IPP) with processor specific optimization (Intel processors).

### ماژول‌های اصلی OpenCV:

- *cv*: توابع اصلی OpenCV
  - *cvaux*: توابع کمکی (عملی) OpenCV
  - *cxcore*: پشتیبانی از ساختمان داده‌ها و جبر خطی
  - *highgui*: تابع GUI
- تمرکز OpenCV بر روی پردازش تصویر بلادرنگ است که کاربردهایی از جمله تشخیص چهره، ابزارهای ۲-بعدی و ۳-بعدی، تشخیص حرکت و پیگیری حرکت، تشخیص اشیاء و شناسایی ساختار از روی حرکت اشیاء را دارد. OpenCV برای پشتیبانی از کاربردهایی که ذکر شد، کتابخانه‌ای از ابزارهای یادگیری ماشین را نیز به همراه دارد:

- [Boosting \(meta-algorithm\)](#)
- [Decision tree learning](#)
- [Gradient boosting trees](#)
- [Expectation-maximization algorithm](#)
- [k-nearest neighbor algorithm](#)
- [Naive Bayes classifier](#)
- [Artificial neural networks](#)
- [Random forest](#)
- [Support vector machine \(SVM\)](#)

برای اطلاعات بیشتر در مورد OpenCV می‌توانید به آدرس‌های زیر مراجعه کنید:

- <http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html>
- <http://code.google.com/p/opencvdotnet/>

#### Tutorial:

- <http://code.google.com/p/opencvdotnet/wiki/Tutorial>

همچنین برای کسب اطلاعاتی درباره نحوه ی نصب آن در Visual Studio 2008، می‌توانید به لینک زیر مراجعه کنید:

- <http://www.mygeeksite.in/2012/03/how-to-setup-opencv21-in-visual-studio.html>

از نسخه OpenCV 2.1 به بعد می‌توانیم از CMake جهت ایجاد و مدیریت پروژه‌های OpenCV استفاده کنیم. CMake یک سیستم ساخت چند سکویی و منبع باز است و جهت کنترل فرآیند کامپایل با استفاده از فایل متنی ساده جهت تعریف فرآیند کامپایل با کامپایلر و سکوی مستقل به کار می‌رود.

با استفاده از CMake می‌توانیم پروژه‌های را در سیستم عامل خودمان (به عنوان مثال لینوکس) ایجاد و کامپایل نموده و روی آن کار کرده و سپس نسخه‌هایی هم برای دیگر سیستم عامل‌ها (به عنوان مثال ویندوز، OS X و غیره) ایجاد کنیم.

همچنین برنامه‌ای (در واقع Wrapper ای) به نام Emgu CV وجود دارد که کتابخانه‌های OpenCV را پوشش داده و قابلیت فراخوانی توابع آن از زبان‌های برنامه‌نویسی مختلف مانند C#, VB, VC++، IronPython را دارد. با کامپایل این برنامه‌ها در محیط MonoDevelop می‌توان آن‌ها را روی ویندوز، لینوکس، Mac OS X، iPad، iPhone و دستگاه‌های با سیستم عامل Android اجرا کرد. برای اطلاعات بیشتر درباره این برنامه می‌توانید به لینک زیر مراجعه کنید:

- [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page)

در جدول زیر این سه Framework با هم مقایسه شده‌اند:

MATLAB	AForge.NET	OpenCV
<ul style="list-style-type: none"> <li>• Integrates computation, visualization and programming in a easy to use environment</li> <li>• Has huge array of provided functions</li> <li>• Great interface for displaying and manipulating data while debugging</li> <li>• Excellent for math and computation , algorithm development, modeling, simulation and prototyping</li> <li>• Excellent documentation</li> <li>• Very slow while processing images</li> <li>• Hardly use for real time image processing</li> </ul>	<ul style="list-style-type: none"> <li>• C# library</li> <li>• Designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence</li> <li>• Excellent documentation</li> <li>• Excellent for different transforms and image manipulation</li> <li>• Easier than openCV and harder than matlab</li> <li>• Poor in matrix operations</li> <li>• Can be used for real time image processing</li> <li>• Not fast and efficient as openCV to process images</li> </ul>	<ul style="list-style-type: none"> <li>• C/C++ library</li> <li>• Open source</li> <li>• Has Windows, Linux and Mac versions</li> <li>• Has basic data structures for matrix operation and image processing</li> <li>• Code is highly optimized for image processing</li> <li>• Fast and efficient</li> <li>• Bad documentation</li> <li>• Don't handle errors</li> <li>• Can be recommended for any complex real time image processing</li> </ul>

اگر بخواهیم این جدول را خلاصه کنیم، متلب آسان‌ترین و کاراترین راه برای پردازش تصویر است و OpenCV کاراترین و البته سخت‌ترین راه! AForge.NET کیفیت‌هایی بین این دو را ارائه می‌دهد. البته به لحاظ گرافیکی و User Friendly بودن می‌توان بر AForge.NET تکیه بیشتری کرد.

چارچوب‌های نرم‌افزاری دیگری هم هستند که در صورت لزوم می‌توانیم در مورد آن‌ها تحقیق کنیم، مثلا:

**ROS -**

[http://en.wikipedia.org/wiki/ROS\\_\(Robot\\_Operating\\_System\)](http://en.wikipedia.org/wiki/ROS_(Robot_Operating_System))

**VXL -**

<http://en.wikipedia.org/wiki/VXL>

**IVT -**

[http://en.wikipedia.org/wiki/Integrating\\_Vision\\_Toolkit](http://en.wikipedia.org/wiki/Integrating_Vision_Toolkit)

در پایان به بیان نظرات برخی از برنامه‌نویسان که تجربیات خود را در سایت‌های معتبری مانند StackOverflow, CodeProject و... قرار داده‌اند، می‌پردازم:

### **Matlab vs Aforge vs OpenCV**

- Combine AForge with a good linear algebra library and you might have something really useful though. As for OpenCV it has a rich feature set but is a bit tricky to program with.

- 
- Well, why not using both?? I am using (literally in C#) EMGUCV.NET (which is an OpenCV C# wrapper) and AForge.NET at the same time: AForge.NET plus its "Image Processing Lab" makes much sense for filtering options (edge detection, thresholds, and so forth) and easing viewing functionalities. OpenCV provides all the rest like SIFT/SURF and other more sophisticated image processing routines. That's why you use .NET: simply bring all the components you need together in one application. So for the final take: both will suit you well, OpenCV is more widely used/supported (though most user-base is C++), AForge is easier. It all comes down to what you want to do.

- 
- Both AForge and OpenCV has a built-in interface to .Net, and OpenCV also with C++, python, and more. Matlab might be more efficient, but if you don't have any experience with it - you should also learn its syntax. Take it into consideration.

- 
- I found AForge a lot friendlier and easier to use since you can easily adjust the processing parameters from a GUI (not so fast/easy) to do in Matlab/simulink. I'd go for Aforge.Net. It's also fast enough if you're worrying about processing speed. I initially started my project using OpenCV but i changed to Aforge.net. The reason being the the documentation was easier for me to understand and it made use of C#, which simplified my work. I have no c# experience and the reason i've began learning it was due to Aforge.net. Pretty simple once you get hands on experience.

- 
- Many people find OpenCV a bit hard to use, but they like AForge.NET. I think it all depends on experience - experienced people may prefer OpenCV since it is C library and may be used as on Windows, as on Linux, and provide good performance. So you better decide which language you want to use and which task you want to solve. Think about which image processing routines you need for your task. Then check both and find if these routines present in these libraries.



## نتیجه گیری

نتیجه گیری آخر هم اینه که به نظرم چون ما قراره با زبان C# کار کنیم، بهتره که از Aforge.NET و Emgu CV با هم استفاده کنیم.

آقای مجتبی هروی هم روی نرم افزار میکروسافت رباتیک (MRDS) کار کردند و موارد مهم را در اختیار دیگر اعضا قرار دادند:

### «آشنایی با نرم افزار میکروسافت رباتیک»

#### (Microsoft Robotics Developer Studio)

وسعت این موضوع خیلی زیاد هستش و تا حد ممکن سعی می کنم که فقط نکات کلیدی و مقدمات و کاربردهای مورد نیاز خودمون تو این بحث رو بگم و در این تحقیق حرفی از کد یا اجرای محیط آن نمی زنم و وارد این بحث نمی شم.

در سال ۲۰۰۶ شرکت میکروسافت محصولی به نام رباتیک استودیو را معرفی کرد (اولین نسخه از این محصول). این نرم افزار که هنوز در حال کامل شدن هستش و نسخه های آزمایشی اون روانه بازار شده، میتونه کمک شایانی به یادگیری و فعالیت در زمینه رباتیک بکنه، جا داره بگم که با این شرایط که هنوز نرم افزارش کامل نشده و در ابتدا کار کردن باهاش کمی پیچیده هست، ولی به طور چشم گیری در پروژه های بزرگ به کار گرفته شده. همانطور که می دانید ساخت یک روبات شامل ۲ بخش اصلی مکانیک و الکترونیک است، در مورد مدارات الکترونیک می توان قبل از ساخت، آن ها را شبیه سازی کرد، که بعضی از بچه های ما باهاش آشنایی دارند. اما برای قسمت مکانیک روبات ها شما می توانید به راحتی قبل از ساخت یک روبات بخش مکانیکی آن را توسط نرم افزار Microsoft Robotics Developer Studio شبیه سازی کنید و از کم و کیف مکانیک ربات خود آگاه شوید.

MRDS یک پلتفرم برای برنامه نویسان و محیطی برای کنترل و شبیه سازی ربات ها بر پایه ی سیستم عامل ویندوز می باشد، که این محیط از زبان های C#، VB.net، JScript و IronPython برای نوشتن و دیباگ کردن برنامه برای ربات ها پشتیبانی میکنه و از آن می توان برای مقاصد علمی، آکادمیک، تجاری، نظامی، دانشگاهی، تفریحی و صنعتی و به طور خلاصه برای مدیریت گسترده وسیعی از ربات ها استفاده نمود، همچنین انواع مختلفی از سخت افزارهای مورد استفاده در روبات ها رو پشتیبانی می کند، اگه بخوام به عنوان یک مثال کوچیک بگم، میشه از دوربین ها و حسگرهای دیداری و شنیداری و صحبت کردن در ربات ها و کنترل حرکت ربات نام برد و یا حتی تشخیص صدا و تصویر که بسیار در کار ما حائز اهمیت هستش.

به طور کلی MRDS در ساخت نرم افزارهای مبتنی بر سرویس برای انواع مختلف سخت افزار کاربرد دارد. MRDS بیشتر از یک نرم افزار برای بازی با روبات هاست. یکی از مهمترین نکات مربوط به نرم افزارهای روباتیک این است که کدی که نوشته اید در محیط شبیه سازی و محیط واقعی به یک صورت عمل کند، یعنی بدون دستکاری کد روبات شما همانطور که در محیط شبیه سازی شده کار می کرده است، در محیط واقعی نیز کار کند، که MRDS به خوبی این نکته را پیاده سازی کرده است. شما روبات خود را با استفاده از ابزارهای بصری مدل می کنید و کدهای آن را می نویسید، سپس در محیط ۳ بعدی رفتار روبات را شبیه سازی می کنید، آن را تست و عیب یابی می کنید و در نهایت در محیط واقعی پیاده سازی اش می کنید.

## «برخی از امکانات این نرم افزار»

۱. پشتیبانی وسیع از انواع سخت‌افزارها.

این نرم‌افزار چگونگی کار با بسیاری از ربات‌ها و سخت‌افزارهای معروف را در خود دارد. به عنوان مثال Generic Articulated Arm یا iRobot.

۲. امکان استفاده کاربرانی که دانش برنامه‌نویسی ندارند.

کاربرانی غیر برنامه‌نویس می‌توانند با استفاده از Microsoft Visual Programming Language به مدیریت و انجام امور مورد نظرشان بپردازند، به این صورت که آن‌ها بلاک‌هایی از پیش تعریف شده را در محیط نرم‌افزار قرار می‌دهند، تنظیمات هر یک را انجام و آن‌ها را به یکدیگر متصل می‌کنند. پس از آن حتی می‌توانند این مجموعه را به صورت یک بلاک در آورند تا در آینده نیز بتوانند از آن استفاده نمایند.

۳. شبیه‌سازی رفتار ربات در فضاهای شبیه‌سازی شده سه‌بعدی.

شما می‌توانید به وسیله تکنولوژی Microsoft Visual Simulation و NVIDIA PhysX تعبیه شده در این نرم‌افزار ربات‌ها را در فضاهای سه بعدی شبیه‌سازی شده قرار دهید و در یک جهان مجازی با اعمال قوانین پیشرفته فیزیکی بر آن‌ها از رفتارشان تخمینی صحیح به دست آورید و در صورت لزوم اصلاحات لازم را انجام دهید.

۴. ارتباط برقرار کردن با ربات از طریق وب یا برنامه کاربردی.

شما می‌توانید به وسیله مرورگرهای وب به ربات دستورات خود را ارسال کنید و پاسخ‌ها را دریافت نمایید، دقیقاً به همان کیفیت و دقتی که می‌توانید از طریق رابط ویندوزی این نرم‌افزار چنین وظایفی را انجام دهید. همچنین برای تست و خطایابی ربات می‌توانید از این رابط‌های مبتنی بر ویندوز و وب استفاده کنید. این رابط‌ها به شما کمک می‌کنند تا به صورت ریموت ربات خود را کنترل و عیب‌یابی کنید.

۵. دستیابی آسان و سریع به سنسورها و فعال‌کننده‌های ربات.

این نرم‌افزار را می‌توان با فراخوانی توابع کتابخانه‌ای‌شان مثلاً در محیط ویژوال استودیو و در بخش سی‌# (بعد از نصب نرم‌افزار این توابع در داخل کتابخانه مربوط به ویژوال استودیو قرار می‌گیرند) برای مسائل و بررسی‌های تخصصی‌تر و بیشتر مورد استفاده قرار داد. به عنوان مثال Microsoft.Robotics.Services، Microsoft.Dss.Core، Microsoft.Ccr.Core و W3C.Soap.

۶. امکان تقسیم پروژه‌ها به بخش‌های کوچکتر.

در این نرم‌افزار امکاناتی در نظر گرفته شده است که به وسیله آن‌ها می‌توان کار را به قسمت‌های کوچکتر تقسیم کرد تا به راحتی قابل انجام باشد.

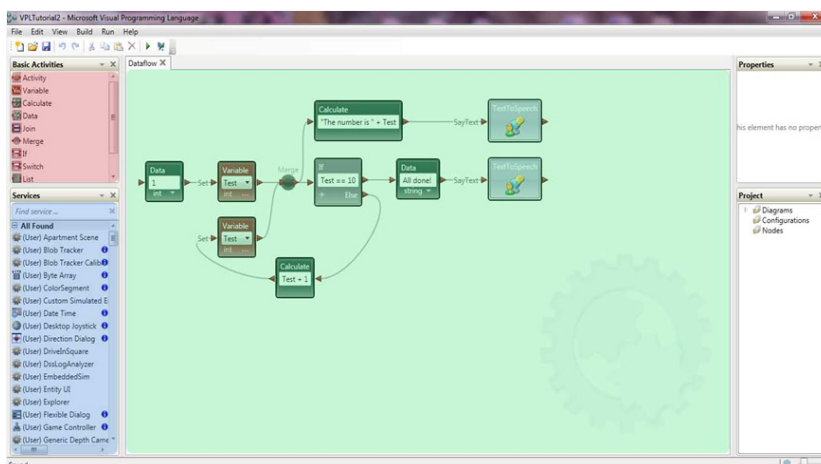
۷. قابلیت توسعه امکانات.

به سادگی می‌توان قابلیت‌ها و امکانات این نرم‌افزار را توسعه داد و شرکت‌ها و افراد متعددی در سراسر جهان هم اکنون در حال انجام این کار هستند.

و بسیاری قابلیت‌های دیگر.

## «اجزای MRDS»

۱. محیط زبان برنامه نویسی ویژوال (Visual Programming Language):



همان‌طور که در بالا توضیح دادم VPL یک محیط توسعه گرافیکی است که با دیاگرام‌های مختلف رفتار را تعیین می‌کند. کاربر می‌تواند به صورت گرافیکی با ربات خود تعامل داشته باشد. یک سرویس یا یک فعالیت به صورت یک بلاک که دارای ورودی و خروجی است نشان داده می‌شود. کاربر فقط باید این بلاک‌ها را از کاتالوگ انتخاب و آن را به دیاگرام خود اضافه نماید. اتصال میان این بلاک‌ها به وسیله ماوس کشیده می‌شوند و جزئیاتی مانند نحوه عملیات و مقادیر ارسالی مشخص می‌شوند. VPL می‌تواند از روی سرویس‌های موجود در دیاگرام که توسط کاربر ایجاد شده‌اند، کد مربوطه را تولید کند. با VPL حتی افرادی که تا به حال یک خط برنامه ننوشته‌اند نیز می‌توانند ربات بسازند!

همان‌طور که ملاحظه می‌کنید در شکل بالا چند بخش با رنگ‌های متفاوتی مشخص شده که بخش سبز رنگ که در مرکز قرار دارد مربوط به محل ترسیم دیاگرام فرایند کار ربات می‌باشد. بخش دیگری که با رنگ قرمز مشخص شده و در سمت چپ بالا قرار دارد، ماژول‌های فعال ساز پایه ای است که کارهایی مثل شرط گذاری، تعریف داده و محاسبات روی آنها را در بر می‌گیرد. و در نهایت بخش آبی رنگ که در سمت چپ پایین قرار دارد، مربوط به سرویس‌هایی می‌باشد که توسط سیستم شما قابل دستیابی است و یا در سخت افزارهای مجازی برنامه برای شما تعریف شده است، به طور مثال اگر سیستم شما به سخت افزار دوربین یا وب کم مجهز باشد در این بخش شما توانایی بهره‌گیری از آن را در طراحی ربات دارید.

۲. Concurrency and Coordination Runtime یک کتابخانه دات نت فریم ورک است که پردازش غیر همزمان (آسنکرون) را پشتیبانی می‌کند. استفاده از پردازش آسنکرون برای ساخت یک روبات بسیار مهم است، زیرا سنسورها ممکن است به صورت همزمان داده‌ها را ارسال و دریافت کنند. به عبارتی دیگر CCR موجود در این نرم افزار امکان پردازش موازی و همزمان ورودی‌هایی که از سنسورهای مختلف ربات می‌آید را فراهم می‌سازد.

۳. Decentralized Software Services یک ران تایم مبتنی بر سرویس بسیار سبک است که از مدل REST برای ارتباط استفاده می‌کند. قابلیت DSS امکان ساده سازی دسترسی و پاسخ به وضعیت ربات را به وسیله رابط کاربری تحت وب و یا برنامه کاربردی ویندوزی می‌دهد و همچنین مشاهده بلادرنگ حسگرهای ربات و پاسخ به موتورها و محرک‌ها را آسان می‌کند.

۴. Visual Simulation Environment که محیطی برای شبیه سازی روبات ساخته شده در MRDS است. شما می‌توانید در این محیط مجازی DSE رفتارهای روبات خود را در شرایط مختلف مشاهده کنید.

## جلسه چهارم

در جلسه‌ای که روز چهارشنبه ۹۱/۱۲/۲ برگزار شد، ابتدا به مرور Paper Work های تهیه شده توسط برخی از افراد و به اشتراک گذاشتن اطلاعات یکدیگر پرداختیم و بعد از آن به بیان یک چارچوب کلی از سیستم مورد بررسی و تقسیم وظایف در فیلدهای مختلف پرداختیم. لازم به ذکر است که در فاصله این ۳ ماه جلسه حضورى تشکیل نشد اما از طریق سایت افراد گروه با یکدیگر در ارتباط بودند و تحقیقات خود را انجام می‌دادند. نتایج بررسی‌ها به ترتیب زیر می‌باشد:

ادامه روند کار به سه بخش تقسیم شد:

یک گروه برنامه‌نویسی روی ANFIS کار خواهد کرد.

یک گروه روی بحث گرافیک پروژه کار خواهد کرد

یک گروه روی بحث مهندسی نرم‌افزار کار خواهد کرد

وظایف به صورت زیر به افراد محول شد:

۱- مریم مجاور رضایی: تهیه DataBase انواع موبایل + مهندسی نرم‌افزار

۲- عاطفه بابایی: مطالعه و استخراج منابع فروشندگی

۳- حوری رضوی: مهندسی نرم‌افزار

۴- رضوان خاندانی: مهندسی نرم‌افزار

۵- مجتبی هروی: گرافیک + MRDS

۶- محسن اسماعیلی: گرافیک + MRDS

۷- مهدی عزیزی: AForge.Net + طراحی DataBase

۸- پگاه خداکرمی: AForge.Net + Documentation

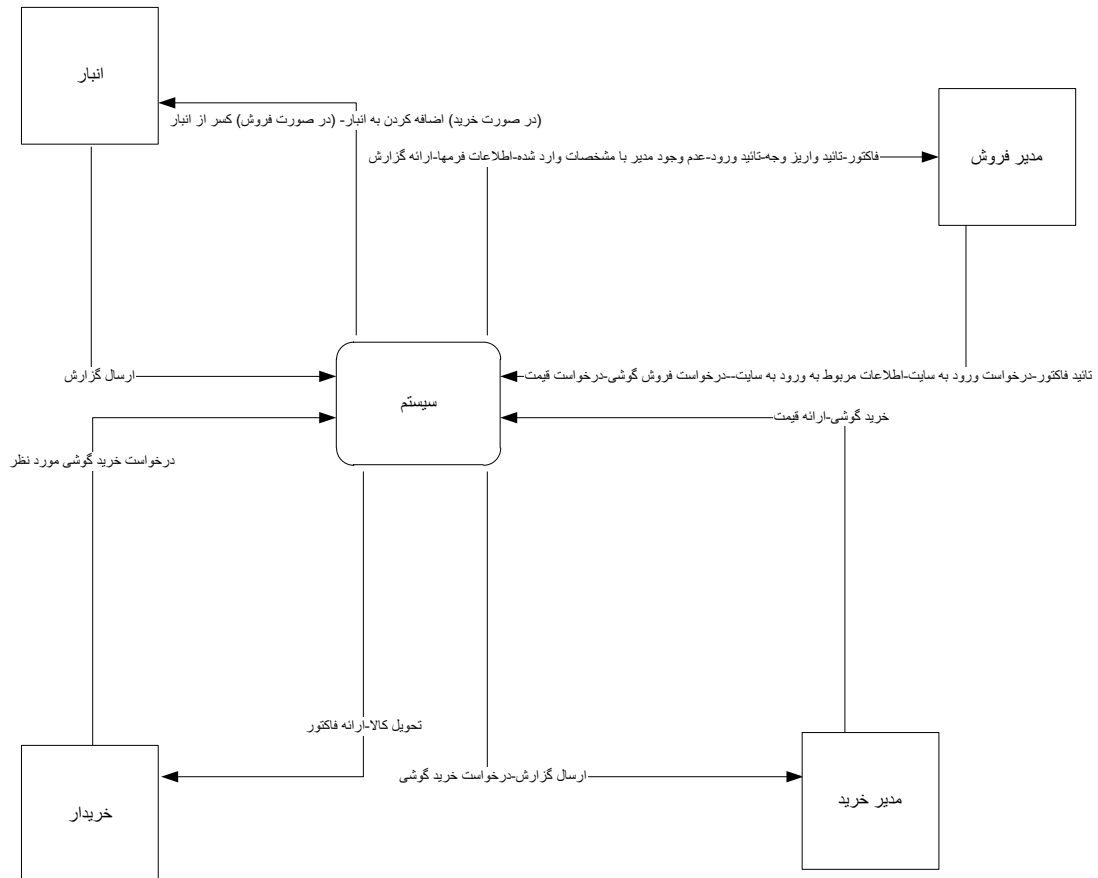
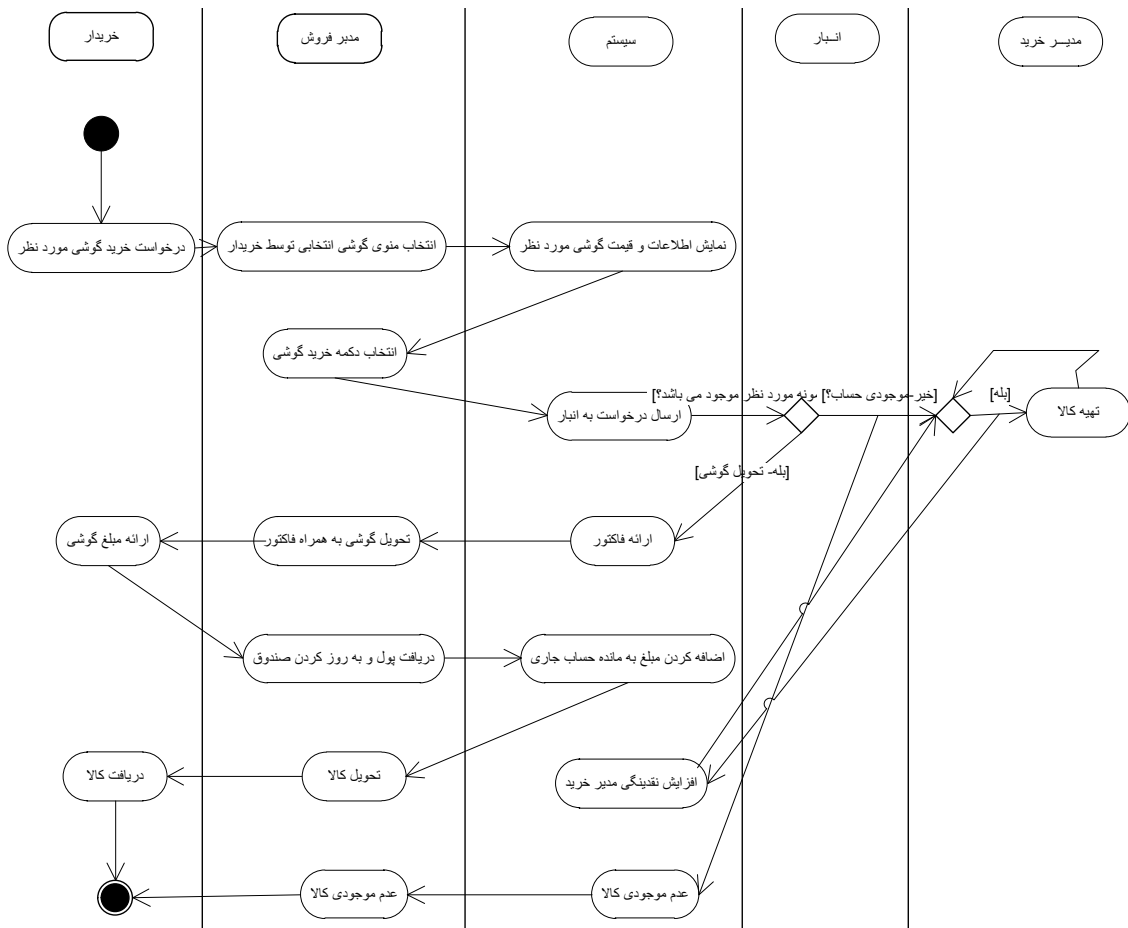
۹- حسن حکمی: برنامه‌نویسی + ANFIS

۱۰- صادق حسینی: برنامه‌نویسی + ANFIS

برای این بخش خانم مریم مجاور رضایی نمودارهایی را آماده کردند که در مراحل بعدی تکمیل تر خواهد شد.

همچنین آقای رضوان خاندانی یک پیشنهاد اولیه برای مسئله مطرح کردند (در قالب یک جدول و یک فایل پاورپوینت که ضمیمه خواهد شد). البته این یک طرح کلی می‌باشد که در آینده کامل تر خواهد شد.

نمودارها و جدول‌ها به ترتیب در صفحات ۳۷، ۳۸ و ۳۹ آورده شده‌اند.



خروجی هر مرحله	توابع و عملیات برنامه‌نویسی	فرآیند و روند خرید	
مشخصات اولیه مشتری: (زمان ورود، جنسیت، محدوده سنی و ...)	الف) تابع تولید مشخصات تصادفی ب) تابع تولید زمان تصادفی برای ورود مشتری ج) اجرای تابع ورود به فروشگاه	مشتری با مشخصات کاملاً تصادفی وارد فروشگاه می‌شود.	۱
	تابع تولید عدد تصادفی	مشتری لحظاتی را در فروشگاه چرخ می‌زند و از گوشی‌ها دیدن می‌کند. ممکن است این بازه زمانی صفر باشد.	۲
مشخصات مشتری update می‌شود: (زمان ورود، جنسیت، محدوده سنی، حد و اندازه آشنایی با مدل‌ها و کاربردهای گوشی‌ها و توان مالی خرید مشتری)	پایگاهی از سوالاتی که مشتری می‌پرسد و پاسخ‌هایی که فروشنده می‌دهد.	مشتری با فروشنده صحبت می‌کند: الف) مشاوره و کمک گرفتن (از چه مدل‌ها و چه کارایی‌هایی؟) ب) گرفتن قیمت (در چه محدوده‌ای؟)	۳
علاقه‌مندی‌های مشتری به مشخصات قبلی‌اش اضافه می‌شود.	ثبت مشخصات مشتری و اجرای تابع خروج از فروشگاه	الف) مشتری خرید انجام نمی‌دهد و از فروشگاه خارج می‌شود	۴
پیشنهاد دقیق‌ترین یا مشابه‌ترین مدل‌های گوشی به علاقه‌مندی‌های مشتری توسط سیستم با حفظ سود قابل قبول!	الف) پر کردن فرم درخواست خرید، توسط مشتری جهت پیشنهاد از سوی سیستم ب) ثبت علاقه‌مندی‌های مشتری (با توجه به اطلاعات فرم) ج) چک کردن انبار (پایگاه داده) و اطلاع از موجودی‌ها د) اجرای عملیات مربوط به شبکه عصبی یا فازی یا هر دو بر اساس مشخصات مشتری و علاقه‌مندی‌های مشتری	ب) مشتری قصد خرید دارد	
	چانه زدن مشتری و عملیات مخ زنی جهت افزایش سود	توافق بین مشتری و فروشنده	۵
الف) افزایش موجودی دخل ب) ثبت گوشی فروخته شده ج) به روز رسانی پایگاه داده	ثبت مشخصات مشتری و اجرای تابع خروج از فروشگاه الف) تابع دریافت پول به صورت دستی یا تابع دریافت پول با دستگاه کارت خوان ب) تابع پر کردن فاکتور توسط سیستم ج) تابع تحویل فاکتور به مشتری د) ثبت فروش انجام شده و به روز کردن پایگاه داده ه) ثبت مشخصات مشتری و اجرای تابع خروج از فروشگاه	نتیجه‌گیری کلی مشتری: الف) مشتری خرید انجام نمی‌دهد و از فروشگاه خارج می‌شود ب) خرید انجام می‌شود.	۶

<p>موقعیت افراد موقعیت اجناس موقعیت فروشندها</p>	<p>Rule هایی که در لحظه اتفاق می افتند</p>
--	--

ویژگی های هر فرد قبل از ورود به صورت Random، تنظیم می شود و بعد از خروج نیز در Database ذخیره می شود.

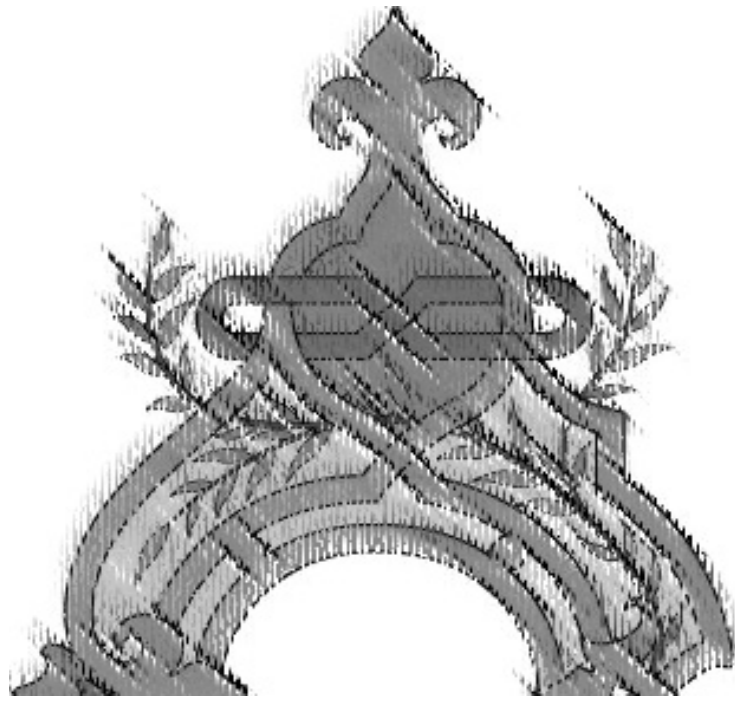
حافظه  
بانک عکس  
.  
.  
.

در پایان با توجه به روند کلی پروژه باید فرایندهای زیر را برای ادامه کار به انجام برسانیم:

- سناریوها را با توجه به دنیای واقعی گسترش داده و دست به استخراج قوانین اگر-آنگاه بزنیم.

- با توجه به پایگاه داده تهیه شده از مدل های موبایل و ویژگی هایی که برای آنها در نظر گرفته شده، شامل قیمت، کیفیت برند، دسته بندی براساس امکانات و غیره، متغیرهای زبانی را استخراج کرده و سه مرحله عملیات فازی شامل فازی سازی، استنتاج قوانین و غیرفازی سازی را به انجام برسانیم. در این میان، انتخاب صحیح متغیرهای زبانی با توجه به رفتار مشتری در دنیای واقعی، انتخاب صحیح توابع عضویت برای آنها و هم چنین تنظیم پارامترهای توابع با آموزش شبکه های عصبی از اهمیت بالایی برخوردار است.

لازم به ذکر است که این گزارش فقط مربوط به فعالیت هایی است که تاکنون Document شده است و افراد گروه هر کدام در بخش های مربوط به خودشان کار می کنند که نتایج آنها در گزارش های بعدی بیان می شود.



گزارش دوم

Second Report





## جلسه پنجم

در جلسه‌ای که در روز سه‌شنبه ۱۳۹۲/۸/۷ برگزار شد، به بررسی بخش‌های مختلفی که در پروژه کاربرد دارند پرداختیم:

- استفاده از AHP برای استخراج دانش

- استفاده از ANFIS برای منطق برنامه که خروجی آن چیزی شبیه درخت تصمیم باشد

- Test Case برای به دست آوردن نقاط گوشه‌ای

- مهندسی نرم‌افزار

- گرافیک: Microsoft Robotics یا هر نرم‌افزار دیگری که برای GUI پروژه مناسب باشد (یعنی یک سری Agent هستند که

طبق منطق برنامه باید یک سری اعمال را انجام دهند. در واقع پروژه می‌تواند به جای شیء‌گرا بودن، عامل‌گرا باشد)

- مراحل برنامه‌نویسی شامل سه لایه زیر می‌باشد:

### Interface

### Logic

اطلاعات دیتابیس به صورت

### DataBase

دینامیک ذخیره می‌شوند

۴۱

- برای بخش مربوط به شبکه عصبی مراحل Train و Test انجام می‌شود. پس سناریوها باید در حافظه save شوند. بین خریدارها هم می‌تواند ارتباط وجود داشته باشد

در این جلسه قرار گذاشته شد که هر کسی ۱۰ سناریو بنویسد که از مجموعه آن‌ها، سناریوهای مورد قبول استخراج شود. همچنین آقای مهدی عزیزی مطالبی را در مورد فروشنده موفق بودن share کرده بودند که گزیده‌ای از آن به شرح زیر است:

### نکاتی که یک فروشنده موفق باید بداند

یک مشکل همیشگی و همه گیر وجود دارد و آن عدم نتیجه‌گیری در زمان فروش است. گویا ما عادت کرده‌ایم همیشه از اوضاع بنالیم و این عدم نتیجه‌گیری را به دیگر موضوعات ربط بدهیم. شاید همین موضوع باعث می‌شود ما کمتر هم نتیجه بگیریم. تکنیک‌های موفقیت در فروش و فروشنده‌گی به این صورت است که شما باید در زمینه نهایی کردن فروش حرفه‌ای شوید تا دیگر شکست در فروش را تجربه نکنید. مراحل نهایی کردن فروش:

نهایی کردن فروش دارای پنج مرحله است که این مراحل به صورت پیوسته و متصل دنبال می‌شوند و بین آن‌ها فاصله یا وقفه‌ای وجود ندارد. اما هر زمانی که فروشنده در به ثمر رساندن خود توفیقی به دست نیابد لازم است یکایک مراحل مذکور را بررسی کند و اشتباه خود را دریابد.

### پنج مرحله نهایی کردن فروش شامل:

۱- به میزان کافی مشتری احتمالی را علاقه‌مند و دلبسته کنید تا حاضر شود با شما صحبت کند.

**نکته:** معمولاً افرادی که برای خرید مراجعه می‌کنند یا فرصت کافی برای خرید ندارند و یا ترجیح می‌دهند از میان فروشگاه‌ها و مؤسسات مختلف، دیگر گزینه‌ها را نیز بررسی کنند. پس اولین گام را محکم بردارید تا هرگز مشتری را از دست ندهید.

۲- آنقدر او را در حالت علاقه‌مندی نگاه دارید تا تشخیص دهید مشتری به چه چیز محتاج است.

**نکته:** این هنر یک فروشنده موفق است تا بتواند در کوتاه‌ترین زمان خواسته‌های مشتری را تشخیص دهد، در عین حال نگذارد مشتری خسته و کلافه شود و میزان علاقه‌مندی خود را از دست بدهد. بهترین مثال برای کسانی که از این تکنیک استفاده می‌کنند قصه‌گوها هستند و می‌دانند چگونه قصه را به مخاطبان‌شان عرضه کنند که تا لحظه آخر به داستان علاقه‌مند بمانند. شما نیز دقیقاً باید همین کار را بکنید.

۳- با توضیح فواید کالایی که عرضه می‌کنید او را برای خرید علاقه‌مند سازید.

**نکته:** حتماً بارها کسانی را دیده‌اید که برای خرید کالایی خاص به بازار رفته‌اند ولی با یک سبد از کالاهایی دیگر بدون خرید آن کالای خاص برگشته‌اند. تمامی فروشندگان که موفق به فروش دیگر کالاها شده‌اند توانسته‌اند فواید کالای خود را به مشتری عرضه کرده و او را به خرید ترغیب کنند. پس، از این نکته در فروش استفاده کنید.

۴- از مشتری احتمالی بخواهید به شما سفارش خرید بدهد.

**نکته:** اگر نتوانید از مشتری سفارش بگیرید پس چرا فروشنده شده‌اید؟ شما تمام تلاش خود را می‌کنید تا بتوانید کالا یا خدمات خود را بفروشید، پس مهمترین قسمت فروش همین جاست. دقیقاً مثل این می‌ماند که تمامی مواد یک کیک را به میزان مشخص با هم مخلوط کنید ولی آن را در فر نگذارید. با این کار هم کلی مواد اولیه را دور ریخته‌اید و هم نتوانسته‌اید کیک بخورید.

۵- کوشش کنید برای انتقادات و ایرادهای مشتری راه‌حل‌های منطقی ارائه دهید.

**نکته:** شما یا باید بتوانید یک مجموعه کالا را بارها به یک مشتری بفروشید یا به دنبال مشتریان جدید بگردید. حتماً دلتان نمی‌خواهد چهار مرحله قبل را دوباره تکرار کنید. پس به دنبال راهکارهایی برای حل منطقی انتقادهای مشتریان بگردید تا مشتریان خود را از دست ندهید.

## اطلاعات لازم برای موفقیت در فروش:

۱- اطلاع از محصول:

\* داشتن اطلاعات کافی و دقیق درباره محصولات که عرضه می‌کنید.

\* داشتن اطلاعات کافی درباره محصولات رقبای.

\* داشتن اطلاعات جدید درباره تغییر و تحولات در سبک زندگی امروز و نیازهای جدید مشتریان.

۲- آگاهی از موارد استعمال محصولات:

\* تشریح فواید و امتیازات کالا برای مشتری.

\* آگاهی از چگونگی بهره‌برداری در محصول.

۳- دانش تبلیغات و بازاریابی

\* عواملی مانند طرز نمایش کالا موجب بالا رفتن حجم فروش می‌شود.

\* دخالت دادن اصول تبلیغات در کار فروش موجب افزایش تعداد مشتریان می‌شود.

\* پس یک فروشنده موفق تنها با داشتن دانش و اطلاعات می‌تواند موفق باشد.

### مشکل مشتری را باید مشکل خودتان بدانید!

ممکن است مشتری در انتخاب کالای مورد نظر خود مشکل داشته باشد. به او فرصت بدهید که مشکل خود را مطرح کند یا او را همراهی کنید و نشان دهید که مشکل او را مشکل خودتان می‌دانید.

فرض کنید مشتری دنبال مارک معینی می‌گردد، اندازه خاصی در نظر دارد، دنبال نوعی از جنس می‌گردد که با بودجه یا توان مالی یا مقدار موجودی‌اش تناسب داشته باشد یا ...

طوری رفتار کنید که در هر یک از این مسایل، مشتری شما را در کنار خود ببیند یا حس کند که می‌خواهید واقعا مشکل او را برطرف کنید. نتایج این گونه موفقیت‌های آینده و پیشرفت شغلی شما، گاه غیر قابل اندازه‌گیری است. فراموش نکنید شاید به ظاهر بدترین مشتری شما بهترین و وفادارترین مشتری شما شود.

آگاهی کامل از ویژگی‌های محصول، در موفقیت فروش از اهمیت خاصی برخوردار است.

منظور از آگاهی و دانش درباره محصول، کلیه حقایق و اطلاعاتی درباره محصول است که شامل:

۱- آگاهی از کاربردهای ممکن محصول.

۲- آگاهی از مواد و روش‌های به کار رفته در محصول.

۳- آگاهی از نحوه نگهداری محصول.

۴- آگاهی از شهرت شرکت.

۵- آگاهی از خدمات پس از فروش و ضمانت.

۶- آگاهی از خصوصیات محصول رقیب.

۷- آگاهی از قیمت کالای رقیب.

۸- آگاهی از محدودیت فروش.

فروشنده، نماینده مشتری برای شرکت است.

دیدگاه قدیمی این بود که فروشنده باید نگران فروش و شرکت در اندیشه فروش باشد اما دیدگاه امروز چنین است:

فروشنده باید به موارد زیر بیندیشد:

\* فروش

\* شیوه تأمین رضایت‌مندی

\* ایجاد سود برای شرکت

\* به داده‌های فروش و توان بالقوه بازار توجه کند.

\* داده‌ها و اخبار محرمانه را درباره بازار جمع‌آوری کند.

\* برنامه استراتژی‌های بازاریابی را تهیه و تدوین کند.

\* باید بداند چگونه تلاش‌های سازمان را در جهت افزایش اقبال مورد نظر مشتری و تامین رضایت او سوق دهد.

### ثبت سوابق و اطلاعات (پرونده مشتریان)

برای یک فروشنده، حفظ سوابق مشتریان و تهیه یک بانک اطلاعاتی کامل و دقیق از مشخصات مشتریان و سابقه مذاکرات، دفعات خرید، رضایتمندی و نارضایتی، سوابق پرداختی، سابقه معاملات، جزئیات شخصی و جغرافیایی و ... بسیار حائز اهمیت بوده و او را در موفقیت در فرآیند فروش یاری می‌دهد. وقتی در پرونده مشتری کلیه ویژگی‌های لازم برای بهتر شناختن وی و نیز سرویس‌دهی بهتر به وی را ثبت می‌کنید این کار به شما کمک می‌کند که در مواقع درست اقدام مناسب انجام دهید. اگر در سوابق مشتری خوش‌حسابی و بدحسابی وی را ثبت کنید این کار باعث می‌شود که در دریافت وجه به‌صورت مدت‌دار و نیز حجم فروش امانی به وی احتیاط کنید. وقتی ویژگی‌های جغرافیایی محل کسب و کار او را می‌دانید این باعث می‌شود که از طیف مناسب خدمت‌رسانی به وی به‌طور دقیق آگاه باشید. ثبت رضایتمندی و نارضایتی وی به شما کمک می‌کند که برخورد مناسب با او را تدبیر کرده و در جلب رضایت او به نحو بهتری عمل کنید. پس برای هر مشتری پرونده‌ای تشکیل دهید. برای ثبت مشخصات مشتریان از فرم‌هایی استفاده کنید که حاوی اطلاعاتی باشند که در خصوص آن‌ها نیاز دارید. این بانک اطلاعاتی می‌تواند بسته به نوع فعالیت شما حاوی اطلاعات مختلفی باشد. توجه داشته باشید که این بانک اطلاعاتی مرتباً باید به‌روز و اصلاح شود.

عوامل فراوانی بر ناموفق بودن فروش اثر می‌گذارد، ولی فقط یک دلیل عمده برای موفقیت فروش وجود دارد و آن رضایتمندی مشتری است. به عنوان فروشنده و کسی که درآمدش وابسته به رضایت مشتریان است، توجه و تمرکز بر نیازها، خواسته‌ها و رضایت مشتری، عامل اصلی موفقیت و کسب درآمد به شمار می‌رود.

### چطور یک فروشنده خوب باشیم؟

#### ۱- به خود اطمینان داشته باشید:

فروشنده‌ی هر چیزی که باشید، افکار و عقایدتان هر چه باشد، با اعتماد و اطمینان به نفس کامل وارد جلسه بحث شوید. مطمئن باشید که ایده‌های شما کمک‌کننده خواهند بود. البته همیشه به خاطر داشته باشید که به خاطر وقت و توجه مخاطبان از آن‌ها تشکر کنید. بهترین فروشندگان آن‌هایی هستند که پس از اولین خرید، مشتریان باز به سراغشان بروند. نه آن‌هایی که مثل برق و باد وارد شده، قراردادهای بزرگ می‌بندند و بعد ناپدید می‌شوند. گروه اول از فروشندگان هستند که برای کار خود ارزش و احترام قائلند.

#### ۲- مردم‌شناس باشید:

شناختن مشتریان اصلی‌ترین رکن در فروشندگی است. همه‌ی تلاش‌تان را بکنید تا مشتریان‌تان را، چه از نظر شخصی و چه کاری بفهمید و درک کنید. هر از چند گاهی با آن‌ها تماس گرفته و احوالشان را جویا شوید. گه‌گاه با آن‌ها بحث‌های کوچکی خارج از مسائل کاری داشته باشید. ولی حتی در این بحث‌ها نیز همیشه به دنبال فرصت‌های مناسب باشید.

#### ۳- صادق باشید:

حقیقت همیشه بهترین چیز است، به همین دلیل سعی کنید از همان ابتدا با مشتریان‌تان صادق و روراست باشید. خارج از مسائل مادی، اطمینان داشته باشید تا زمانی که مشتری از صحت و شایستگی قرارداد مطمئن نباشد، اطمینانی که شما به آن‌ها می‌دهید هیچ ارزشی نخواهد داشت. اگر همیشه با مشتریان صادق و روراست باشید، همیشه به یاد آن‌ها خواهید ماند و در زمان احتیاج به سراغتان می‌آیند.

#### ۴- فرصت‌های خوب را بشناسید:

هر روز که به سر کار می‌روید، چشم و گوش‌تان را باز نگاه داشته و هوشیار باشید. فرصت‌های خوب همیشه وجود دارند، اما فقط باید شما آن‌ها را تشخیص دهید. فقط زمانی فرصت‌های کوچک و پیش پا افتاده را نادیده بگیرید، که مطمئن باشید فرصت بزرگتری در دست دارید.

#### ۵- معاملات کوچک را دست کم نگیرید:

یک فروشنده خوب، برای معاملات کوچک نیز به اندازه‌ی قراردادهای بزرگ ارزش قائل است. از این گذشته، اگر می‌خواهید پیشرفت کنید، باید از معاملات کوچک شروع کنید. پس عجله نکنید. همین معاملات کوچک باعث رشد و پیشرفت شما می‌شوند.

#### ۶- معقول باشید:

به یاد داشته باشید که نباید برای هر معامله، یک میزان خرج کنید. حسن نیت شما نباید متزلزل شود، اما سرمایه‌تان چرا. اگر معامله‌ی بزرگی به اندازه‌ی ۲۰ تا ۱۰۰ معامله‌ی کوچک عایدی دارد، مسلماً باید روی اولی متمرکز شوید.

#### ۷- از شکست متنفر باشید:

به کسی فرصت ندهید. اجازه ندهید که مشتری بر شما غالب شود. باید پرتکاپو و سلطه‌جو برخورد کنید تا تصور کنند اگر با شما کار نکنند دچار اشتباه شده‌اند. البته حرف‌هایتان فقط نباید در مرحله‌ی ادعا باقی بماند، باید همه‌ی تلاش‌تان را در راضی کردن آن‌ها به کار بگیرید. گرچه معاملات خوب همیشه آن‌هایی است که از دستشان می‌دهیم.

#### ۸- از فروش منصرف شوید:

اگر می‌بینید که مشتری نمی‌خواهد به اندازه‌ی ارزش کار، پول بپردازد، شما قیمت‌تان را پایین نیاورید و به هیچ وجه کوتاه نیایید. صرف‌نظر کنید. همیشه این شانس وجود دارد که دوباره سراغتان بیایند.

#### ۹- بدانید که به آن‌ها کمک می‌کنید:

درست است که نباید همیشه به روی خود بیاورید، ولی مطمئن باشید که آن‌ها بیش از آن مقدار که شما به آن‌ها نیاز دارید، به شما نیاز دارند. خلاقیت رکن اساسی فروشندگی است. اگر خلاق هستید، پنهانش نکنید. آن را به همه نشان دهید و سودش را ببرید.

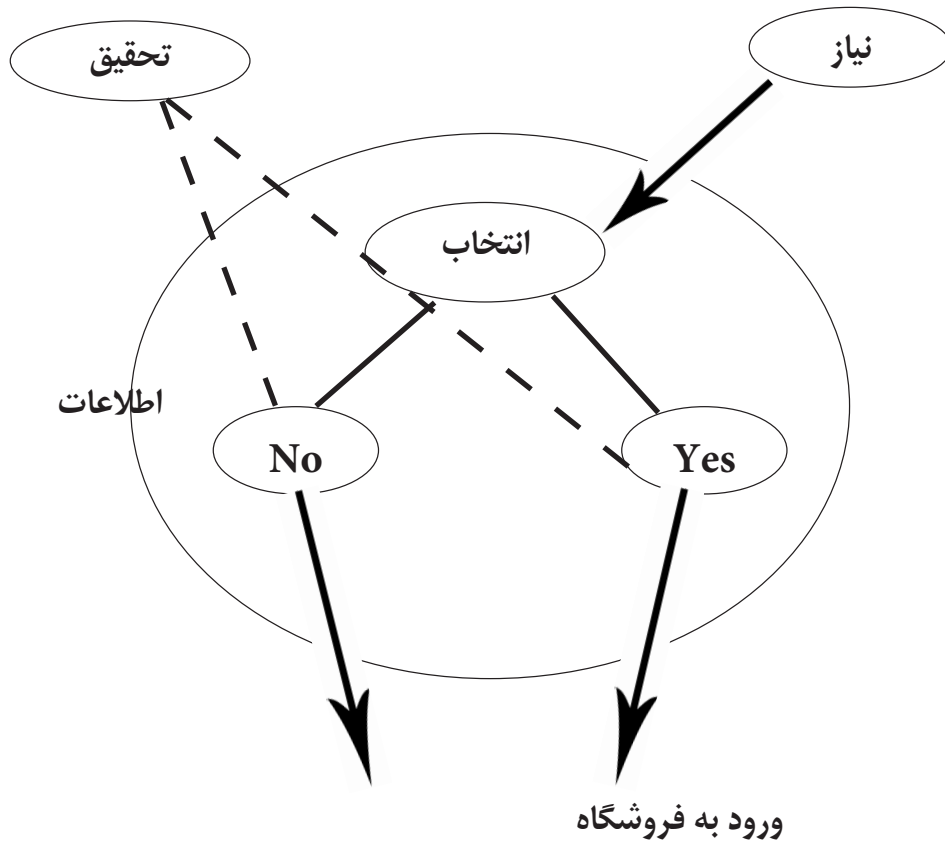
#### ۱۰- کارت‌تان را تفریح در نظر بگیرید:

بدون در نظر گرفتن قرارداد، پول، جلسه و میتینگ، سعی کنید تا می‌توانید از کارت‌تان لذت ببرید. آن را مثل یک ورزش فرض کنید و خود را یکی از بهترین ورزشکاران بدانید. فقط در این صورت است که مشتریان هم به سراغتان می‌آیند و می‌خواهند آن‌ها را هم بازی دهید.

#### ۱۱- کاری که شروع کرده‌اید را به اتمام برسانید:

چه به دنبال کار، چه به دنبال بستن قرارداد، و چه به دنبال همسر دلخواهتان باشید، کاری که شروع کرده‌اید را تمام و کمال انجام داده و به اتمام برسانید.

در جلسه‌ای که در روز سه‌شنبه ۱۳۹۲/۸/۱۴ برگزار شد، به بررسی سناریوهایی که خانم مریم مجاور رضایی تهیه کرده بودند پرداختیم. همچنین چند سناریوی جدید و به‌طور کلی بررسی سیستم از بالا انجام شد. نتایج به صورت زیر است:



#### سناریوها:

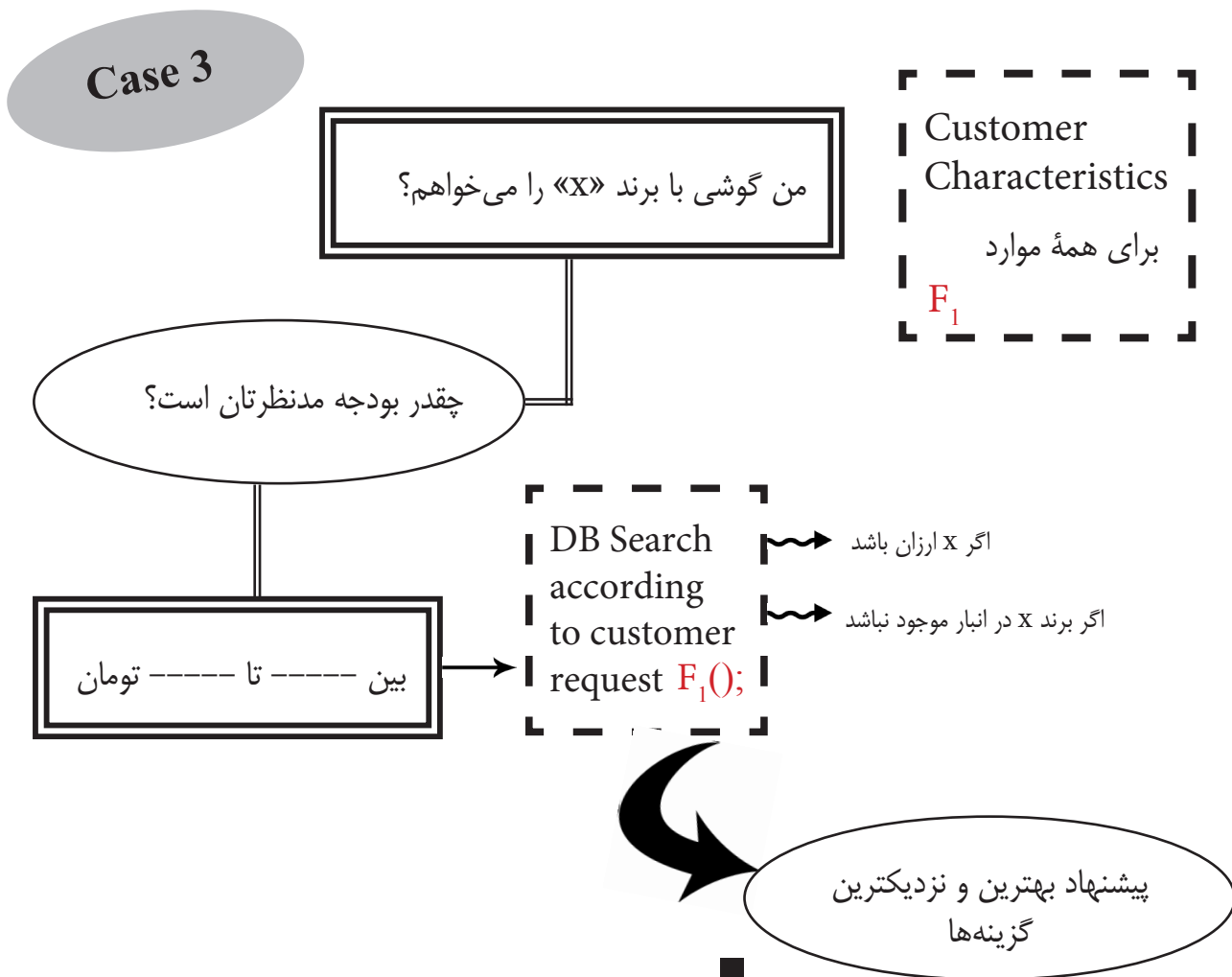
- ۱- مشتری به فروشگاه مراجعه می‌کند. با دانستن اطلاعات گوشی مورد نظر، اقدام به پرسش درباره‌ی قیمت و گارانتی گوشی مورد نظرش می‌کند. فروشنده قیمت و نوع گارانتی را اعلام می‌کند و مشتری در صورت تمایل اقدام به خرید می‌کند.
- ۲- مشتری به فروشگاه مراجعه می‌کند. با دانستن اطلاعات گوشی مورد نظر خود اقدام به پرسش درباره مقایسه بین گوشی مورد نظر و گوشی‌های مشابه (از نظر فنی، قیمت، ظاهر و ...) می‌کند. اینجا فروشنده می‌تواند پیشنهاد بهتری با تفاوت قیمت بالاتری به خریدار بدهد. مشتری در صورت تمایل اقدام به خرید می‌کند.
- ۳- مشتری به فروشگاه مراجعه می‌کند. هیچ اطلاعاتی از نوع و قیمت گوشی ندارد. فروشنده از مبلغ مورد نظر مشتری سوال می‌کند. مشتری یک محدوده‌ی قیمت می‌دهد (فازی). فروشنده اقدام به جستجو و پیشنهاد به مشتری می‌کند. البته در این مورد فروشنده سعی می‌کند با توجه به مزایای مورد نظر مشتری و بررسی انبار (گوشی‌های مانده در انبار) در محدوده‌ی قیمت مشتری آن‌ها را بفروشد.
- ۴- مشتری به فروشگاه مراجعه می‌کند. هیچ اطلاعاتی از نوع و قیمت گوشی ندارد. فروشنده از مبلغ مورد نظر مشتری سؤال می‌کند. مشتری یک محدوده‌ی قیمت می‌دهد. فروشنده اقدام به جستجو و پیشنهاد به مشتری می‌کند. فروشنده می‌تواند گوشی‌های بهتر خود را با قیمت کمی بالاتر ارائه بدهد، زیرا مشتری به احتمال زیاد اطلاعاتی از قیمت گوشی مورد نظر فروشنده ندارد.

۵- مشتری به فروشگاه مراجعه می‌کند. از قبل گوشی خاصی را انتخاب نکرده ولی حدود قیمت چند نوع گوشی را دارد. پس فروشنده متوجه می‌شود که به مشتری هر گوشی با هر قیمتی را نمی‌تواند پیشنهاد بدهد، زیرا ممکن است مشتری از فروشگاه خارج و اقدام به پرس و جو در مورد قیمت و نوع گوشی بکند. پس سعی می‌کند قیمت مناسبی را ارائه بدهد. مشتری در صورت تمایل اقدام به خرید می‌کند.

۶- مشتری به سایت‌های خرید اینترنتی مراجعه می‌کند و اقدام به انتخاب نوع و قیمت محصول پیشنهادی می‌کند. در صورت تمایل اقدام به خرید می‌کند.

۷- مشتری از دوستان خود که مشتریان سابق این فروشگاه بوده‌اند اقدام به پرس و جو می‌کند، در صورتی که سابقه فروشگاه و نوع ارائه محصول همواره مناسب باشد، مشتری‌های سابق خود نوعی تبلیغ برای فروشگاه شده و به مشتری‌های دیگر، فروشگاه را معرفی می‌کنند. در اینجا چون اعتماد بین مشتری و فروشنده به‌وجود آمده، فروشنده معمولاً هر نوع پیشنهادی بدهد مشتری می‌پذیرد (در صورتی که نوع و قیمت از قبل برایش مشخص نشده باشد).

۸- مشتری به فروشگاه مراجعه می‌کند. از قبل از قیمت و نوع گوشی خود با خبر است. فروشنده اقدام به ارائه پیشنهاد بهتری می‌کند تا نظر مشتری را به پیشنهاد خود تغییر دهد. اگر مشتری نظرش را تغییر داد فروشنده می‌تواند حتی پیشنهاد قبلی خود را تغییر داده و مورد سودآورتری را پیشنهاد بدهد زیرا متوجه می‌شود که مشتری در تصمیم خود متزلزل است، در غیر این صورت با پیشنهاد نصب نرم افزار یا گارانتی خاصی سعی در افزایش سود خود می‌کند. اگر مشتری نپذیرفت اقدام به ارائه فروش تجهیزات جانبی گوشی مورد نظر مشتری می‌کند. اگر مشتری باز هم نپذیرفت چاره‌ای جز فروش گوشی مورد نظر مشتری ندارد.



در جلسه‌ای که در روز سه‌شنبه ۱۳۹۲/۸/۲۸ برگزار شد، به بررسی ساختار دیتابیس و نحوه دینامیک بودن آن پرداختیم. آقای مهدی عزیزی که طراحی دیتابیس را به عهده دارد، توضیحاتی داد که به شرح زیر است:

پایگاه داده برنامه از نوع پایگاه داده رابطه‌ای (Relational) است که به صورت پویا (Dynamic) طراحی شده است. در این پایگاه داده موجودیت‌های (Entity) زیر را داریم که هر کدام در جدول‌های مختلف ویژگی‌های (Features) خود را نگهداری می‌کنند. جدول‌هایی که موجودیت‌های اصلی برنامه را تشکیل می‌دهند به شرح زیر هستند:

– customers: جدول نگهداری ویژگی‌های مشتری

– products: جدول نگهداری ویژگی‌های کالا

– factors: جدول نگهداری ویژگی‌های فاکتورهای فروش

هر کدام از این جدول‌ها به چند جدول دیگر متصل هستند که اطلاعات و ویژگی‌های موجودیت را به صورت پویا نگهداری می‌کنند. به عنوان مثال برای جدول کالاها (products) روابط زیر را داریم:

– products: در این جدول، اسامی کالاها به همراه شماره و کد مخصوص نگهداری می‌شود.

– product\_descriptions: در این جدول توضیحاتی درباره کالا داده می‌شود.

– product\_features: در این جدول ویژگی‌هایی برای موجودیت کالا به صورت کلی در نظر گرفته می‌شود. خصوصیت پویا بودن پایگاه

داده در اینجا مشخص می‌شود، چرا که هر تعداد ویژگی که برای کالاها (یا حتی برای یک کالا) مد نظر باشد را می‌توان به راحتی در این جدول افزود. حتی اگر آن ویژگی به گونه‌ای باشد که در زمان طراحی پایگاه داده اصلاً وجود نداشته، می‌توان آن را به راحتی و بدون نیاز به تغییر ساختار پایگاه داده، به برنامه اضافه کرد.

– product\_feature\_descriptions: توضیحات ویژگی‌های جدول بالا، در این جدول می‌آید.

– product\_feature\_variants: مقادیری که هر ویژگی می‌تواند بگیرد در این جدول ذکر می‌شود. این جدول نیز از خصوصیت پویا

بودن پایگاه داده پشتیبانی می‌کند. اگر یک ویژگی، در آینده نیاز باشد تا مقادیر متغیر و متفاوت دیگری را اخذ کند، آن مقادیر را به راحتی به آن ویژگی اضافه می‌کنیم.

– product\_feature\_variant\_descriptions: توضیحات مقادیر جدول بالا نیز در این جدول می‌آید.

– product\_features\_values: این جدول یک جدول کلیدی و محوری است که مقداری را که یک کالای مشخص در یک ویژگی مشخص دارد ذخیره می‌کند.

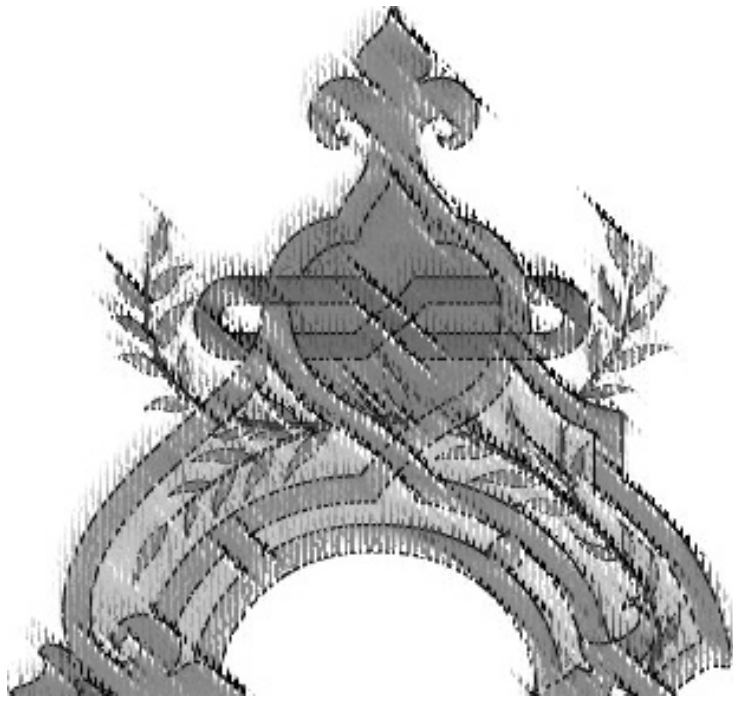
از جدول‌های دیگری که به جدول products متصل هستند می‌توان به جدول‌های زیر اشاره کرد:

– products\_categories: این جدول یک جدول واسطه‌ای بین جدول کالاها و جدول دسته‌بندی کالاها (categories) موجود است. در این جدول مشخص می‌کنیم که هر کالا به کدام دسته از محصولات تعلق دارد.

– companies: اطلاعات شرکت‌هایی که کالاها به آن‌ها تعلق دارند در این جدول نگهداری می‌شود.

– also\_bought\_products: کالاهایی که با یکدیگر خریداری می‌شوند، به همراه تعداد خرید، در این جدول نگهداری می‌شوند.





گزارش سوم

# Third Report



## جلسه هشتم

در جلسه‌ای که در روز چهارشنبه ۱۳۹۲/۹/۲۷ برگزار شد، گروه گرافیک (آقای هروی و آقای اسماعیلی) محیط طراحی شده فروشگاه را نشان دادند که ورژن ابتدایی طراحی بود. بقیه دوستان در این مورد ایده‌هایی مطرح کردند که بصورت زیر است:

۱. ویتترین قرار دادن برای گوشی‌های فروشگاه بطوری که هر برند یک ویتترین جداگانه داشته باشد.

۲. با رفتن به سمت ویتترین‌ها گوشی‌های موجود در برند مربوط به نمایش در بیایند.

۳. با وارد شدن مشتری مشخصاتش در صفحه کناری فروشگاه به نمایش در بیایند.

۴. برای مشتری شکل آدمک بهتری گذاشته شود.

قرار شد در ورژن بعدی این اصلاحیات اعمال شود.

گروه مهندسی نرم افزار (خانم رضوی، خانم بابائی، خانم رضایی و آقای خاندانی) در مورد کلاس‌های پروژه بحث نمودند. نتیجه بصورت زیر است:

کلاس‌های موجود در پروژه:

۱. کلاس مشتری

۲. کلاس محیط

۳. کلاس محصولات

۴. کلاس فروشنده

۵۰

---

## جلسه نهم

در جلسه‌ای که در روز پنج‌شنبه ۱۳۹۲/۹/۲۸ تشکیل شد، گروه مهندسی نرم افزار Attributes و Functions مربوط به کلاس‌ها را بحث نمودند. نتایج بصورت زیر است:

### Customer Class

### کلاس مربوط به مشتری

#### Attributes:

1. Gender
2. Age
3. Appearance
4. Expert knowledge
5. Income Level
6. Single
7. Mood
8. Education
9. Position

10. Entrance Time (of customer)

11. Customer ID

### Functions:

#### 1. Generate Function

اگر خروجی تابع Enter در کلاس Environment، یک باشد این تابع فراخوانی می‌شود. در این تابع ویژگی‌های مربوط به مشتری و مکان مربوط به مشتری و نحوه حرکت او تا رسیدن به فروشنده باید مشخص شود. در نهایت این تابع، تابع بعدی یعنی movement را فراخوانی می‌کند.

#### 2. Movement Function

این تابع توسط تابع قبلی فراخوانی می‌شود. در این تابع دو حالت کلی تعریف می‌شود یکی اینکه مشتری مستقیماً به سمت فروشنده می‌رود و دیگر اینکه گذری در فروشگاه دارد و ویتترین‌ها را مشاهده می‌کند و بعد به سمت فروشنده می‌رود، یا تصمیم می‌گیرد که از سیستم خارج می‌شود. (در یک حالت هم مشتری می‌تواند در حین مشاهده ویتترین از فروشنده سوال بپرسد). خروجی این تابع موقعیت فضایی پایانی مشتری است که

#### 3. Update Function

تابعی است که در هر لحظه وضعیت کلی فروشگاه را به‌روز می‌کند که شامل Time, Space, Position, Dialog board می‌شود.

#### 4. Dialog Function

با توجه به منطق سیستم جمله‌ای را که قرار است به فروشنده بگوید را در Dialog Board بنویسد.

#### 5. Interesting Function

در اینجا مشتری هنگام مشاهده ویتترین به یک یا چند محصول علاقه‌مند می‌شود و باید در این حالت شماره آن محصول را به خاطر بسپارد و پس از آن به سمت فروشنده برود و با او مکالمه کند.

#### 6. Exit Function

در این تابع عملیات خروج مشتری که شامل خالی شدن فضا، کم شدن مشتری‌های سیستم و ... انجام می‌شود.

---

### Environment Class

### کلاس مربوط به محیط

### Attributes:

1. Time

2. Max num of Customers

3. Space (ماتریسی است که فضای مربوط به فروشگاه را تعریف می‌کند مثلاً یک ماتریس ۲۰۰ در ۲۰۰)

4. Shop work time (بازه زمانی که فروشگاه در آن باز است)

## 5. Dialog Board (تابلویی است که مکالمات در آن نمایش داده می‌شود)

### Functions:

#### 1. Enter function

این تابع با استفاده از توزیع‌های احتمالاتی بیان می‌کند که در این لحظه مشتری وارد شود یا خیر. ورودی آن تعداد ماکزیمم مشتری‌ها و خروجی آن صفر یا یک است که اگر یک بود مشتری وارد می‌شود یعنی تابع Generate کلاس مشتری، فراخوانی می‌شود.

#### 2. Update Function

تابعی است که در هر لحظه وضعیت کلی فروشگاه را به‌روز می‌کند که شامل Time, Space, Dialog board می‌شود. با اجرای این تابع، توابع به‌روز رسانی کلیه کلاس‌ها فراخوانی می‌شود.

#### 3. Delay Function

این تابع فاصله زمانی ورود مشتری تا زمان کنونی را محاسبه کرده و اگر از حدی بیشتر شد و اگر فروشنده نیز بیکار بود ( $Busy=0$ ) و اگر توابع interesting () و dialog () مشتری نیز فعال نبود، تابع Dialog فروشنده فعال شود.

#### 4. Terminate Function

در حالتی که زمان پایان کار فروشگاه فرا می‌رسد ورود مشتری‌ها متوقف می‌شود (تابع Enter در Environment غیر فعال می‌شود) و به کار مشتری‌های داخل فروشگاه رسیدگی می‌شود (که البته در اینجا time مشخصی تعیین خواهد شد) و فروشگاه بسته می‌شود.

---

Product Class	کلاس مربوط به محصولات
---------------	-----------------------

### Attributes:

1. price
2. position (مکانی که محصول در آن قرار دارد)
3. Brand
4. Guaranty (y/n)
5. Number (تعداد)
6. Features (شامل مشخصات گوشی)

### Functions:

#### 1. Update

زمانی که یک محصول فروخته می‌شود از دیتابیس حذف می‌شود. (و مقادیر محصولات کلاً آپدیت می‌شود)

**Attributes:**

1. Position
2. Busy (متغیری است دودویی که بیانگر بیکار بودن یا نبودن فروشنده است)
3. Seller ID

**Functions:**

1. Dialog Function

با توجه به منطق سیستم جمله‌ای را که قرار است به مشتری بگوید را در Dialog Board بنویسد.

2. Movement Function

نحوه جابه‌جایی فروشنده و گاهی اوقات اگر بیکار باشد به سمت مشتری که در حال تماشای ویتترین است، می‌رود.

3. Update Function

تابعی است که در هر لحظه وضعیت کلی فروشگاه را بروز می‌کند که شامل Time, Space, Dialog board, می‌شود. با اجرای این تابع توابع به‌روز رسانی کلیه کلاس‌ها فراخوانی می‌شود.

4. Sell Function

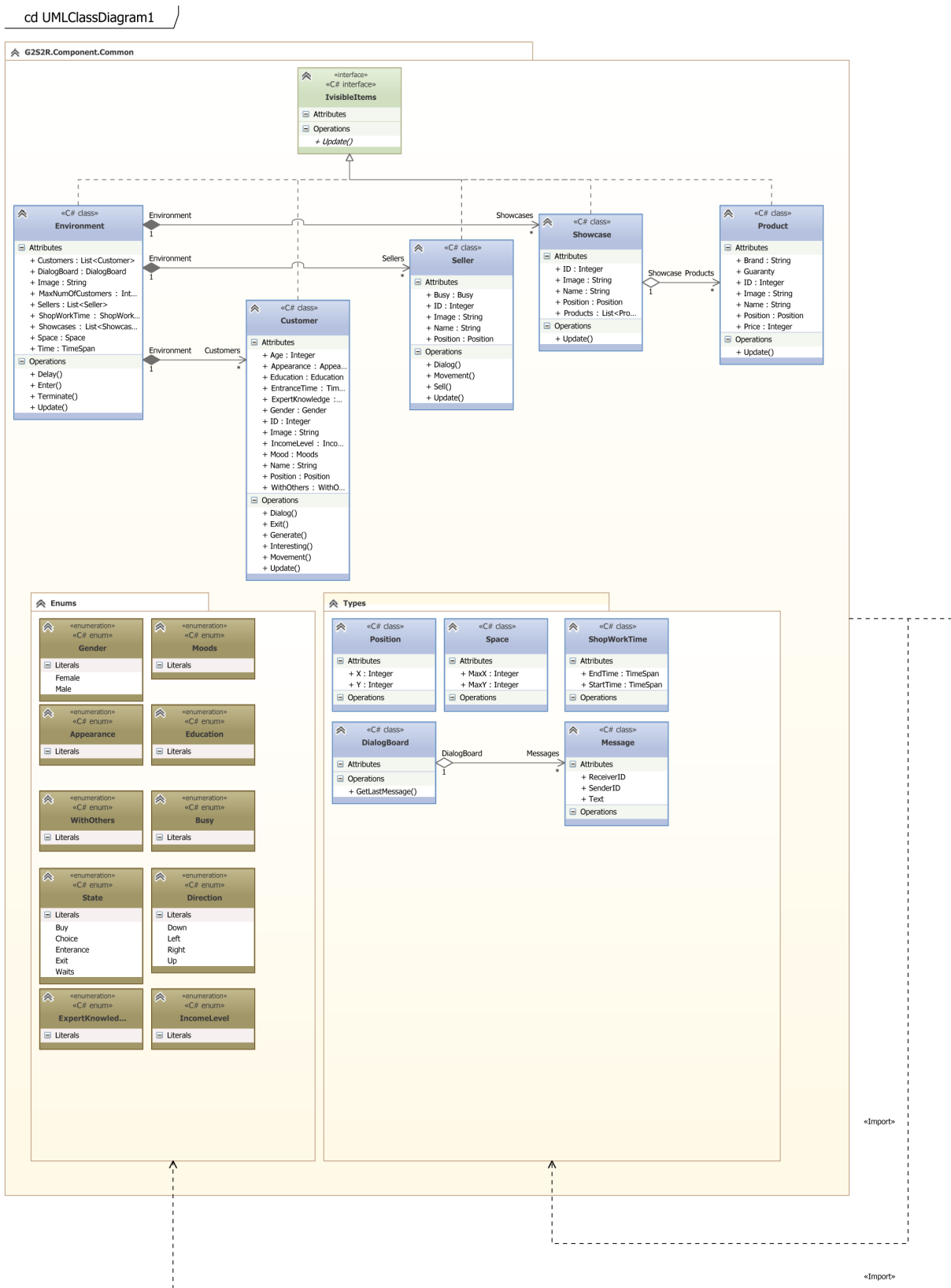
در اینجا گوشی به مشتری تحویل داده می‌شود. بعد از او در مورد وسایل جانبی پرسیده می‌شود و در نهایت قیمت کلی محصولات محاسبه شده و مشتری مبلغ را پرداخت کرده و فاکتور تحویل می‌گیرد.



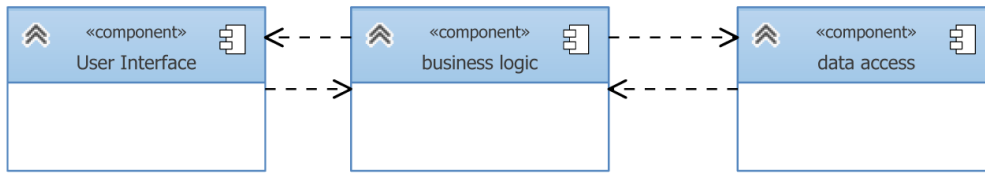
در پایان نیز بهتر است این نکات در نظر گرفته شود:

- فضای فیزیکی و متراژ باید مشخص شود تا افراد داخل و بیرون فروشگاه از هم تمیز داده شوند.
- دخالت دادن ریسک‌پذیری، به عنوان مثال در نظر گرفتن شهرت افراد
- نمایش فروشگاه، افراد و ورود و خروجشان، میزان درآمد و سود در هر مرحله و هر روز
- ویژگی‌های هر فرد قبل از ورود به صورت Random، تنظیم میشود و بعد از خروج نیز در Database ذخیره میشود

در جلسه‌ای که در روز سه‌شنبه ۱۳۹۲/۱۰/۳ تشکیل شد، گروه مهندسی نرم‌افزار در مورد دیاگرام‌های مختلفی که به تحلیل سیستم کمک می‌کند و ساختار کلی سیستم را نشان می‌دهد بحث نمودند که نتایج به صورت زیر است:

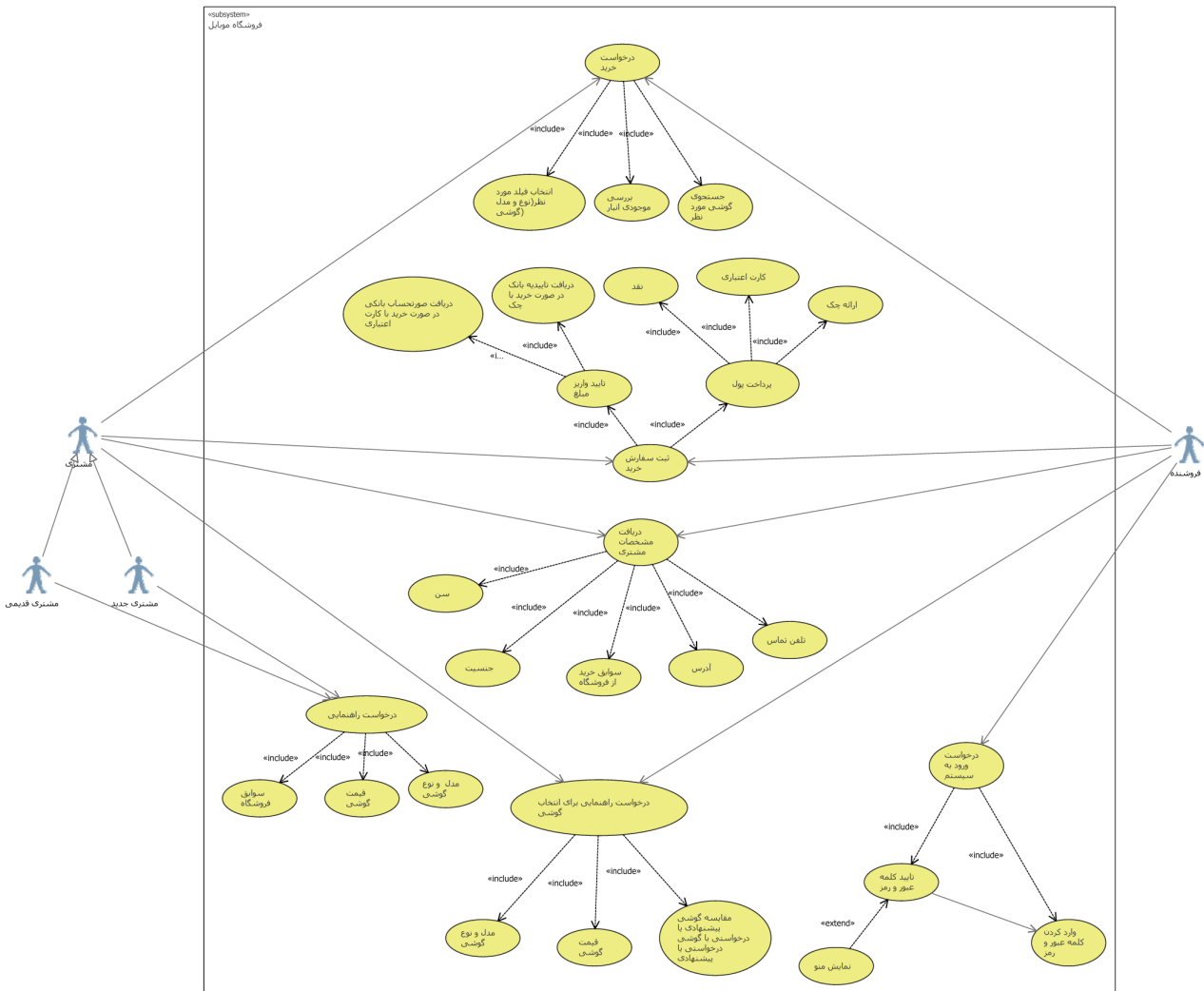


# cmp UMLComponentDiagram1

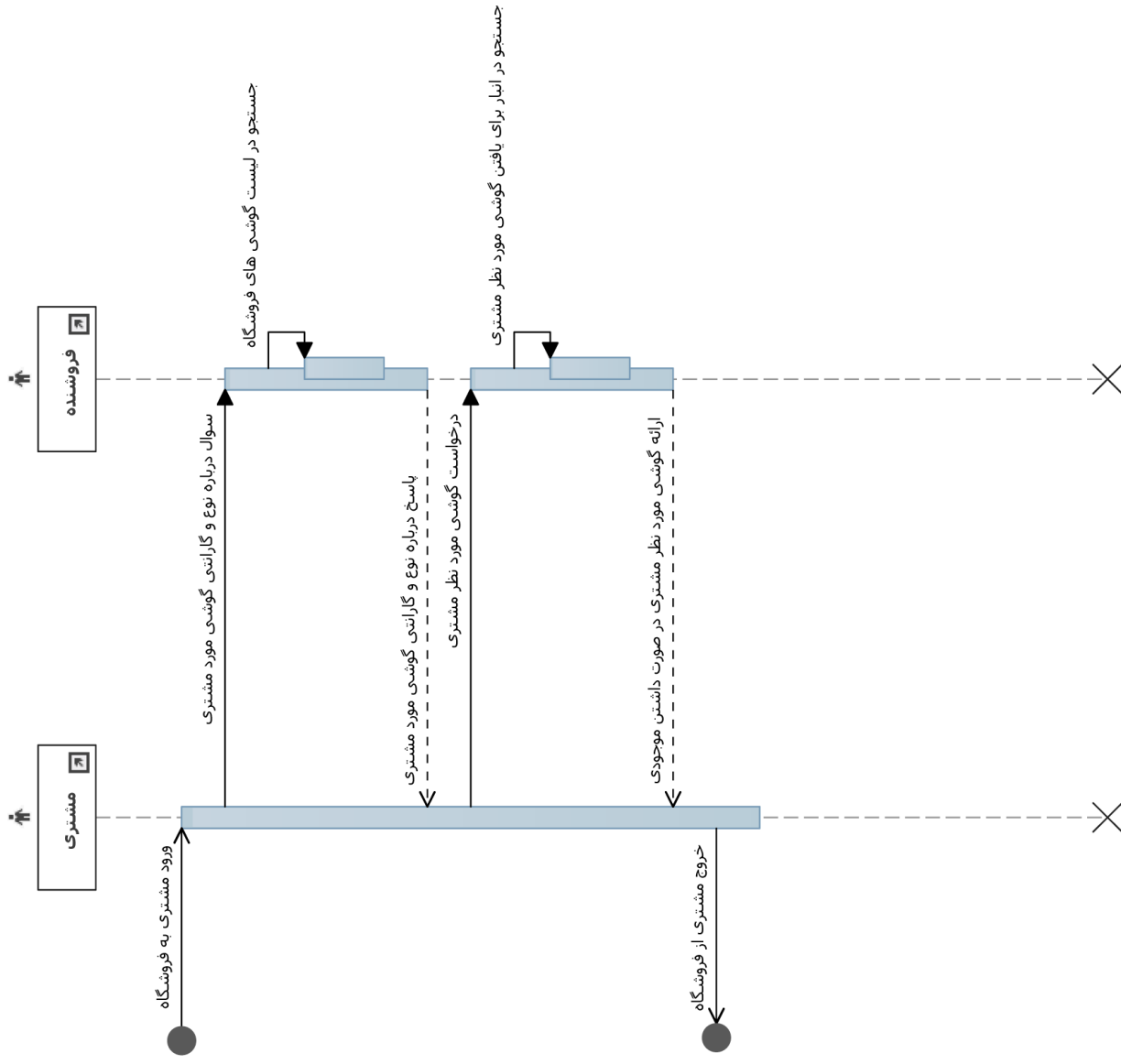


The general Component schema of the system

# uc UMLUseCaseDiagram1

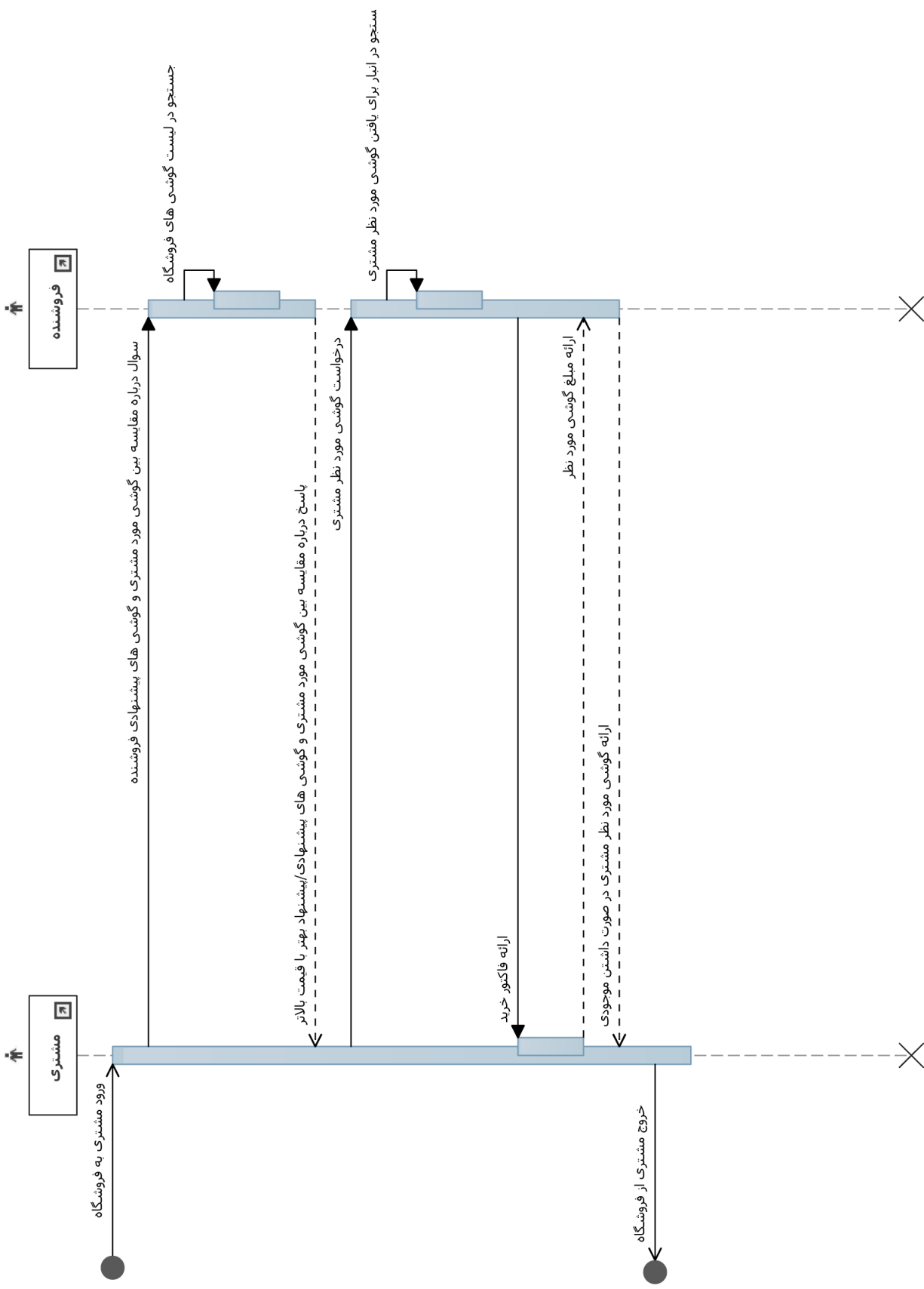


# sd Sequence1

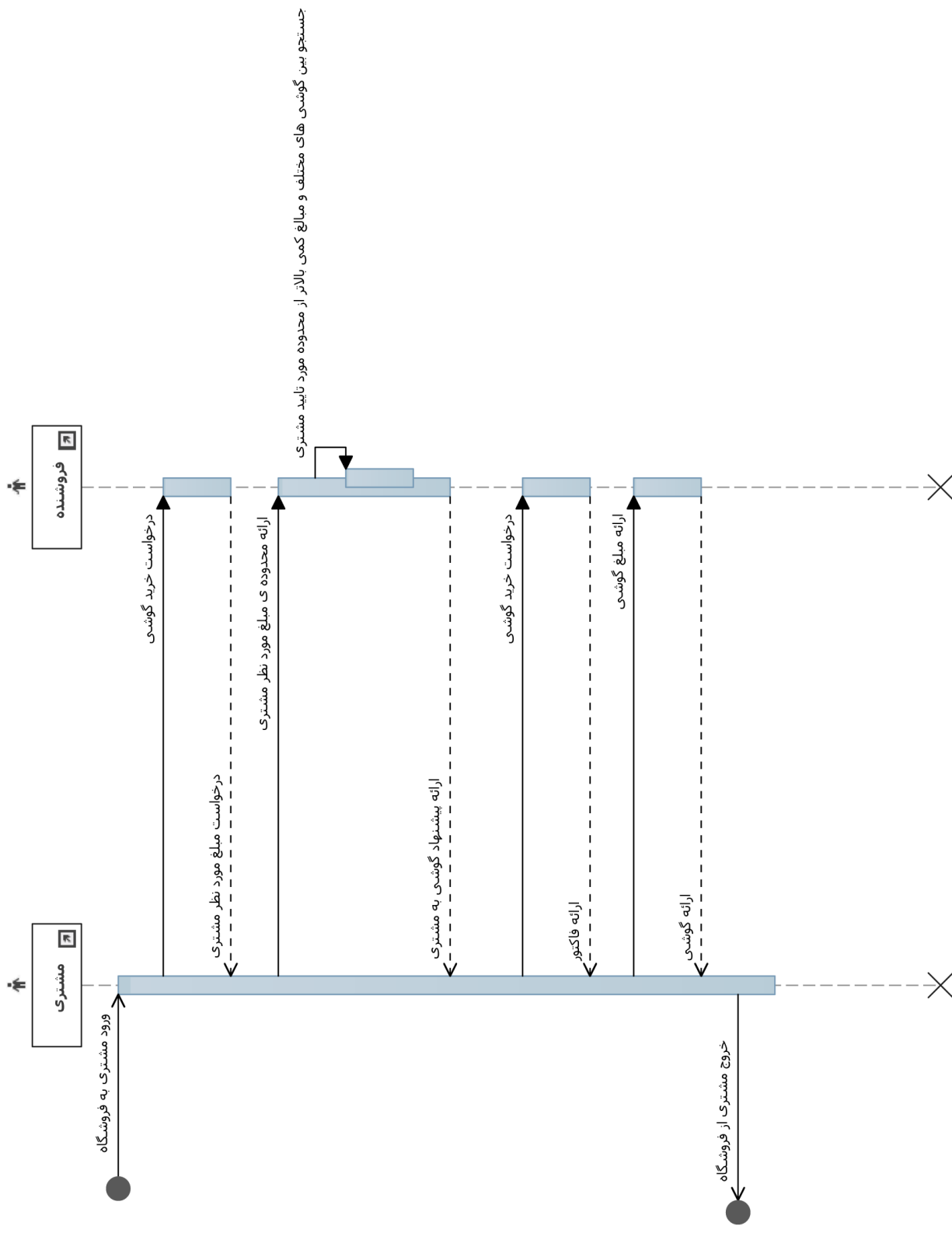




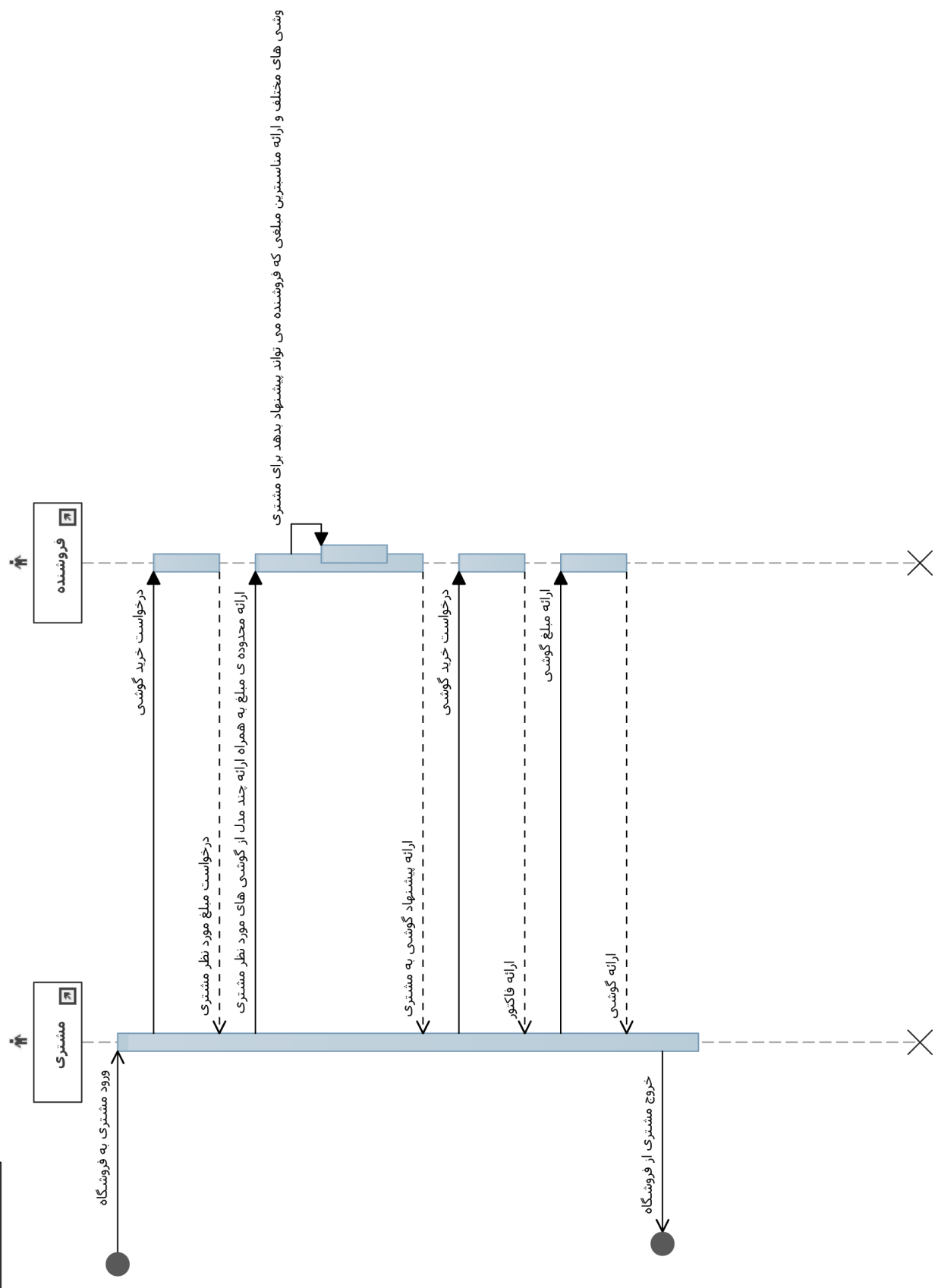
## sd Sequence2



### sd Sequence3



### sd Sequence4



# sd Sequence5

