|  |
| --- |
| C:\Users\Sadra Hakim\AppData\Local\Microsoft\Windows\INetCache\Content.Word\University-of-Isfahan.png  دانشگاه اصفهان  دانشکده فنی و مهندسی  گروه مهندسی کامپیوتر |
| سامانه ساوا |
| سامانه انتخاب واحد و برنامه‌ریزی |
|  |
| پدیدآورندگان:  گروه Runtime Terror  صدرا حکیم  علی گلی  پریسا محمدی  رضا مهتری  صبا هنرمند  استاد راهنما : دکتر بهمن زمانی |
| **آذر 1399** |

|  |
| --- |
|  |

فهرست مطالب

[فصل اول: سند نیازمندی های نرم افزار 4](#_Toc56816742)

[1-1 مقدمه 4](#_Toc56816743)

[1-1-1 هدف 4](#_Toc56816744)

[1-1-2 قلمرو 4](#_Toc56816745)

[1-1-3 تعاریف، سرنام‌ها و کوته نوشت‌ها 5](#_Toc56816746)

[1-1-4 مراجع 5](#_Toc56816747)

[1-1-5 شرح کلی 5](#_Toc56816748)

[1-2 شرح کلی 5](#_Toc56816749)

[1-2-1 چشم انداز محصول 6](#_Toc56816752)

[1-2-2 کارکرد محصول 8](#_Toc56816753)

[1-2-3 مشخصات کاربر 8](#_Toc56816754)

[1-2-4 قیود 9](#_Toc56816755)

[1-2-5 مفروضات و وابستگی ها 9](#_Toc56816756)

[1-3 نیازمندی‌های خاص 9](#_Toc56816757)

[1-3-1 نیازمندی‌های واسط خارجی 9](#_Toc56816758)

[1-3-2 نیازمندی‌های کارکردی 10](#_Toc56816759)

[1-3-3 کارایی 12](#_Toc56816760)

[1-3-4 قیود طراحی 12](#_Toc56816761)

[1-3-5 صفت‌های سیستم نرم‌افزاری 12](#_Toc56816762)

[فصل دوم: مدل دامنه 14](#_Toc56816763)

[1-2 جمع‌آوری اطلاعات دامنه کاربرد 14](#_Toc56816764)

[2-2 طوفان فکری 14](#_Toc56816765)

[3-2 دسته‌بندی مفاهیم طوفان فکری 14](#_Toc56816766)

[4-2 به تصویر کشیدن مدل دامنه 17](#_Toc56816767)

[5-2 مرور مدل دامنه 17](#_Toc56816768)

[فصل 3: طراحی معماری 18](#_Toc56816769)

[1-3 فرایند طراحی معماری 18](#_Toc56816770)

[3-1-1 اهداف طراحی معماری 18](#_Toc56816771)

[3-1-2 تعیین نوع سیستم 18](#_Toc56816772)

[3-1-3 استفاده از سبک‌های معماری 18](#_Toc56816773)

[3-1-4 تعیین واسط‌ها و عملیات زیرسیستم 19](#_Toc56816774)

[2-3 سبک معماری و نمودار بسته 20](#_Toc56816775)

[3-3 اعمال قوانین طراحی نرم‌افزار 20](#_Toc56816776)

[3-3-1 طراحی برای تغییر 20](#_Toc56816777)

[3-3-2 جداسازی دغدغه‌ها 21](#_Toc56816778)

[3-3-3 پنهان‌سازی اطلاعات 21](#_Toc56816779)

[3-3-4 چسبندگی زیاد 21](#_Toc56816780)

[3-3-5 جفت‌شدگی کم 22](#_Toc56816781)

[3-3-6 ساده و احمقانه فرض کن 22](#_Toc56816782)

# فصل اول: سند نیازمندی های نرم افزار

## مقدمه

بدون شک موفقیت یک دانشجو در ثبت یک برنامه درسی ایده‌آل مدیون اطلاعات، دقت و سرعت انالیز در انتخاب وضعیت بهینه است. در این طرح با افزایش فیلترهای اطلاعاتی، سامانه‌ای طراحی شده است که علاوه بر کمک قابل توجه به سیستم آموزش در ارائه بهینه برنامه کلیه دروس، قادر است با دقت بسیار بالا بخش عمده‌ای از وظیفه‌ی انتخاب واحد صحیح و ایده‌آل را برای هر دانشجو، خود به عهده گیرد.

### هدف

برای تولید یک نرم‌افزار، لازم است سندی از نیازمندی‌های نرم افزار را تهیه کنیم تا بتوانیم محصول مناسبی را ایجاد کنیم.

سند نیازمندی‌های نرم‌افزار، یک نقشه از تمام ویژگی‌های محصول نرم‌افزاری را به تمامی اعضای تیم توسعه نشان می‌دهد و کمک می‌کند که اعضای تیم در یک خط فکری قرار بگیرند و همگی در راستای یک هدف مشخصی قدم بردارند. همچنین این سند کمک می‌کنند که اگر شخص جدیدی می‌خواهد عضو تیم شود بتواند در جریان اصلی کار قرار بگیرد و در مدت زمان کوتاهی بتواند نقش موثری در تیم داشته باشد. سند نیازمندی‌های نرم‌افزار (Software Requirements Specification) که به اختصار سند SRS نیز شناخته می‌شود، سندی است که جزئیات بخش‌های مختلف پروژه، امکانات و ویژگی های هر بخش و هر آنچه که نیاز داریم نرم افزار برایمان براورده سازد را توصیف می‌کند. مخاطبان این پروژه نرم‌افزاری دانشجویان، اساتید و کارکنان آموزشی دانشگاه‌هایی می‌باشند که قصد دارند فرایند ثبت نام اصلی خود را ارتقا دهند.

### قلمرو

نام این محصول سامانه انتخاب واحد ارتقا یافته یا به اختصار سامانه ساوا می‌باشد.

این محصول کلیه فرآیند های مربوط به فرایند انتخاب واحد اعم از برنامه ریزی درسی، ارزیابی استاید، عملیات ثبت نام مقدماتی، فرایند اصلی انتخاب واحد و ترمیم را پشتیبانی و انجام می‌دهد. اما سایر موارد آموزشی مثل برگزاری کلاس ها، انتخاب اساتید، عملیات پیشخوان خدمت و ... را انجام نمی‌دهد.

هدف این سامانه بهبود و سهولت در امر انتخاب واحد بوده و از مزایای آن می‌توان به نکات زیر اشاره کرد.

* برنامه ریزی درسی در هر لحظه با استفاده از الگوریتم‌های خاص
* امکان نظر‌سنجی و ارزیابی اساتید
* ایجاد بستری برای تبادل نظر دانشجویان
* امکان ایجاد و ذخیره‌سازی برنامه دلخواه دانشجو و استفاده آن در انتخاب واحد
* ایجاد فرایند ثبت‌نام مقدماتی جهت سهولت در امر برنامه ریزی برای مسئولین آموزشی

### تعاریف، سرنام‌ها و کوته نوشت‌ها

* وب اپلیکیشن: (Web Application) وب اپلیکیشن یا برنامه تحت وب، نرم‌افزاری است که با استفاده از مرورگر وب و از طریق شبکه‌های محلی یا اینترنت مورد استفاده قرار می‌گیرد.
* سرور: سرور یک کامپیوتر قدرتمند به لحاظ سخت‌افزاری است که به واسطه یک سری ملزومات قادر است یک یا چندین سایت را میزبانی نماید.
* API: مخفف واژگان Application Programming Interface است که به صورت تحت‌الفظی می‌توان آن را به «رابط برنامه‌نویسی نرم‌افزار» ترجمه کرد.
* بکاپ (Backup) : بکاپ یا نسخه‌ی پشتیبان به یک نسخه‌ی همانند از اطلاعات ما گفته می‌شود که در جایی دیگر برای اطمینان و امنیت بیشتر ذخیره خواهد شد.

### مراجع

مهندسی نرم افزار شئ گرا – یک متدولوژی چابک یکنواخت. نوشته ی سی کونگ، دیوید . ترجمه بهمن زمانی و افسانه فاطمی . انتشارات دانشگاه اصفهان.

### شرح کلی

این سند کلیه نیازمندی‌های اساسی، روابط بین سامانه با سایر سامانه‌های دانشگاه و همچنین زیر سیستم‌های خود سامانه را به طور تفضیلی توضیح می دهد.

## شرح کلی

این سامانه برای بهبود عملکرد فرآیند انتخاب واحد بوده، از این روی برای سهولت در برنامه‌ریزی، بخش برنامه‌ریزی لحظه‌ای برای دروس تعبیه شده تا دانشجویان به راحتی برنامه دلخواه خود را ساخته و استفاده کنند.

همچنین برای تصمیم‌گیری راحت‌تر برای انتخاب اساتید، بخش نظرسنجی و ارزیابی اساتید اضافه شده است.

کلیه فرآیند ثبت نام مقدماتی و انتخاب واحد و ترمیم نیز همانند سامانه‌های قبلی صورت می‌گیرد که در ادامه کلیه موارد را به طور کامل شرح خواهیم داد.



### چشم انداز محصول

سامانه ساوا دارای دو بخش سایت و وب اپلیکیشن است. جهت بهبود عملکرد در برنامه‌ریزی درسی در پیش از انتخاب واحد وب اپلیکیشنی برای برنامه‌ریزی هفتگی دروس با استفاده از الگوریتم های برنامه ریزی تعبیه شده تا کاربر پس از یک بار گرفتن اطلاعات دروس بتواند بطور افلاین و بدون نیاز به دسترسی به اینترنت برنامه ریزی خود را انجام دهد. با توجه به تغییرات پیشبینی نشده در ارائه دروس کاربر می‌تواند با استفاده از وب اپلیکیشن اطلاعات جدید دروس را در هر لحظه از پایگاه دریافت کند. همچنین وب اپلیکیشن امکان تبادل و نظرسنجی دانشجویان در مورد اطلاعات دروس و اساتید را دارا بوده که این بخش آن نیاز به اینترنت داشته و بصورت آنلاین اجرا می‌شود.

همچنین کاربر عملیات های ثبت نام مقدماتی، انتخاب واحد، ترمیم و دریافت گزارشات آموزشی را در سایت انجام خواهد داد.

#### واسط‌های سیستم

با توجه به نیاز برنامه به اطلاعات دروس سامانه ساوا نیاز به API هایی از پایگاه داده دانشگاه دارد. همچنین پس از انتخاب واحد نتایج انتخاب واحد هر فرد پس از تایید مسئول آموزشی باید در پایگاه داده ثبت شود.

سامانه همچنین باید نتایج ثبت نام مقدماتی را به واسطه پایگاه‌داده، به دانشکده‌های مربوطه جهت اطلاع مسئولین آموزشی بدهد و همچنین تغییرات احتمالی ایجاد شده در برنامه های درسی یا سایر اطلاعات و سیاست های جدید آموزش کل یا مسئول آموزشی دانشکده ها را دریافت نماید.

همچنین برای عملیات اننتخاب واحد و ثبت نام مقدماتی نیاز به سرور‌های قدرتمند برای پردازش وذخیره‌سازی اطلاعات داریم.

#### واسط‌های کاربر

با توجه به اینکه کاربران سامانه چهار دسته می‌باشند، سامانه ساوا دارای چهار نوع سطح دسترسی و واسط کاربری است:

* واسط کاربری دانشجویان: برای دانشجویان دو واسط کاربری در نظر گرفته شده:

1. واسط کاربری وب اپلیکیشن برای برنامه‌ریزی دروس و نظرسنجی دروس و اساتید است که با توجه به ماهیت وب اپلیکشین، روی تمامی سکو ها قابل دسترسی است.
2. واسط کاربری سایت برای عملیات ثبت نام مقدماتی و انتخاب واحد و ترمیم که برای آن نیاز به مرورگر داریم.

* واسط کاربری مهمان: واسط کاربری مهمان، سایت می‌باشد که کاربر توانایی ساختن و استخراج برنامه‌های هفتگی را دارد.
* واسط کاربری مدیر سطح دو: واسط کاربری مدیر سطح دو، سایت می‌باشد که در آن کار‌های مشاهده ی نتایج مربوط به ثبت نام مقدماتی و سایر اطلاعات آموزشی دانشجویان و همچنین دریافت گزارشات و درخواست مجوز‌های دانشجویان برای انتخاب واحد و پاسخگویی به آن ها را انجام می‌دهد.
* واسط کاربری مدیر سطح یک: واسط کاربری مدیر سطح یک نیز سایت می‌باشد که در آن عملیات مربوط به حذف، اضافه یا ویرایش حساب های‌کاربری سایر کاربران و دادن سطح دسترسی به مدیر سطح دو انجام می شود.

#### واسط‌های سخت‌افزاری

با توجه به اینکه سامانه ساوا، یک سامانه نرم‌افزاری بوده نیاز مبرهنی به واسط سخت‌افزاری خاصی ندارد اما با توجه به نیاز سامانه به اینترنت می‌توان کارت شبکه و مودم و همچنین سرور را به عنوان واسط سخت افزاری نام برد.

#### واسط‌های نرم‌افزاری

مرورگر کاربران که قرار است سامانه در آن نمایش داده شود، باید از JavaScript پشتیبانی کند.

همچنین پایگاه داده سامانه، دیگر واسط نرم‌افزاری بوده که از پایگاه دادهMySQLاستفاده می شود.

برای چت و نظر سنجی از کتابخانه‌ی Socket.io استفاده می شود.

#### واسط‌های ارتباطی

سامانه روی سرورهای دانشگاه استقرار یافته و از طریق تارنمای مربوط به سامانه گلستان قابل دسترسی است. (در واقع سامانه یکی از پنل های سامانه گلستان خواهد بود؛ این کار جهت عدم اعمال تغییرات زیاد در سیستم کلی آموزشی دانشگاه صورت گرفته است.)

وب اپلیکیشن سامانه نیز از طریق سایت سامانه قابل دانلود است و برای قابل اجرا روی هر سکویی می باشد.

همچنین ارتباط سامانه با سایر بخش‌ها و سیستم‌های دیگر آموزشی در بخش واسط‌های سیستم به طور کامل توضیح داده شده است.

#### واسط‌های حافظه

سامانه ساوا برای ذخیره و بازیابی سریع‌تر اطلاعات دانشجویان نیاز به حافظه‌ی جانبی SSD دارد. چون سامانه ساوا نیاز به نوشتن و خواندن با سرعت زیاد دارد، این نوع حافظه بسیار مناسب‌تر از حافظه های HDD می باشد.

هر چه حافظه رم زیادتری برای این سیستم در نظر بگیریم، سرعت انجام و پردازش عملیات هایی نظیر انتخاب واحد که عملیات‌های همزمان بسیار بیشتر خواهد بود.

#### واسط‌های عملیات

اطلاعات دروس ارائه شده به واسطه پایگاه داده به وب اپلیکیشن سامانه ما داده شده و همچنین نتایج آن نیز پس از تایید دستی مسئول آموزشی به صورت خودکار در پایگاه داده ذخیره می شود.

اطلاعات ثبت نام مقدماتی نیز به صورت خودکار به مسئولین آموزشی دانشکده‌ها داده شده و بررسی اطلاعات آن ها به صورت دستی انجام شده و دروس ارائه شده برای هر ترم بصورت دستی توسط مدیر سطح یک به پایگاه داده‌ها، داده می‌شود.

با توجه به اینکه سامانه روی سرورهای دانشگاه بوده، بکاپ آن نیز روی همان سرور ها بوده و تا قبل از فرایند انتخاب واحد بصورت هفتگی بکاپ گیری انجام می‌شود. همچنین در بازه زمانی انتخاب واحد‌ها، بکاپ گیری پس از اتمام زمان انتخاب واحد هر ورودی انجام می‌شود.

اطلاعات انتخاب واحد باید برای حداکثر 10 سال روی سرور‌ها باقی بماند.

#### نیازمندی‌های سازگاری با محیط نصب

وب اپلیکیشن سامانه قابلیت اجرا روی هر سیستم عامل و سکویی را دارا بوده و حتی نیاز به نصب نیز ندارد.

سایت سامانه نیز فقط نیاز به مرورگر دارد.

### کارکرد محصول

سامانه ساوا برای بهبود در فرآیند انتخاب واحد طراحی شده که کارهایی نظیر برنامه ریزی درسی با توجه به دروس ارائه شده در هر لحظه، ارتباط با سایر دانشجویان جهت نظرسنجی، بحث و تبادل پیرامون دروس ارائه شده و اساتید ارائه‌دهنده آن، ثبت نام مقدماتی، انتخاب واحد و عملیات ترمیم و دریافت و انتقال گزارشات آموزشی را انجام می‌دهد.

### مشخصات کاربر

سامانه ساوا چهار نوع کاربر را دارا می‌باشد:

* دانشجویان: با توجه به دانشجو بودن داشتن مدرک دیپلم امری ضروری بوده و همچنین آشنا بودن با فرآیند انتخاب واحد، رویه‌های آموزشی موردنیاز و آشنایی با زبان فارسی نیز موردنیاز می‌باشد.
* مهمان: مهمان در این سامانه می‌تواند وارد شود و به برنامه هفتگی برای خود ایجاد و تنظیم کند. همچنین کاربر امکان استخراج کردن این برنامه را دارد. در سامانه ساوا، کاربر مهمان با آی‌پی شناسایی می‌شود.
* مدیر سطح دو: دارای حداقل مدرک لیسانس، داشتن پست آموزشی در دانشگاه و دانشکده مربوطه، آشنایی با کلیه فرآیند ها، قوانین و آیین‌نامه‌های آموزشی دانشگاه و توانایی کار با ابزارهای رایانه‌ای و آشنا به زبان فارسی از جمله نیازمندی‌های این کاربر می‌باشد.
* مدیر سطح یک: دارای حداقل مدرک لیسانس، تجربه کار با سامانه های دیگر، آشنایی ابتدایی به قوانین آموزشی و فرایند انتخاب واحد و همچنین شناخت مسوولین آموزشی و وظایف آن‌ها از جمله نیازمندی های این کاربر می باشد.

### قیود

* سامانه باید دارای سایت و همچنین وب‌اپلیکیشن باشد.
* سایت سامانه باید قابلیت اجرا روی همه‌ی مرورگرهای مشهور را داشته باشد.
* واسط کاربری سامانه باید کاربر پسند بوده و کار با آن راحت باشد.
* سامانه باید حداقل تغییر در سیستم آموزشی دانشگاه را داده و بصورت بخشی از سامانه گلستان پیاده‌سازی شود.
* زمان طراحی این سامانه باید حداکثر 6 ماه و توسعه نرم‌افزاری آن حداکثر 1 سال باشد.
* تیم توسعه نرم‌افزاری باید داری تجربه توسعه سامانه‌های دیگری را داشته باشند.
* تیم توسعه و پیاده‌سازی سامانه باید تابعیت ایران داشته باشند.
* تیم توسعه و پیاده‌سازی سامانه باید تضمین حقوقی برای انجام این کار بدهند.
* تیم توسعه و پیاده‌سازی باید تا حداقل یک سال پشتیبانی نرم‌افزار را انجام داده و پس از آن در صورت عدم تمایل به ادامه پشتیبانی باید کدهای منبع و سایر اطلاعات موردنیاز را در اختیار مشتری قرار دهند.
* بودجه پیاده‌سازی کلی سامانه باید حداکثر 300 میلیون تومان باشد. با توجه به نوسان قیمت ارز، این مبلغ قابل ویرایش می‌باشد.

### مفروضات و وابستگی ها

* به دلیل اینکه سایت سامانه تحت وب می‌باشد نیاز به مرورگر دارد.
* سکویی که وب‌اپلیکیشن ما روی آن ها اجرا می‌شود و هچنین مرورگر کاربر باید از JavaScript پشتیبانی کنند.

## نیازمندی‌های خاص

### نیازمندی‌های واسط خارجی

کلیه توضیحات واسط های خارجی در بالا شرح داده شده و نیازی به تکرار آن‌ها نمی‌باشد.

### نیازمندی‌های کارکردی

1. سیستم باید برای کاربران مهمان، دانشجو، معاون آموزشی و مدیر سطح یک سیستم سطح دسترسی‌های مختلف داشته باشد.
2. سیستم باید دروس پیشنهادی شامل
   1. درس‌های چارت درسی
   2. درس‌های افتاده دانشجو
   3. درس‌هایی که پیش نیاز آنها پاس شده‌اند

برای عملیات ثبت‌نام مقدماتی چند روز قبل از شروع فرآیند به کاربر دانشجو نمایش دهد.

1. سیستم باید اطلاعات موردنیاز جهت نمایش دروس پیشنهادی را از پایگاه‌داده دانشگاه بگیرد.
2. در این سیستم، کاربر معاون آموزشی دانشکده باید بتواند تاریخ و زمان شروع و پایان انجام فرآیند ثبت نام مقدماتی برای کاربران دانشجو را مشخص کند.
   1. عملیات ثبت‌نام مقدماتی باید در زمان مقررشده برای هر گروه امکان‌پذیر باشد.
   2. پس از تمام شدن مهلت ثبت‌نام مقدماتی برای هر کاربر دانشجو باید بخش ثبت‌نام مقدماتی برای او بسته شود و دیگر نمایش داده نشود.
3. سیستم باید محیطی برای کاربر دانشجو فراهم کند تا بتواند دروس موردنظر خود را برای ترم تحصیلی آینده از بین دروس ارائه‌شده انتخاب کند.
4. سیستم باید پس از اتمام فرآیند ثبت‌نام مقدماتی، برنامه‌ی انتخابی توسط دانشجو را در پایگاه داده دانشگاه ثبت کند.
5. سیستم باید پس از اتمام فرآیند ثبت نام مقدماتی، اطلاعات هر گروه از کابران خاص جهت مشاهده‌ی معاون آموزشی، دسته بندی و مرتب سازی شود.
6. سیستم باید دروس ارائه شده در ترم جاری را از پایگاه داده دانشگاه بگیرد.
7. سیستم باید توانایی اضافه کردن دوستان به لیست همراهان خود با اجازه طرف مقابل جهت هماهنگی بهتر به کاربر دانشجو را بدهد.
8. سیستم باید اطلاعاتی پیرامون ارزیابی و نظرسنجی و طرح درس هر استاد را در اختیار دانشجویان قرار بدهد.
9. سیستم باید توانایی ارائه چندین برنامه‌ی پیشنهادی به دانشجو بر اساس پالایه‌های اعمال‌شده توسط دانشجو در زمینه های استاد انتخابی، زمان برگزاری کلاس‌ها، محدوده تعداد واحد درسی مدنظر را داشته باشد.
10. سیستم باید بتواند حداکثر 4 برنامه از برنامه‌های پیشنهادی به انتخاب کاربر دانشجو را با عنوان "برنامه‌های موردعلاقه" ذخیره کند.
11. سیستم باید این امکان را به کاربر معاون آموزشی بدهد که تغییراتی در مسائل آموزشی از جمله رعایت پیش‌نیاز و همیناز و موارد این چنینی را ایجاد کند.
12. در صورت تغییرات در مسائل آموزشی توسط معاون آموزشی که منجر به بروز خطا در زمان انتخاب واحد طبق برنامه‌های موردعلاقه دانشجو می‌شود، دانشجو را مطلع سازد.
13. سیستم باید توانایی اشتراک‌گذاری برنامه نهایی دانشجو با همراهانش را داشته باشد.
14. سیستم باید به طور خودکار تداخل تاریخ و ساعت امتحانات دروس را چک کند و درصورت تداخل، اخطار های لازم را ایجاد کند و گزینه ی مناسب و جایگزین از همان درس را در صورت وجود به کاربر پیشنهاد دهد.
15. سیستم باید مراقب رعایت پیش‌نیاز و همنیاز یک درس باشد و در صورت رعایت نشدن، اخطار های لازم را ایجاد کند.
16. سیستم باید فضایی جهت ارتباط و گفت‌و‌گو با استاد و معاون آموزشی را برای دانشجو فراهم کند.
17. اطلاعات ثبت‌نام مقدماتی و ثبت‌نام اصلی، جهت مشاهده معاون آموزشی باید در قالب فایل Excel در اختیار او قرار گیرد.
18. سیستم باید بتواند برنامه‌های هفتگی ثبت شده توسط کاربر را از پایگاه داده‌ها دریافت کند و در صفحه انتخاب واحد به او نمایش دهد.
19. سیستم بتواند در مواردی که لازم است، به کاربر اخطارهایی را نمایش دهد و همچنین به او اجازه برداشتن درس را ندهد
    1. زمانی که کاربر پیشنیاز دروس را رعایت نکرده است.
    2. زمانی که ظرفیت هر یک از دروس به حد نصاب خود رسید.
    3. زمانی که تعداد واحد های کاربر از سقف تعداد واحدهای مجاز او (که بر اساس معدلش مشخص می شود) بیشتر شد.
20. سیستم باید تمام دروس ارائه شده توسط دانشکده (به جز دروس گذرانده شده توسط دانشجو) را در صفحه انتخاب واحد نمایش دهد.
21. سیستم باید بتواند اطلاعات هر یک از دروس شامل کد درس، نام درس، نام استاد، ساعت برگزاری کلاس، ظرفیت درس، تاریخ امتحان، تعداد دانشجویانی که این درس را اخذ کرده اند، تعداد دانشجویانی که در صف انتظار درس هستند و همچنین نحوه ارائه درس در طول ترم را به کاربر نمایش دهد.
22. سیستم باید زمانی که کاربر تعداد واحد های مجازش (غیر از واحد های ثبت شده در ثبت نام مقدماتی) را اخذ کرد، به او فقط دروسی که در ثبت نام مقدماتی ثبت شده است را نشان دهد.
23. سیستم باید در زمان حذف و اضافه همچنان محدودیت های اخذ دروس مطابق با ثبت نام مقدماتی را اعمال کند.
24. سیستم باید توانایی ایجاد گزارش از وضعیت ثبت نام دانشجو و همچنین برنامه درسی نهایی شده توسط او را داشته باشد.
25. سیستم باید اطلاعات کافی از فرایند ثبت‌نام را در اختیار معاون آموزشی قرار دهد.
26. سیستم باید اطلاعات نهایی شده را به گلستان به طور دقیق و مرتب ارسال کند.
27. سیستم باید به کاربر مهمان توانایی ساخت برنامه هفتگی برای خود بدهد و در صورت نیاز بتواند آن را استخراج کند.

### کارایی

* سامانه ساوا باید توانایی پاسخگویی همزمان به 5000 کاربر را داشته باشد.
* سامانه ساوا باید در هنگام ثبت نام اصلی 99% زمان‌ها در دسترس باشد.
* اگر در فرایند انتخاب واحد، عملیات انتخاب واحد کاربر به مشکلی برخورد، سامانه ساوا باید بتواند اطلاعات ثبت شده توسط کاربر را ذخیره و نگهداری کند.
* سامانه ساوا باید اطلاعات مهم کاربر مانند «رمز عبور» را رمزگذاری کرده و همچنین از امنیت اطلاعات ثبت شده کاربران اطمینان حاصل کند.
* سامانه ساوا باید از واسط کاربری و همچنین تجربه کاربری خوبی برخوردار باشد.
* زمان پاسخگویی به درخواست‌های کاربر در هنگام ثبت‌نام اصلی و ثبت‌نام مقدماتی حداکثر 300 میلی‌ثانیه و در فرایندهای دیگر حداکثر 2 ثانیه می‌باشد.

### قیود طراحی

* سامانه ساوا باید در دستگاه‌های موبایل و تبلت بصورت واکنشگرا نمایش داده شود.
* در سامانه ساوا، معاون آموزشی توانایی تغییر برنامه دانشجو را نداشته باشد. همچنین اگر دانشجو به معاون آموزشی اجازه مشاهده برنامه هفتگی خود را بدهد، معاون آموزشی توانایی ویرایش آن را ندارد.
* مدیر سطح یک علاوه بر توانایی انجام عملیات‌های واسطهای کاربری دیگر، توانایی تعریف، تغییر و حذف حساب‌کاربری دانشجویان و معاون آموزشی را داشته باشد.
* هر دانشجو در زمان ثبت‌نام مقدماتی فقط توانایی انتخاب حداکثر 20 واحد و حداقل 12 واحد را از بین دروس ارائه شده، دارد.
* هر دانشجو در زمان ثبت‌نام اصلی باید حداقل 60% درصد از دروسی که در ثبت‌نام مقدماتی انتخاب کرده است را اخذ نماید.
* پس از اتمام فرایند ثبت‌نام اصلی، سامانه نباید اجازه دسترسی به آن صفحه را به دانشجویان بدهد.

### صفت‌های سیستم نرم‌افزاری

* سرعت
  + سامانه ساوا نرم‌افزاری تحت وب و سریع می‌باشد.
  + زمان پاسخگویی سامانه به درخواست‌های کاربر بسیار سریع است.
* در دسترس بودن
  + سامانه در هنگام ثبت نام اصلی 99% زمان‌ها و همچنین در مرحله ثبت نام مقدماتی، 90% زمان ها در دسترس است.
  + سامانه باید از طریق مرورگرهای کروم، فایرفاکس و IE قابل دسترس باشد.
* امنیت
  + سامانه ساوا اطلاعات کاربران را بصورت امن از طریق شبکه به پایگاه داده‌ها انتقال می‌دهد.
  + این سامانه دارای مجوزهای لازم در این زمینه می‌باشد.
* ظاهر مناسب
  + این سامانه محیطی زیبا و کاربر پسند دارد.
  + استفاده از سامانه نیاز به مهارت خاصی ندارد و کاربران به راحتی می‌توانند نیازهای خود را در محیط سامانه برطرف سازند.
* پشتیبانی مناسب
  + پس از استقرار سامانه، در صورت بروز مشکل، تیم پشتیبانی به مدت دو ترم تحصیلی (1سال) پاسخگو می‌باشد.

# فصل دوم: مدل دامنه

## جمع‌آوری اطلاعات دامنه کاربرد

سامانه‌ی ساوا دارای ۴ سطح دسترسی دانشجو، مهمان، مدیر سطح ۱ و مدیر سطح ۲ می‌باشد. همه‌ی کاربران برای ورود به سیستم نیاز به نام‌کاربری و رمزعبور دارند. همچنین دانشجویان نیز دارای اطلاعات دانشجویی هستند. در این سامانه امکان ثبت‌نام مقدماتی در درس‌ها برای دانشجویان فراهم شده است. در این سامانه موجودیتی به نام استاد تعریف شده است که می‌تواند درس‌هایی را ارائه دهد. همچنین استاد می‌تواند طرح درس خود را در این سامانه قرار دهد و دانشجویان نیز می‌توانند نظرسنجی و طرح درس هر استاد را مشاهده کنند. هر دانشجو می‌تواند در صورت نیاز با معاون آموزشی و اساتید در ارتباط باشد. این سامانه به کاربر توانایی ساختن برنامه‌های مورد علاقه‌ی خود را می‌دهد. دانشجو پس از انتخاب برنامه‌های مورد علاقه‌ی خود، می‌تواند آنها را با همراهان خود به اشتراک بگذارد. در فرایند ثبت‌نام اصلی دانشجو می‌تواند درس‌های ارائه شده را به همراه اطلاعات آنها مشاهده و در آنها ثبت‌نام کند. پس از ثبت‌نام اصلی، سامانه این توانایی را به دانشجو می‌دهد تا او بتواند در فرایند حذف و اضافه درس‌هایی را با توجه به محدودیت گذاشته شده، حذف و یا اضافه کند.

## طوفان فکری

پس از جمع‌آوری اطلاعات، مفاهیم مهم دامنه شناسایی شدند و درنهایت نتایج آن در جدول گام بعدی دسته‌بندی شدند. در شناسایی مفاهیم دامنه، از مفاهیم مربوط به طراحی و پیاده‌سازی چشم‌پوشی شده است.

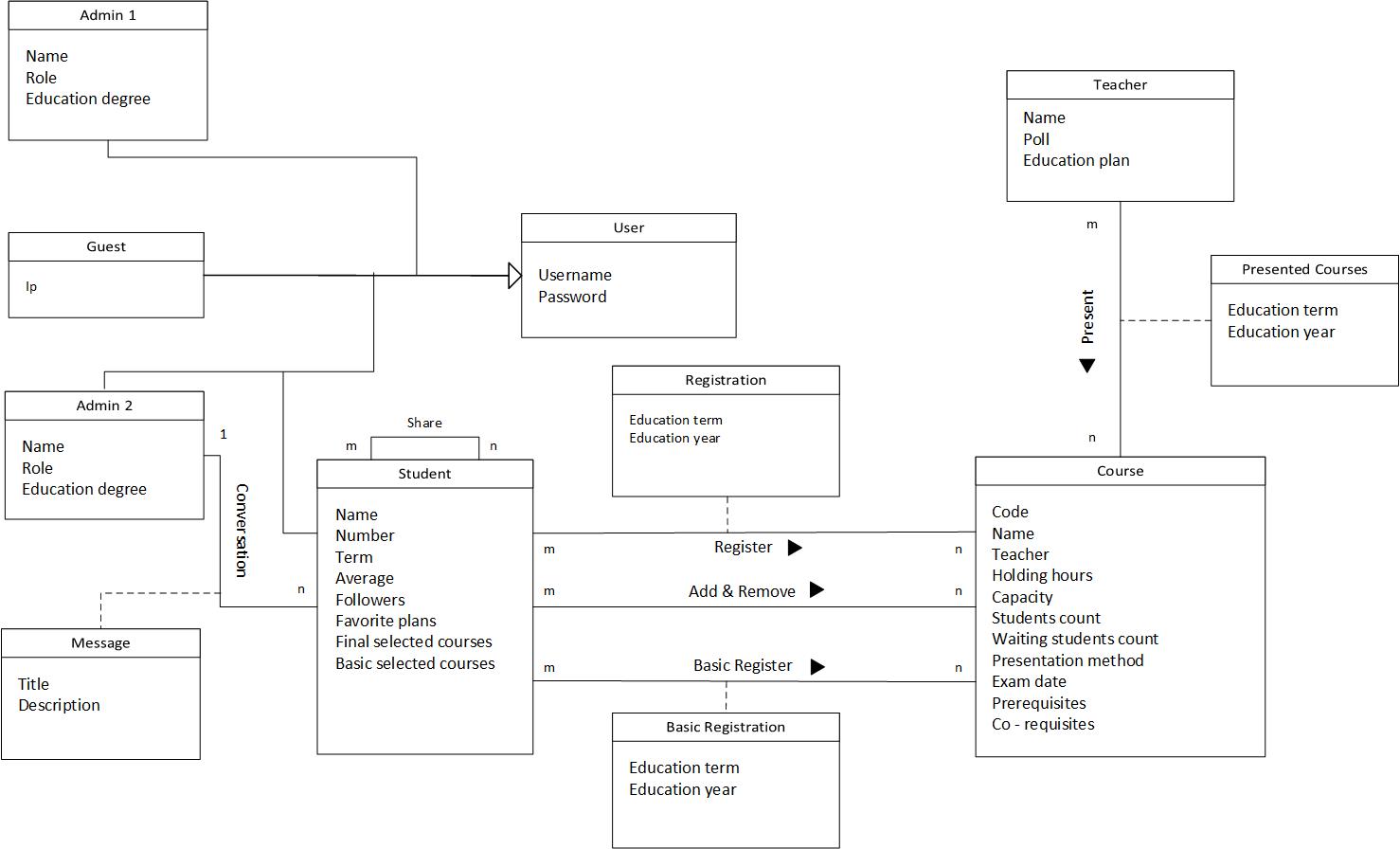
## دسته‌بندی مفاهیم طوفان فکری

در این گام کلاس‌ها، ویژگی‌ها، و روابط دسته‌بندی می‌شوند. محصول نهایی این گام جدول زیر می‌باشد

|  |  |  |
| --- | --- | --- |
| **قانون** | **نتیجه دسته بندی** | **لیست طوفان فکری** |
| 1 (a) | (C) User | کاربران |
| 1 (a) | (C) Guest | مهمان |
| 1 (a) | (C) Student | دانشجو |
| 1 (a) | (C) Admin 1 | مدیر سطح 1 |
| 6 (b) | (A) Name | نام مدیر سطح 1 |
| 6 (b) | (A) Role | سِمت مدیر سطح 1 |
| 6 (b) | (A) Education degree | مدرک تحصیلی مدیر سطح 1 |
| 6 (b) | (A) Ip | آی پی مهمان |
| 1 (a) | (C) Admin 2 | مدیر سطح 2 |
| 6 (b) | (A) Name | نام مدیر سطح 2 |
| 6 (b) | (A) Role | سِمت مدیر سطح 2 |
| 6 (b) | (A) Education degree | مدرک تحصیلی مدیر سطح 2 |
| 9 | (I) ISA (Guest, User) (I) ISA (Student, User) (I) ISA (Admin 2, User)  (I) ISA (Admin 1, User) |  |
| 6 (b) | (A) Username | نام کاربری |
| 6 (b) | (A) Password | رمز عبور |
| 6 (b) | (A) Name | نام دانشجو |
| 6 (b) | (A) Student number | شماره دانشجویی |
| 6 (b) | (A) Student term | ترم دانشجو |
| 6 (b) | (A) Student average | معدل دانشجو |
| 6 (b) | (A) Followers | لیست همراهان |
| 6 (b) | (A) Final selected courses | دروس نهایی انتخاب شده |
| 6 (b) | (A) Basic selected courses | دروس ثبت نام مقدماتی |
| 3 | (AS) Suggest (Course, Student)  (0…n,1) | پیشنهاد بدهد |
| 6 (b) | (A) Favorite Plans | برنامه‌های مورد علاقه |
| 1 (a) | (C) Teacher | استاد |
| 6 (b) | (A) Teacher poll | نظرسنجی هر استاد |
| 6 (b) | (A) Teacher education plan | طرح درس هر استاد |
| 6 (b) | (A) Teacher name | نام استاد |
| 3 | (AS) Present (Teacher, Course)  (m, n) | ارائه بدهد |
| 1 (C) | (AC) Presented Courses (present) | درس‌های ارائه شده |
| 6 (b) | (A) Education term | نیمسال تحصیلی |
| 6 (b) | (A) Education year | سال تحصیلی |
| 1 (a) | (C) Course | درس |
| 6 (b) | (A) Course code | کد درس |
| 6 (b) | (A) Course name | نام درس |
| 6 (b) | (A) Course teacher | نام استاد |
| 6 (b) | (A) Course holding hours | ساعت برگزاری کلاس |
| 6 (b) | (A) Course capacity | ظرفیت درس |
| 6 (b) | (A) Course students count | تعداد دانشجویان ثبت نام شده |
| 6 (b) | (A) Course waiting students | تعداد دانشجویان در صف انتظار |
| 6 (b) | (A) Course presentation method | نحوه ارائه دروس |
| 6 (b) | (A) Course exam date | تاریخ امتحان |
| 6 (b) | (A) Prerequisites | پیش‌نیازهای درس |
| 6 (b) | (A) co - requisite | همنیاز بودن |
| 3 | (AS) Share (Student, Student)  (m, n) | اشتراک‌گذاری |
| 3 | (AS) Conversation (Student, Admin 2) | ارتباط و گفتگو کردن |
| 1 (c) | (AC) Message | پیام |
| 6 (b) | (A) Message title | عنوان پیام |
| 6 (b) | (A) Message description | متن پیام |
| 3 | (AS) Register (Student, Course)  (m, n) | ثبت‌نام اصلی |
| 1 (c) | (AC) Registration | ثبت‌نام |
| 6 (b) | (A) Education term | نیمسال تحصیلی |
| 6 (b) | (A) Education year | سال تحصیلی |
| 3 | (AS) Basic Register (Student, Course)  (m, n) | عملیات ثبت‌نام مقدماتی |
| 1 (c) | (AC) Basic Registration | ثبت‌نام مقدماتی |
| 6 (b) | (A) Education term | نیمسال تحصیلی |
| 6 (b) | (A) Education year | سال تحصیلی |
| 3 | (AS) Add-remove (Student, Course)  (m, n) | حذف و اضافه |

## به تصویر کشیدن مدل دامنه

در این گام با استفاده از اطلاعات به دست آمده از جدول بالا می‌توان کلاس‌ها صفت‌ها و روابط را شناسایی و درنهایت تصویرسازی کرد.



## مرور مدل دامنه

همان‌طور که مشاهده می‌شود مدل دامنه تنها شامل کلاس‌های دامنه می‌باشد و کلاس‌های طراحی و پیاده‌سازی در آن وارد نشده‌اند.

# فصل 3: طراحی معماری

## فرایند طراحی معماری

### اهداف طراحی معماری

* سامانه ساوا خدماتی را فراهم می‌کند و کنشگرها (کاربران) از آن خدمات بهره می‌برند.
* سامانه نیاز به الگوریتم‌های سنگین و زیاد نیز ندارد.
* سامانه باید به تمام درخواست‌های کاربران پاسخ مناسب بدهد.
* این سیستم نیاز به تعامل با کاربر دارد.
* به دلیل نیاز به ایجاد تغییرات احتمالی در داده‌های ورودی به پایگاه‌داده، این امر بسیار مهم است که به ازای تغییر داده‎ها، نیاز به تغییر در سیستم پایگاه‌داده نباشد.
* به دلیل نیاز به پشتیبانی سیستم و نیاز به ایجاد تغییرات احتمالی در سیستم باید زیر سیستم‌ها به صورت زیرسیستم‌های مجزا و با وابستگی کم طراحی شوند.
* برای اطمینان از صحت و درستی اطلاعات دریافت شده از کاربر قبل از ثبت در پایگاه داده، نیاز به صحت‌سنجی اطلاعات می‌باشد.
* سامانه ساوا برای کنترل سطح‌دسترسی کاربران نیاز به یک واحد جهت کنترل این امر دارد.

### تعیین نوع سیستم

تعامل بین سامانه ساوا و کنشگر (کاربران) برای انجام فرایند انتخاب واحد، از یک دنباله تقریبا ثابت از درخواست‌هایی نظیر برنامه‌ریزی، عملیات ثبت‌نام مقدماتی، عملیات ثبت‌نام اصلی و ترمیم و در مقابل، پاسخ‌های سیستم تشکیل می‌شود. این سامانه باید به تمام درخواست‌های کنشگرها پاسخ بدهد. این سیستم در هر نشستی که برقرار می‌کند تنها می‌تواند با یک کنشگر در ارتباط باشد. همچنین موجودیت‌های خارجی این سامانه، انسان‌ها هستند که همان کنشگرهای سیستم محسوب می‌شوند. تعامل کاربر ما از ابتدایی‌ترین مرحله که برنامه‌ریزی می‌باشد، شروع شده و تا عملیات ترمیم که آخرین مرحله است، ادامه می‌یابد. کاربر ما با سامانه، نوعی رابطه مشتری-خادم از خود نشان می‌دهد؛ زیرا کاربر ما عملیات انتخاب واحد را از سامانه درخواست کرده و سامانه خدمات خواسته شده را به کاربر ارائه می‌دهد. فرایند کلی سامانه بصورت گام‌به‌گام می‌باشد و از این روی، پیشرفت فرایند انتخاب واحد برای کاربر کاملا مشهود است.

### استفاده از سبک‌های معماری

با توجه به اینکه نوع سیستم سامانه ما تعاملی بوده، بهترین و معمول‌ترین سبک معماری برای این نوع سامانه، معماری سه لایه می‌باشد که با ویژگی‌های سامانه ما همخوانی دارد. این سامانه از لایه‌های واسط گرافیکی، منطق که خود شامل دو زیرسیستم کنترلگر و انتخاب واحد و لایه داده که شامل زیرسیستم پایگاه‌داده می‌باشد، تشکیل شده است. هر کدام از این لایه‌ها فقط می‌توانند به لایه‌های پایین‌تر از خود درخواست ارسال کنند.

### تعیین واسط‌ها و عملیات زیرسیستم

عملیات زیرسیستم‌ها به شرح زیر می‌باشند:

1. لایه Presentation: این لایه، ورودی‌های موردنیاز را از کاربر گرفته و به لایه ی Logic انتقال می‌دهد. این لایه وظیفه نمایش تمام اطلاعات پردازش شده در لایه Logic را بر عهده دارد. این لایه نیز برنامه‌های پیشنهادی به دانشجو را بر اساس ورودی‌های دریافت شده از کاربر، پردازش و نمایش می‌‌دهد.
2. لایه Logic: این لایه وظیفه اجرای دستورات، پردازش اطلاعات و تمام تصمیم‌گیری‌های منطقی را بر عهده دارد. این لایه نیز خود شامل دو زیرسیستم دیگر می‌باشد:
   1. Controller: مسئول برخورد با رویدادهای مربوط به یک موردکاربرد مشخص می‌باشد.
   2. Business: این لایه مسئول پردازش تراکنش‌های کسب‌وکار که با موارد‌کاربرد نشان داده شده‌اند، می باشد. اشیای کسب‌وکار در این زیرسیستم وجود دارند.
3. لایه Data: در این لایه، اطلاعات در پایگاه‌داده ذخیره و از آن بازیابی می‌شوند که این اطلاعات برای پردازش، به لایه Logic ارسال می‌شوند و نتایج آن، جهت نمایش به کاربر به لایه Presentation برگشت داده می‌شوند.

واسط‌ زیرسیستم‌ها نیز به شرح زیر می‌باشند:

* لایه Presentation به کنترلگر نیاز دارد؛ زیرا در این لایه درخواست‌هایی از طرف کنشگر (کاربر) ایجاد می‌شود که نیاز است این درخواست ها به لایه کنترلگر جهت مدیریت و نظارت، فرستاده بشوند.
* لایه کنترلگر به زیرسیستم Business نیاز دارد؛ زیرا درخواست‌های کاربران به زیرسیستم کنترلگر فرستاده می‌شوند و در نتیجه ممکن است زیرسیستم کنترلگر، به زیرسیستم Business که کلاس‌های اصلی برنامه در آن وجود دارند، نیاز داشته باشد.
* همچنین لایه کنترلگر به زیرسیستم Database نیز نیاز دارد؛ زیرا ممکن است درخواست‌های کاربران نیاز به ارتباط مستقیم با پایگاه داده داشته باشند.
* لایه Business به زیرسیستم Database نیاز دارد؛ زیرا در مواقعی نیاز است که اطلاعات مربوط به کلاس‌های اصلی برنامه با توجه به درخواست کاربران، در پایگاه‌داده ذخیره شوند.

## سبک معماری و نمودار بسته

## اعمال قوانین طراحی نرم‌افزار

در عمل مشکلات بسیاری مربوط به قسمت طراحی رایج می‌باشندکه این مشکلات بر کارایی و کیفیت نرم‌افزار تاثیر منفی داشته و هزینه‌های نگهداری نرم‌افزار را به شدت افزایش می‌دهد. یک از راه‌حل‌ها برای رفع این مشکلات، پیروی از قوانین طراحی نرم‌افزار می‌باشد. در ادامه به بررسی این قوانین و اعمال آن در در معماری سیستم ساوا می‌پردازیم .

### طراحی برای تغییر

رویداد‌های زیادی می‌توانند باعث تغییر در سیستم ما بشوند. برخی از این رویدادها عبارتند از:

* تغییراتی که ممکن است در محیط کسب و کار نرم‌افزار ایجاد شوند و نیازمندی‌های نرم‌افزار را تغییر دهند. به طور مثال اگر که مشتری تصمیم به حذف و یا تغییر در کسب‌وکار مربوط به ارائه برنامه پیشنهادی به دانشجو بکند، نیازمندی‌های سیستم دچار تغییرات زیادی می‌شوند.
* در طول کار سیستم ممکن است مشکلاتی ایجاد شود که برای رفع آن‌ها لازم است که تغییراتی در سیستم نرم‌افزاری صورت گیرد.
* تغییراتی که ممکن است در پایگاه‌داده‌ی مرتبط با سیستم ایجاد شوند، باعث تغییر در سیستم نرم‌افزاری می‌شوند.
* تغییراتی که در آیین‎نامه‌ها و سیاست‌های آموزش دانشگاه‌ها صورت می‌گیرد باعث می‌شوند که نیازمندی‌های سیستم ما تغییر کنند.
* به دلیل پیشرفت روز افزون فناوری و نیاز به تغییر در سیستم، جهت کارامد بودن سامانه در طول زمان، لازم است تغییراتی در فناوری سامانه ایجاد شود.
* برای ارتقای کارایی، قابلیت اطمینان بودن، بهره وری، امنیت و کاربر پسند بودن لازم است تغییراتی در سیستم اعمال شود.
* تغییراتی که ممکن است در برنامه‌ریزی پروژه، زمان تحویل و بودجه‌بندی آن انجام شود باعث تغییر در سیستم و نیازمندی‌های آن می‌شود.

این رویدادهای قابل پیش‌بینی لازم است در طراحی معماری نرم‌افزار در نظر گرفته شوند و این طراحی، سازوکارهایی را برای تطابق با این تغییرات شامل گردد. برای پاسخ به رویدادهای قابل پیش‌بینیِ سیستم تعاملیِ ساوا، ما معماری چندلایه را انتخاب نمودیم که در آن کل سیستم را به چند زیرسیستم دیگر تبدیل نموده و وظایف را بین آنها تقسیم نمودیم. برای مثال اگر نیاز به اعمال تغییراتی در قسمت ظاهر سیستم داشتیم می‌توانیم بدون هیچ نگرانی و نیاز به تغییر در بقیه اجزای سیستم این کار را در لایه‌ی Presentation انجام دهیم و دیگر نیازی به تغییر در ساختار پایگاه‌داده و ... نباشد. همچنین اگر نیاز به تغییر الگوریتمی در قسمت کسب‌وکار و یا پایگاه‌داده داشتیم نیز می‌توانیم تغییرات را بدون نیاز به تغییر بر روی کل سیستم فقط بر روی لایه‌ی مورد نظر اعمال کنیم .

### جداسازی دغدغه‌ها

در طراحی مولفه‌های مستقل نرم‌افزاری، جداسازی دغدغه‌ها به ما می‌گوید که هر مولفه باید بر یک جنبه از موضوع عملکرد نرم‌افزار تمرکز داشته باشد به همین دلیل ما سیستم را به سه لایه‌ی مختلف تقسیم نمودیم که در آن مولفه‌ی Presentation باید بر نمایش اطلاعات به کاربر و دریافت درخواست‌های کاربر برای تغییر اطلاعات، مولفه‌ی مربوط به داده باید بر ذخیره و بازیابی اطلاعات و مولفه Logic نیز برای کنترل درخواست‌های دریافت شده از لایه باید بر مسئولیت‌های خاص کاری خود متمرکز باشد.

### پنهان‌سازی اطلاعات

برای کاهش اثرات تغییرات از قبل پیش‌بینی شده باید جزئیات پیاده‌سازی یک بدنه نرم‌افزاری از سایر قسمت‌های سیستم نرم‌افزاری پنهان شود که این کار با اختصاصی کردن داده‌های یک کلاس و ثابت نگه داشتن ثابت‌های آن کلاس صورت می‌گیرد. در همین راستا در طراحی سیستم باید جزئیات پیاده‌سازی زیر سیستم‌ها از دید کل سیستم مخفی باشد که این امر با معماری چندلایه و در نظر گرفتن زیر سیستم‌ها به صورت لایه‌های جدا از یکدیگر و اختصاصی کردن داده‌های هر لایه صورت می‌گیرد.

### چسبندگی زیاد

چسبندگی میزان ارتباط این توابع با ماموریت اصلی آن زیرسیستم را اندازه‌گیری می‌کند. طبق قانون چسبندگی بالا باید توابع هر زیر سیستم به گونه‌ای باشد که تمام آنها بیشترین درجه‌ی ارتباط با مسئولیت اصلی زیرسیستم را داشته باشند. ما در سامانه ساوا، معماری را به صورت چندلایه در نظر گرفته و توابع را در لایه‌ی مخصوص به خود قرار دادیم. عملکرد توابع در هر زیرسیستم در راستای برآورده ساختن مسئولیت اصلی آن زیر سیستم می باشد. برای مثال هر کدام از توابع از لایه‌یPresentation ، عملکرد‌های متفاوتی دارند ولی عملکرد آنها در راستای تحقق مسئولیت این لایه یعنی تعامل با کاربر و نمایش اطلاعات به کاربر و دریافت درخواست‌های کاربر برای تغییر اطلاعات می‌باشد.

### جفت‌شدگی کم

واضح است که در یک سیستم نرم‌افزاری، زیر سیستم‌ها با یکدیگر در ارتباط هستند که بهتر است در معماری طراحی نرم‌افزار میزان این اثرگذاری اندازه‌گیری شود و در زمان اجرا میزان تاثیر هر زیرسیستم بر سایر زیرسیستم‌ها مشخص شود که این کار با توجه به وابستگی‌ها و عملیات بین زیرسیستم‌ها صورت می‌گیرد. یکی از قوانینی که در طراحی نرم‌افزار ها باید در نظر گرفت، قانون جفت شدگی کم می‌باشد که این قانون تعیین می‌کند که هنگام تغییر یک زیرسیستم، لازم است چند زیر‌سیستم دیگر تغییر کند. استفاده از این قانون در طراحی معماری، موجب کاهش اثرات زمان اجرا و تاثیر تغییر در هر زیرسیستم بر زیرسیستم‌های دیگر است. در معماری چندلایه‌ای که ما برای سامانه ساوا انتخاب نمودیم، لایه‌ها جفت‌شدگی کمی دارند و هر لایه عملیات‌های خود را به طور جداگانه انجام داده و نتایج خود را در قالب خروجی به بقیه زیرسیستم ها می‌دهد و عملیات‌های هر زیر سیستم کمترین وابستگی به عملیات‌های دیگر زیرسیستم ها دارد. برای مثال توابع داخل لایه پایگاه‌داده برای عملکرد خود، نیاز مستقیمی به عملیات‌های داخل لایه‌ی Logic ندارند و فقط طبق واسطی که بین این دو لایه است، کار خود را انجام می دهد.

### ساده و احمقانه فرض کن

در طراحی معماری یک نرم‌افزار بهتر است بعضی مواقع از اشیای نادان استفاده کنیم. بهتر است در معماری نرم‌افزار، در بعضی مواقع یک زیرسیستم کارش را بر اساس یک معیار بدون طرح هیچ سؤالی انجام دهد در اینجا زیرسیستم یک شی نادان است. در طراحی شی‌گرا، طراحی اشیای نادان همان قانون "ساده و احمقانه فرض کن" هست. با اعمال این قانون می‌توانیم طراحی‌های ساده و قابل‌فهمی را داشته باشم. به همین دلیل ما در طراحی چندلایه‌ی خود ما سعی بر این داشتیم که اشیاء را به‌صورت احمق و ساده در نظر بگیریم. برای مثال برای اجرای توابع مختلف بر اساس دستورات دریافتی از کاربر، لایه Logic را به دو زیرسیستمController وBusiness تقسیم نموده و زیر کلاس‌ها ساده‌تر تعریف کردیم که در زیرسیستم Controller، کنترل‌کننده‌های مختلفی را قرار دادیم که هرکدام بر اساس اطلاعات دریافتی از لایه‌ی Presentation ، توابعی که در زیرسیستمBusiness باید اجرا شوند را فراخوانی می‌کند و ما با این کار، شی‌ءها را احمق‌تر تعریف کردیم.