



دانشگاه آزاد اسلامی واحد تهران مرکزی

دانشکده فنی و مهندسی

مجتمع آیت الله هاشمی رفسنجانی (ره)

گروه آموزشی مهندسی کامپیوتر

گزارش پروژه‌ی کارشناسی

عنوان:

پیاپاده‌سازی سامانه رزرو و ثبت زمان‌بندی‌های دفاع پایان‌نامه‌ها

استاد راهنما:

جناب آقای دکتر محسن جهانشاهی

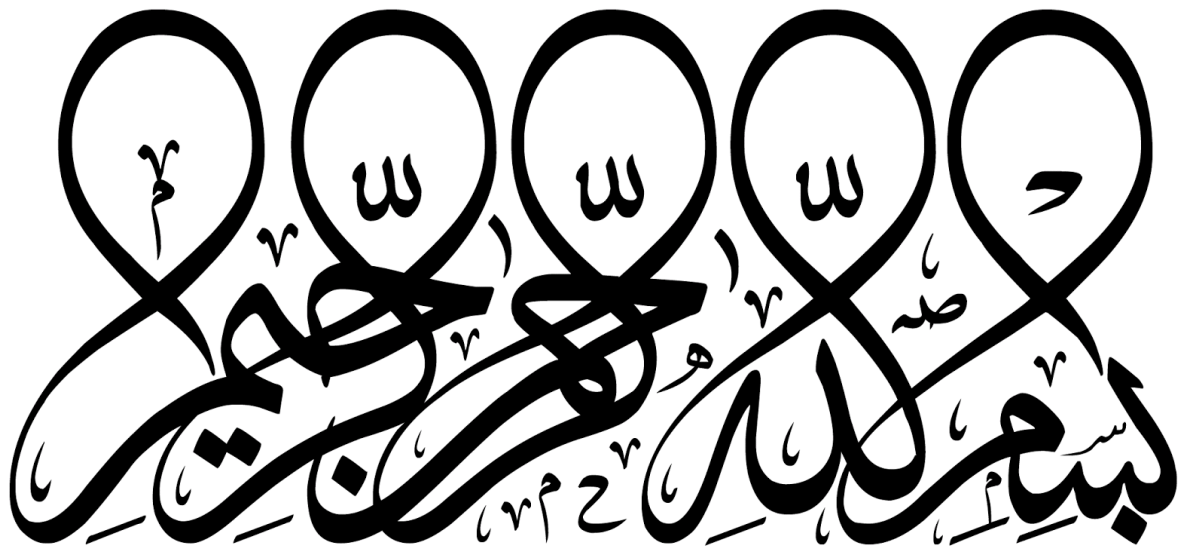
نگارش:

سید اشکان طراوتی

دانشجوی کارشناسی مهندسی کامپیوتر گرایش نرم‌افزار

شماره دانشجویی: ۹۳۰۲۲۵۸۲۰

نیمسال دوم ۹۷-۱۳۹۶



قدردانی

از استاد گرانقدر جناب آقای دکتر جهانشاهی هم به عنوان استاد راهنما و هم به عنوان مدیریت گروه مهندسی کامپیوتر بسیار سپاسگزارم که همواره در چالش‌های گوناگون رهنما و آموزگار بنده بوده اند و بدون راهنمایی‌های ایشان پیشبرد این پروژه بسیار مشکل می‌نمود.

همچنین از اساتید گرامی جناب آقای دکتر وزیری بابت راهنمایی‌ها و تعلیم بی‌دریغ ایشان در طراحی پایگاه داده‌ها، جناب آقای دکتر نصیری بابت راهنمایی‌های ایشان در طراحی و معماری نرم‌افزار، جناب آقای مهندس علامه بابت تدریس رهیافت جدید شبکه، جناب آقای مهندس تناوش بابت تلاش‌های ایشان در آشنایی ما با همروندی و مشکلات مربوط به آن، جناب آقای مهندس عابدسلیمی بابت تدریس بی‌نقص ایشان در رابطه با سیستم عامل لینوکس که عصای دست بنده در این پروژه بود، جناب آقای دکتر هاشمی طباطبائی بابت تذکرات و نکات مهمی که ایشان در مبحث مهندسی نرم‌افزار به بنده و هم‌شاگردی‌ها دادند و در نهایت سرکار خانم دکتر اصغری که در کنار تعلیم همیشه راهنما و مشوق همگی ما بوده اند .

همچنین جا دارد از دوستانی چون آقای مهندس شالباف‌زاده، آقای مهندس ایرانشاهی، خانم مهندس دشتی و خانم مهندس قمی که در روند این پروژه بارها به بنده کمک کردند نیز تشکر و قدردانی نمایم.

از پدر و مادر عزیزم سپاسگزارم که در این مدت مشغله‌ی بنده را تحمل کردند.

فهرست مطالب

۳	۱	مقدمه و معرفی سیستم
۳	۱,۱	شناخت سیستم و صورت مسئله
۳	۱,۲	ویژگی‌های سیستم
۴	۲	مروری بر مفاهیم و شاکله‌ی علمی سیستم
۴	۲,۱	چرخه‌ی حیات توسعه‌ی نرم‌افزار
۱۰	۲,۲	نرم‌افزار کاربردی
۱۱	۲,۳	سامانه‌ی اطلاعاتی
۱۱	۲,۴	لایه‌ی کاربرد شبکه و وب
۱۳	۲,۵	معماری client/server
۱۸	۲,۶	نرم‌افزار تحت وب
۱۹	۲,۷	معماری نرم‌افزار و الگوی معماری
۲۱	۲,۸	معماری MVC و تفکیک دغدغه‌ها
۲۱	۲,۹	فریم‌ورک
۲۱	۲,۱۰	مدلسازی نرم‌افزار
۲۲	۲,۱۱	مستندهای متداول پروژه‌های نرم‌افزاری
۲۲	۲,۱۲	اصل DRY در توسعه‌ی نرم‌افزار
۲۳	۲,۱۳	حمله‌ی CSRF
۲۴	۲,۱۴	برنامه‌نویسی شیء‌گرا
۲۵	۲,۱۵	نگاشت داده‌ها (ORM و Data Mapper)
۲۷	۳	معرفی تکنولوژی‌های به کار رفته در توسعه‌ی سیستم
۲۷	۳,۱	ابزار کنترل نسخه‌ی کد منبع گیت
۲۷	۳,۲	سیستم عامل گنو/لینوکس
۲۸	۳,۳	زبان پایتون
۳۲	۳,۴	زبان جاوا اسکریپت
۳۶	۳,۵	فریم‌ورک جنگو
۳۷	۳,۶	فریم‌ورک بوت‌استرپ
۳۸	۳,۷	تکنولوژی AJAX
۴۰	۳,۸	پایگاه داده Sqlite
۴۰	۴	مستندات فنی پروژه
۴۰	۴,۱	مدلسازی سیستم
۴۴	۴,۲	تشریح اجزای سیستم
۴۴	۴,۳	ساختار پروژه‌های جنگو
۵۱	۴,۴	نمونه کدهای سمت سرور
۵۵	۴,۵	نمونه کدهای سمت کلاینت
۶۲	۴,۶	روش راه‌اندازی سامانه در محیط توسعه
۶۴	۵	راهنمای کاربری سامانه
۶۴	۵,۱	راهنمای بخش جدول زمان‌بندی
۶۶	۵,۲	راهنمای بخش دانشجو
۶۹	۵,۳	راهنمای بخش مدیریت
۷۲	۶	منابع و مراجع

فهرست اشکال

۴	تصویر ۱ توصیف بصری چرخه‌ی حیات توسعه‌ی نرم‌افزار
۳۹	تصویر ۲ نمود تصویری عملکرد AJAX
۴۱	تصویر ۳ نمودار موارد کاربرد (Use Case)
۴۲	تصویر ۴ نمودار کلاس
۴۳	تصویر ۵ دیاگرام موجودیت‌بستگی
۴۴	تصویر ۶ نمودار موجودیت‌بستگی جداول فریم‌ورک
۴۵	تصویر ۷ ساختار دایرکتوری محیط توسعه یا استقرار
۴۶	تصویر ۸ ساختار دایرکتوری اصلی پروژه
۴۷	تصویر ۹ ساختار پوشه‌ی مربوط به پیکربندی
۵۱	تصویر ۱۰ ساختار یک app
۶۴	تصویر ۱۱ صفحه‌ی اصلی سامانه
۶۵	تصویر ۱۲ جدول زمانبندی جلسات
۶۵	تصویر ۱۳ اعلامیه‌ی دفاع
۶۶	تصویر ۱۴ چاپ اعلامیه
۶۶	تصویر ۱۵ صفحه‌ی ورود
۶۷	تصویر ۱۶ داشبورد دانشجو
۶۷	تصویر ۱۷ داشبورد دانشجو - پیگیری درخواست
۶۸	تصویر ۱۸ صفحه‌ی زمان‌ها قابل رزرو
۶۸	تصویر ۱۹ لغو رزرو
۶۹	تصویر ۲۰ داشبورد مدیریت
۶۹	تصویر ۲۱ رویدادهای اخیر
۷۰	تصویر ۲۲ مدیریت کاربران
۷۰	تصویر ۲۳ ویرایش کاربر
۷۱	تصویر ۲۴ گروه‌ها (نقش‌ها)
۷۱	تصویر ۲۵ مدیریت دانشجویان

۱ مقدمه و معرفی سیستم

۱,۱ شناخت سیستم و صورت مسئله

این نرم‌افزار بر بستر وب به دانشجویان تحصیلات تکمیلی گروه‌های آموزشی و دفتر پژوهش دانشکده اجازه می‌دهد تا در تعامل با یکدیگر به آسانی اتاق دفاعیه را برای جلسه‌ی دفاع و حضور اساتید راهنما، مشاور و داور رزرو و هماهنگ نمایند.

کاربران اصلی و عمده‌ی این سامانه دانشجویان و پرسنل پژوهش می‌باشند.

۱,۲ ویژگی‌های سیستم

مطابق با موضوع پروژه و شناخت ارائه‌شده، این مستند به تشریح مراحل استحصال «سامانه رزرو و ثبت زمان‌بندی‌های دفاع پایان‌نامه‌ها» می‌پردازد. این محصول یک نرم‌افزار کاربردی از نوع سامانه‌ی اطلاعاتی تحت وب می‌باشد که با استفاده از تکنولوژی‌های مدرن توسعه‌ی وب به کمک زبان‌های برنامه‌نویسی پایتون و جاوااسکریپت بر پایه‌ی فریم‌ورک‌های جنگو (یک فریم‌ورک سمت سرور مبتنی بر الگوی معماری MVC) و بوت‌استرپ پیاده‌سازی شده و هدف آن تسهیل امور برنامه‌ریزی و تخصیص اتاق‌های دفاع برای برگزاری جلسات دفاع دانشجویان مقاطع تحصیلات تکمیلی می‌باشد. این مستند ترکیبی از چند نوع مستند متداول نرم‌افزار است و شامل مدلسازی و قطعه‌کد می‌باشد.

با توجه به موارد فوق ویژگی‌های زیر برای این سامانه مطرح خواهد بود:

- نگهداری‌پذیری
- انعطاف و قابلیت اضافه کردن امکانات بیشتر
- استقرار آسان
- توزیع‌پذیری
- مقیاس‌پذیری نه‌چندان دشوار
- ساختار ساده و قابل مطالعه
- استفاده از تکنولوژی‌های مدرن و پایدار
- دسترسی آسان بر بستر وب
- تجربه‌ی کاربری مناسب
- سازگار با توسعه‌ی چابک
- استفاده از امکانات مدرن مرورگرهای وب و واکنش‌گرایی

۲ مروری بر مفاهیم و شاکله‌ی علمی سیستم

در این قسمت به طور اجمالی به معرفی و بررسی زیرساخت‌های علمی سامانه می‌پردازیم. در حقیقت این قسمت بیان می‌دارد که از کدام آموزه‌های مهندسی کامپیوتر در اجرای این پروژه استفاده شده است.

۲.۱ چرخه‌ی حیات توسعه‌ی نرم‌افزار

این مبحث در مهندسی نرم‌افزار و رشته‌های مرتبط با موضوع سیستم‌های اطلاعاتی مورد استفاده قرار می‌گیرد و موضوع آن تشریح فرآیندهای مرتبط با برنامه‌ریزی و هدف گذاری، تحلیل، تولید، آزمایش، استقرار و نگهداری سیستم‌های نرم‌افزاری است که اغلب در یک چرخه مانند آنچه در دیاگرام مشاهده می‌شود تداوم و تکرار پیدا می‌کنند. این چرخه همچنین می‌تواند در برگیرنده فرایندهای مربوط به تأمین سخت افزار یا ترکیبی از سخت افزارها و نرم افزارها نیز دانسته شود.



تصویر ۱ توصیف بصری چرخه‌ی حیات توسعه‌ی نرم‌افزار

کلیه پروژه های نرم افزاری الزامی است دارای یک رهیافت چرخه حیات باشند:

- مرحله UR - بیان نیازهای کاربر User Requirements
- مرحله SR - بیان نیازهای نرم افزار Software Requirements
- مرحله AD - طراحی معماری Architectural Design
- مرحله DD - طراحی تفصیلی و تولید برنامه Detailed Design
- مرحله TR - انتقال و واگذاری نرم افزار برای بهره برداری Transfer of the software
- مرحله OM - بهره برداری و نگهداری Operations & Maintenance

چهار مرحله اول با یک بازبینی که بوسیله نشانه "R/R" نمایش داده شده است خاتمه می یابند (به عنوان مثال UR/R بازبینی نیازهای کاربر است). خواه پروژه توسط کارکنان داخلی و یا از طریق شرکتها و پیمانکاران صنعت نرم افزار انجام شود، این مراحل با توجه به اندازه، کاربرد (مثلا علمی، اداری، بلادرنگ، دسته ای)، سخت افزار، سیستم عامل یا زبان برنامه نویسی استفاده شده انجام می پذیرند، هر چند که هر یک از این عوامل رهیافت تولید، شیوه ومحتوی اقلام تحویل دادنی را تحت تاثیر قرار می دهند.

تعیین نیازهای کاربر - مرحله UR

مرحله UR مرحله تعریف مسئله در یک پروژه نرم افزاری است.

- دامنه و وسعت سیستم باید مشخص گردد.
- نیازهای کاربر باید تعیین گردد. این امر می تواند بوسیله مصاحبه، بازدید و بررسی و یا ساخت نمونه های اولیه صورت پذیرد.
- خواسته های مشخص کاربر باید تعیین شده و در سند نیازهای کاربر (URD) نوشته شوند.

پیش از تکمیل بازبینی نیازهای کاربر (UR/R) طرح مدیریت پروژه نرم افزار حاوی رئوس مطالب کل پروژه باید بوسیله تولید کننده نرم افزار تهیه شود. این طرح باید شامل برآورد هزینه پروژه باشد. همچنین طرحهای تفصیلی تر برای مرحله تعیین نیازهای نرم افزار نیز باید تهیه شوند.

تعیین نیازهای نرم افزار - مرحله SR

- مرحله SR مرحله تحلیل پروژه نرم افزاری است.
- یک بخش ضروری فعالیت تحلیل، ساخت "الگویی" است که بیانگر آن باشد که نرم افزار چه کاری را باید انجام دهد، نه اینکه چگونه باید آن را انجام دهد. در این حالت ممکن است ساخت نمونه های اولیه به منظور روشن نمودن نیازهای نرم افزار ضروری باشد.

تحويل دادنی اصلی در این مرحله سند نیازهای نرم افزار (SRD) است. SRD همیشه باید برای هر پروژه نرم افزاری تولید گردد. اصطلاحات پیاده سازی می بایست از SRD حذف شود. این سند باید رسماً "توسط کاربر، مهندسین نرم افزار و سخت افزار کامپیوتر و مدیران ذیربط در حین بازبینی نیازهای نرم افزار (SR/R) مورد بررسی قرار گیرد.

در طول مرحله SR در قسمت طرح مدیریت پروژه نرم افزار، رئیس مطالب باقیمانده پروژه باید بروز آورده شود. طرح باید شامل برآورد مجموع هزینه های پروژه باشد. همچنین طرحهای تفصیلی تر برای مرحله طراحی معماری نیز باید تهیه شود.

طراحی معماری - مرحله AD

هدف مرحله طراحی معماری (AD) تعیین ساختار نرم افزار است.

- الگوی ساخته شده در مرحله SR نقطه شروع این مرحله است. این الگو با تخصیص کارکردها به مولفه های نرم افزار و تعیین گردش اطلاعات و عملیات بین آنها به طرح معماری نرم افزار تغییر می یابد.
- در این مرحله ممکن است طرح چندین بار تکرار شود. مشکلات تکنیکی و یا قسمتهای بحرانی و حساس طرح باید مشخص گردد. ممکن است نمونه سازی بخشهای حساس نرم افزار جهت تایید فرضیات طرح اصلی ضروری باشد. در این مرحله طرحهای جانشین نیز می توان در نظر گرفت که نهایتاً یکی از آنها باید انتخاب شود.

عنصر تحويل دادنی که خروجی اصلی این مرحله به شمار می آید، سند طراحی معماری (ADD) است. ADD همواره باید برای هر پروژه نرم افزاری تولید گردد. ADD باید رسماً "بوسیله مهندسین نرم افزار و سخت افزار کامپیوتر، کاربران و مدیران ذیربط در حین بازبینی طرح معماری (AD/R) مورد بررسی قرار گیرد.

- در طول مرحله AD طرح مدیریت پروژه نرم افزار حاوی رؤس مطالب باقیمانده باید تهیه شود. این طرح باید شامل برآورد هزینه پروژه (حداکثر ۱۰٪ خطا بسیار مطلوب است) باشد. همچنین طرحهای جزئی تر برای مرحله طراحی تفصیلی (DD) نیز می باید تهیه شود.

طراحی تفصیلی و تولید - مرحله DD

- هدف این مرحله طراحی تفصیلی نرم افزار، برنامه نویسی، مستندسازی و آزمون آن است.
- سند طراحی تفصیلی (DDD: Detailed Design Document) و راهنمای کاربر نرم افزار (SUM: Software User Manual) همزمان با برنامه نویسی و آزمون آن تولید می شود. در آغاز DDD و SUM بخشهای متناظر با سطوح فوقانی سیستم را در بر می گیرند. در پایان این مرحله، مستندات تکمیل می شوند و به همراه برنامه ها عناصر تحویل دادنی این مرحله را تشکیل می دهند.
- در طول این مرحله فعالیت های آزمون واحد، آزمون یکپارچگی و آزمون سیستم بر طبق طرحهای مصوب تایید شده در مراحل SR و AD اجرا می گردند. به موازات این آزمون ها می بایست کیفیت نرم افزار نیز مورد سنجش قرار گیرد.

سه عنصر تحویل دادنی (برنامه، DDD و SUM) که مستلزم بازبینی های مرحله ای در طول مرحله DD می باشند باید رسماً توسط مهندسین نرم افزار و مدیریت ذیربط در حین بازبینی طرح تفصیلی (DD/R) مورد بررسی قرار گیرد. در پایان فرآیند بازبینی، نرم افزار جهت آزمونهای پذیرش موقت آماده است.

انتقال - مرحله TR

هدف این مرحله تایید آن است که نرم افزار کلیه خواسته های کاربر را که در سند نیازهای کاربر (URD) مطرح و تنظیم گردیده اند، برآورد می نماید. این امر بوسیله نصب نرم افزار و اجرای آزمون پذیرش انجام می شود.

زمانی که نرم افزار جهت ارائه قابلیت های مورد نیاز نمایش داده می شود، نرم افزار می تواند موقتاً "مورد پذیرش قرار گرفته و بهره برداری از آن آغاز شود."

سند انتقال نرم افزار (STD) باید به منظور مستند ساختن انتقال نرم افزار به گروه بهره برداری در حین مرحله TR تنظیم شود.

بهره برداری و نگهداری - مرحله OM

زمانی که نرم افزار آماده بهره برداری می گردد، می بایست به منظور حصول اطمینان از آنکه کلیه خواسته های تعیین شده در سند نیازهای کاربر را برآورده می سازد به دقت مورد بررسی قرار گیرد. برخی از خواسته ها به مدت زمانی جهت مشخص شدن نیاز دارند.

هنگامی که نرم افزار کلیه آزمونهای پذیرش را با موفقیت گذراند، می تواند مورد پذیرش نهایی قرار گیرد.

سند تاریخیچه پروژه (PHD: Project History Document)، اطلاعات مدیریتی مهم جمع آوری شده در طول جریان پروژه را خلاصه می نماید. این سند باید پس از پذیرش نهایی صادر شود. این سند می بایست در پایان چرخه حیات، به همراه سایر اطلاعات گردآوری شده در مرحله بهره برداری و نگهداری مجدداً منتشر و اعلام گردد.

- پس از پذیرش نهایی، نرم افزار باید به منظور تصحیح خطاهای آشکار شده در طول مراحل قبل و یا به علت نیازهای جدیدی که رخ می دهند، اصلاح گردد. این عمل "نگهداری" نامیده می شود.
- در تمام دوران بهره برداری می بایست توجه ویژه ای جهت به هنگام نگهداشتن مستندات اعمال گردد. اطلاعات خطاها و نواقص می بایست به منظور تهیه داده های خام برای تعیین معیارهای کیفیت نرم افزار در پروژه های آتی، ضبط و ثبت گردد. همچنین می بایست از ابزار لازم به منظور تسهیل در جمع آوری و تحلیل داده های کیفیت نرم افزار استفاده شود.

چرخه حیات توسعه نرم افزار به سه فاز اصلی و یک فاز چتری (پشتیان) تقسیم بندی می شود. که فازهای اصلی عبارتند از:

(۱) تعریف

(۲) توسعه

(۳) نگهداری

هر یک از فازها خود به فازهایی تقسیم می شوند.

۱- فاز تعریف

در مرحله تعریف تاکید بر چه (WHAT) است:

چه توابع و عملکردی انتظار می رود؟

چه رفتاری باید سیستم داشته باشد؟

چه واسط‌های کاربری باید طراحی شود؟
چه محدودیت‌هایی برای طراحی وجود دارد؟
معیارهای صحت اعتبار برای تعریف یک نرم افزار موفق چیست؟

در این مرحله روش مورد استفاده با انجام سه فعالیت زیر مشخص می شود:
مهندسی سیستم یا اطلاعات System/Information Engineering
برنامه ریزی پروژه نرم افزاری Software Project Planning
تحلیل نیازها Requirements' Analysis

۲- فاز توسعه

مرحله توسعه تاکید بر چگونگی (HOW) دارد:
چگونه داده ها باید سازمان دهی شوند؟
تکمیل وظایف چگونه در معماری وظایف گنجانده می شود؟
ارتباط و توصیف پردازشها چگونه انجام می گیرد؟
ویژگی های واسط‌های کاربری چیست؟
چگونه آزمایش انجام می گیرد؟
چگونه طراحی تبدیل به معماری برنامه و برنامه نویسی می گردد؟ (مهمترین فعالیت)

در این مرحله مدل های مورد نیاز تولید می گردد و روند کلی برنامه ها تعیین شده و ساختار پایگاه داده ها نیز رسم می شود.

سه فعالیت در این مرحله عبارتند از:
طراحی نرم افزار Software Design
تولید کد Coding
تست Testing

۳- فاز نگهداری

مرحله سوم مرحله نگه داری است که شامل:
نگه داری اصلاحی (Correction): اشکالات در چرخه زندگی پروژه شناسایی، بررسی و رفع می گردند.

نگهدار تطبیقی (Adaptation): تغییر به مرور زمان و با توجه به نیازهای محیطی و کاربر. نیاز جدید user را باید با محیط تطبیق داد منوط به اینکه نرم افزار قابلیت توسعه را داشته باشد.

نگه داری ارتقایی (Enhancement): ارتقا نرم افزار پس از راه اندازی که توسط کاربر خواسته می شود.

نگه داری پیشگیرانه (Prevention): ایجاد تغییراتی در برنامه ها که اصلاح و بهبود نرم افزار را تسهیل می کند.

نرم افزار پس از دوره مشخصی بازنگری شده و اصلاحات لازم انجام می شود، درواقع نرم افزار جدیدی تولید می گردد.

فعالیت های چتری (پشتیبانی) :

فعالیت های پشتیبانی زیر در کلیه مراحل فوق انجام می گردد:

کنترل و نظارت بر برنامه ریزی پروژه

بازنگری های مستمر فنی و رسمی (Quality Control)

اطمینان مرغوبیت نرم افزار (شامل تمامی مراحل) Quality Assurance

مدیریت پیکربندی نرم افزار Configuration/Change Management

تهیه و تدوین مستندات

اندازه گیری (مقیاسهای اندازه گیری) Software Measurements

مدیریت ریسک (پیامدهایی که برآورد زمان و هزینه را تغییر دهد) Risk Management

۲.۲ نرم افزار کاربردی

نرم افزار کاربردی که در زبان انگلیسی به آن Application software می گویند، عبارت است از نرم افزاری که با استفاده از منابع و قابلیت های رایانه کاری را مستقیماً برای کاربر انجام می دهد. باید توجه داشت که این عبارت در مقابل عبارت نرم افزار سیستمی معنی پیدا می کند.

نرم افزار سیستمی در مقابل در پس زمینه عمل می کند و خدماتی را فراهم می کند که دیگر نرم افزارها یا سیستم عامل می توانند برای انجام کارهای خود از آن استفاده کنند. اما در عوض معمولاً مستقیماً با کاربر عادی در تماس نیست و خدماتی را به او ارائه نمی دهد.

در عمل، بیشتر نرم‌افزارهایی که کاربران با آنها سر و کار دارند از این دسته محسوب می‌شوند. برای مثال می‌توان به نرم‌افزارهای رومیزی یعنی واژه‌پردازها، آفیس، نرم‌افزارهای طراحی گرافیکی، بازی‌های رایانه‌ای و امثال آنها اشاره کرد.

بسیاری از نرم‌افزارهای کاربردی، برای توسعه‌دهندگان ابزار رابط برنامه‌نویسی کاربردی را فراهم می‌کنند تا بتوان از قابلیت‌های نرم‌افزار در نرم‌افزارهای جدید استفاده کرد. برای مثال نرم‌افزار ادوبی آکروبات هنگام نصب، ای‌پی‌آی فراهم می‌کند که برنامه‌نویس می‌تواند با استفاده از آن توانایی‌های آکروبات را در برنامه خود به کار گیرد.

۲,۳ سامانه‌ی اطلاعاتی

سیستم اطلاعاتی یا سامانه اطلاعاتی یک پایگاه داده برای ذخیره، پردازش و تجزیه و تحلیل نتایج گزارش‌هایی است که به‌طور منظم در حال انجام است. به این دلیل سیستم‌های اطلاعاتی نه تنها به عنوان یک پایگاه داده، نرم‌افزار و سخت‌افزار بلکه به عنوان یک سیستم بزرگتر که به کمک و بهره‌گیری از آن می‌توان مدیریت تمام امور دستی و دستگاهی و تفسیر سیستم‌های ارتباطی را به راحتی انجام داد، مطرح می‌شوند.

با این حال، این اصطلاح در معنای وسیع‌تر به هر وسیله‌ای که باعث برقراری ارتباط علمی بین افراد (به‌طور مثال از طریق ارتباط کلامی، سیستم‌های پانچ کارت و سیستم‌های تصادفی نوری ساده) می‌شود، اطلاق می‌شود. همچنین گاهی اوقات تنها برای اشاره به نرم‌افزار مورد استفاده برای اجرای یک پایگاه داده کامپیوتری یا برای اشاره به یک سیستم کامپیوتری استفاده می‌گردد. مجموعه سیستم‌های اطلاعاتی، برای مطالعه علمی، دقیق و واقعی شبکه‌های متشکل از سخت‌افزار و نرم‌افزار، که مردم و سازمان‌ها برای جمع‌آوری، فرایند ایجاد و توزیع اطلاعات از آنها استفاده می‌کنند مورد استفاده قرار می‌گیرند. هدف هر سیستم ارتباطی حمایت از عملیات، مدیریت و تصمیم‌گیری و در معنای وسیع‌تر کاهش مدت استفاده می‌باشد؛ که می‌توان به فناوری اطلاعات و ارتباطاتی که یک سازمان، یا تکنولوژی که مردم در حمایت از فرایندهای کسب و کار انجام می‌دهند، اشاره کرد.

۲,۴ لایه‌ی کاربرد شبکه و وب

مدل OSI:

این مدل برگرفته از کلمه‌ی Open System Interconnection است و برای ارتباط بین دو کامپیوتر مبدأ و مقصد به کار می‌رود. این مدل در سال ۱۹۸۰ توسط سازمان ISO طراحی و پیاده‌سازی شده است و طبق سالیان متوالی تغییراتی روی آن صورت گرفته است، هرچند که همان ساختار اصلی خود را حفظ کرده است.

این مدل بر اساس یکی سری قراردادها با لایه-ی مقابل خود در کامپیوتر دیگر ارتباط برقرار می‌کند و این کار باعث افزایش سرعت و امنیت در شبکه خواهد شد.

لایه کاربرد شبکه در مدل OSI

لایه کاربرد (Application Layer) را می‌توان ملموس‌ترین لایه در مدل مرجع OSI نام برد. زیرا حتی نا آشنا ترین افراد هم در حوزه شبکه حداقل یک بار با این لایه کار کرده‌اند. بطور مثال، همه‌ی ما با مرورگرها کار کرده‌ایم. لایه اپلیکیشن بالا ترین لایه در مدل OSI می‌باشد و رابطی است بین کاربر و سیستم برای ارائه سرویس به کاربر. این ارتباط می‌تواند بصورت Command Line و یا بصورت گرافیکی باشد. این لایه سرویس‌ها و پروتکل‌های مورد نیاز ما را تامین می‌کند و همچنین وظیفه ارتباط با لایه‌های زیرین خود را نیز دارد. چندین سرویس را که این لایه به ما ارائه می‌دهد:

HTTP

این سرویس برای زمانی است که کاربر مرورگر خود را باز کرده و اسم سایت مورد نظر خود را در قسمت URL وارد می‌کند. در این زمان لایه اپلیکیشن درخواست کاربر را بصورت HTTP ارسال می‌کند و درواقع درخواست می‌کند که محتوا یا Content سایت مورد نظر را برای ما به نمایش بگذارد.

File Transfer Protocol) FTP

این سرویس مخصوص انتقال فایل می‌باشد. بدین صورت که در محیط‌های سازمانی برای ذخیره سازی اطلاعات و یا دسترسی به اسناد از این سرویس استفاده می‌کنند.

Remote Access

این سرویس در واقع به کاربر این امکان را می‌دهد که از راه دور به سیستم خود متصل شود. از جمله نرم افزار هایی که این سرویس را ارائه می‌دهند نرم افزار Team Viewer را می‌توان نام برد.

SMTP

این سرویس مخصوص درخواست ارسال Email می‌باشد.

این نکته را باید بدانیم که لایه اپلیکیشن در واقع پروتکل‌ها را ایجاد می‌کند. برای رفع نیازهای کاربر و شرکت‌های مختلف نرم افزارهایی را بر اساس این پروتکل‌ها ایجاد می‌کنند که همان سرویس را به کاربر بدهد.

شبکه client / server

سرور یک برنامه (کامپیوتری Process) است که دارای اطلاعات است و برای دیگر برنامه های کامپیوتری سرویس و خدمات فراهم می کند.

کلاینت یک برنامه (کامپیوتری Process) است که نیاز به اطلاعات دارد و از سرویس ارایه شده توسط سرور استفاده می کند.

ارتباط بین دو پروسس کلاینت و سرور با درخواست از طرف کلاینت و ارایه پاسخ از طرف سرور انجام می شود.

سرور به دیگر کامپیوترهای شبکه سرویس می دهد. در واقع به تقاضای کامپیوترهای دیگر پاسخ مناسب می دهد که این پاسخ

می تواند اطلاعات مورد تقاضای آن ها و یا اجرای دستورالعمل باشد.

برای تحویل درخواست (Request) یا پاسخ (Response) نیاز به آدرس دیگری به نام شماره پورت است

تا آن به یکی از پروسس های در حال اجرا با پروتکل خاص روی هاست مقصد تحویل داده شود.

ویژگی ها :

- در معماری فوق از سرویس دهندگان و سرویس گیرندگان با خصایص متفاوت استفاده می شود.
- اصل تقسیم کار دنبال و سرویس دهنده عملیات سنگین با پردازش بالا و سرویس گیرنده عملیات سبک را انجام خواهند داد.
- دو بخش متفاوت یک برنامه ، در جهت انجام عملیات با یکدیگر تشریک مساعی می نمایند.
- سرویس گیرنده با ارسال درخواست و سرویس دهنده با پاسخ به درخواست جلوه ای از همیاری در پردازش عملیات را بنمایش می گذارند.
- پلات فورم و سیستم های عامل سرویس دهنده و سرویس گیرنده می تواند متفاوت باشد.
- عملیاتی را که یک برنامه انجام می دهد بین سرویس دهنده و سرویس گیرنده تقسیم می گردد.

مزایا :

- بهره گیری مناسب از پتانسیل های سخت افزاری موجود با توجه به اصل تقسیم عملیات ها
- بهینه سازی استفاده و بکارگیری منابع اشتراکی .
- بهینه سازی توانائی کاربران از بعد انجام فعالیت های متفاوت

معایب :

- عدم وجود امکانات لازم برای کپسوله نمودن سیاست های راهبردی نرم افزار
- کاهش کارائی برنامه همزمان با افزایش تعداد کاربران همزمان
- بهبود عملکرد برنامه و یا اعمال اصلاحات مورد نظر همواره یکی از چالش های جدی است .

اتصالات کلاینت به سرور:

اتصال به سرور از طریق LAN

در اینگونه اتصالات کامپیوترها بوسیله کابل هایی به سرور متصل میشوند.

خط تلفن:

کامپیوتر کلاینت بوسیله خط تلفن با یک سرور از راه دور میتواند ارتباط برقرار کند.

یک شبکه WAN:

در این مورد اتصالات مبتنی بر TCP/IP است و در اینترنت انجام میشود.

یک شبکه بی سیم:

این گزینه که در سطح وسیع انجام نمیشود امکان اتصال کامپیوترها بوسیله امواج بی سیم را فراهم میکند.

انواع پردازش

۱ - پردازش های مبتنی بر میزبان .

مدل فوق بمنزله یک مدل سرویس دهنده / سرویس گیرنده تلقی نشده و مشابه مدل MainFarme است .

۲ - پردازش های مبتنی بر سرویس دهنده .

در این مدل سرویس دهنده تمامی پردازش های مربوطه را انجام و سرویس گیرنده مسئولیت ایجاد بخش رابط کاربر را برعهده خواهد داشت.

۳ - پردازش های مبتنی بر سرویس گیرنده .

تمامی عملیات بر روی سرویس گیرنده انجام خواهد شد. عملیات مربوط به بررسی صحت داده ها و سایر عملیات مربوط به منطق بانک های اطلاعاتی بر روی سرویس دهنده انجام خواهد شد.

۴ - پردازش های مبتنی بر همیاری .

در این مدل سرویس دهنده و سرویس گیرنده بمنظور انجام یک فعالیت با یکدیگر تشریک مساعی خواهند کرد.

شماره پورت port number

شماره پورت یک عدد ۱۶بیتی است برای تعیین این که یک بسته اطلاعاتی در شبکه به چه برنامه ای در هاست مقصد تعلق دارد.

برای انتقال صفحات وب،پورت شماره ۸۰ (HTTP)

برای انتقال فایل،پورت شماره ۲۱ (FTP)

برای انتقال ایمیل پورت شماره ۲۵ (SMTP)

معماری توزیع شده client/server

(۱) کلیه خصوصیات این نوع از معماری شبیه ThreeTier است.

(۲) انواع سرور در این معماری وجود دارد.

(۳) در یک معماری توزیع شده نقاط منطقی ممکن است ازتعدد بالایی برخوردار باشند.

(۴) از نقاط قوت معماری توزیع شده کاهش بار ترافیک در محیط هایی با پهنای باند کم میباشد.

توزیع شدگی:

مبتنی بر توزیع بار کاری در اعضای یک سیستم به منظور دستیابی به کارایی و بهره وری بالاتر.

معماری توزیع شده راه حل مناسبی برای طراحی ساخت سیستم هایی است که خصوصیات زیر را داشته باشند:

(۱) قدرت پردازش در حجم بالا

(۲) پهنای باند ارتباطی کم

(۳) تحمل خرابی

(۴) توان عملیاتی

(۵) فشار زمانی بالا

(۶) پشتیبانی ساده

انتخاب الگوی توزیع

برای انتخاب الگوی مناسب باید به معیارهای زیر توجه کنیم.

(۱) مقیاس پذیری

(۲) هزینه

(۳) پشتیبانی ساده

(۴) تحمل خرابی

کلاینت و سرور مبتنی بر وب

با رشد اینترنت (هم چنین اینترنت و اکسترانت)، به مرور توسعه نرم افزارها به صورت Web Based Application مورد توجه قرار گرفت

در این مدل دیگه نیازی به پیاده سازی سوکت پروگرامینگ نیست چون نرم افزار تحت وب ذاتا آن را در خودش دارد. در این روش تعداد کاربران نامحدود است.

بیشتر کاربران دارای ابزاری کلاینتی برای دسترسی به محتوای وبی هستند چون که سیستم عامل های امروزی شامل مرورگرهای وب نیز می باشند.

برنامه های کلاینت سروری تحت وب به صورت URL-Driven است.

FAT CLIENT

یک Fat Client با وجود اینکه به سرور متصل می شود ولی بیشتر پردازش ها در سمت کلاینت Client Side انجام می شود.

با Fat Client پردازش های زیادی از طریق شبکه انجام نمی شود و برای شبکه هایی با تراکم بالا و یا سرعت ارتباطی پایین بسیار مناسب است.

مزایا:

- (۱) هزینه پایین تر برای راه اندازی سرور
- (۲) کارایی بیشتر در نرم افزار های چندرسانه ای
- (۳) بالا رفتن ظرفیت سرور برای پشتیبانی کردن از تعداد بیشتری کاربر

THIN CLIENT

اساساً Thin Client یک اپلیکیشن web based است و بیشتر پردازش ها در سمت سرور Server Side انجام می شود.

Thin client جهت انجام بسیاری از کارهای خود به شدت وابسته به سرور است

مهم ترین مزیت Thin Client مدیریت و نظارت آسان است.

معمولاً نرم افزار Thin Client روی یک کامپیوتر با حداقل امکانات و اندازه کوچک و باریک نصب می شود.

نمونه هایی از معماری Client Server

اینترنت بستری مناسب برای پیاده سازی برنامه های توزیع شده است . وجود زیر ساخت های لازم در این زمینه شرایط مطلوبی را برای پیاده سازی برنامه های توزیع شده ، فراهم آورده است

دستیابی به یک وب سایت و یا ارسال و دریافت پیام های الکترونیکی نمونه هایی از برنامه های توزیع شده بر روی بستر اینترنت می باشند.

برنامه های پست الکترونیکی نمونه ای مناسب در این زمینه می باشند

برنامه های Chat نمونه دیگری از برنامه های توزیع شده بر روی بستر اینترنت می باشند.

۲.۶ نرم افزار تحت وب

در نرم افزارهای تحت وب دیگر پردازش های مربوط به نرم افزار مورد نظر بر روی سیستم کاربر صورت نمی گیرد. کاربر یک درخواست به سیستم خدمات دهنده (سرور) ارسال می کند، سیستم خدمات دهنده بر روی بستر اینترنت قرار دارد و به عبارتی تحت وب است، سپس سرور پردازش های لازم را انجام داده و نتایج را برای کاربر ارسال می کند.

بدین شکل کاربر با ساده ترین و قدیمی ترین سیستم ها نیز می توانید به قوی ترین خدمات تحت وب و نرم افزارهای تحت وب دسترسی داشته باشید زیرا دیگر نیازی به یک پردازنده قدرتمند ندارید و سیستم کاربر فقط وظیفه نمایش اطلاعات را بر عهده دارد نه پردازش داده ها را.

از طرفی وقتی کاربر یک نرم افزار را روی رایانه شخصی خود نصب می کند برای دسترسی به این نرم افزار باید حتما از همان رایانه استفاده کنید ولی در نرم افزارهای تحت وب دیگر کاربر محدود به هیچ سیستم یا مکان خاصی نیستید. هر کجا که اینترنت باشد کاربر به خدمات نرم افزارهای تحت وب دسترسی خواهید داشت.

به طوری کلی می توان گفت تمام مزایای یک نرم افزار محلی در نرم افزارهای تحت وب وجود دارد ولی بسیاری از ویژگی های نرم افزارهای تحت وب در نرم افزارهای محلی قابل دسترس نیست. مزایای متعدد نرم افزارهای تحت وب دنیا را به سمت آنلاین شدن سوق می دهد، به طوری که در آینده ای نه چندان دور دیگر نیازی به سیستم های قدرتمند شخصی جهت پردازش و ذخیره سازی اطلاعات نداریم و وجود اینترنت پر سرعت، ما را از تمامی مزایای نرم افزارهای تحت وب بهره مند خواهد ساخت.

از جمله مهمترین مزایای نرم افزارهای تحت وب می توان به موارد زیر اشاره کرد:

- محدود نبودن به مکان خاص و در هر جایی که اینترنت باشد در دسترس خواهند بود
- محدود به یک سیستم خاص نیستند
- تمامی پردازش های پیچیده و سنگین بر روی سرور انجام شده و می توان با سیستمی ساده از خدمات نرم افزارهای تحت وب استفاده نمود.
- در داده های حساس می توان از امنیتی به مراتب بالا و قابل اطمینان برخوردار بود.

۲,۷ معماری نرم افزار و الگوی معماری

معماری لایه-ای

لایه بندی یکی از رایج ترین تکنیک هایی است که طراحان نرم افزار برای بخش بندی و طراحی سیستم های پیچیده از آن استفاده می-کنند. به طوری که هر لایه-ای کار خودش را به صورت منسجم انجام می-دهد و از ریز کار لایه-ی دیگر باخبر نیست تنها از سرویس های لایه-ی پایین تر استفاده می-کند و یک سری سرویس را در اختیار لایه-ی بالاتر می-گذارد. همچنین لایه بندی باعث می-شود تا لایه ها کمترین وابستگی را به هم داشته باشند.

معماری ۳ لایه

همزمان با اینکه معماری Client-Server در محبوبیت به سر می برد، دنیای شی گرایی نیز در حال رشد بود و و به دنبال راه حلی برای حل مشکل Domain Logic و یا به زبان ساده تفکیک منطق برنامه از سایر قسمت ها بود، در نهایت این تلاش ها منجر به معرفی معماری ۳ لایه گردید. در این روش شما هر برنامه از سه لایه ی اصلی تشکیل می شود.

Presentation

بالاترین لایه Presentation نام دارد و در واقع لایه ایی است که کاربران آن را مشاهده می کنند. این لایه همان UI برنامه می باشد.

Domain Logic

در لایه دوم منطق برنامه قرار دارد، این لایه Domain Logic نام دارد. لایه‌ای که منطق برنامه آنرا قرار می‌گیرد، در واقع جایی که دستورات، شرطی‌ها و بررسی‌های یک فرایند نوشته می‌شود را Domain Logic می‌گویند.

Data Access

در نهایت در معماری سه لایه، لایه آخر مربوط به فعالیت‌های مرتبط با پایگاه داده می‌باشد و از همین رو نام این لایه را Data Access می‌نامند.

در این روش می‌توان به سادگی کدهای هر بخش را از یکدیگر جدا کرد و با استفاده از همین تفکیک پذیری ساده می‌توانید قسمت‌های مختلف یک پروژه را با تیم‌های تخصصی طراحی و پیاده‌سازی کرد، مثلاً از بخش UI درخواست کرد بدون هیچ نگرانی UI صفحات مختلف و یا فرم‌های مختلف پروژه را طراحی کنند و سپس در بخش دیگر برنامه نویسان کدهای این صفحات و فرم‌ها را تکمیل کنند. همچنین با جدا شدن لایه Data Access می‌توان کدهای کار با پایگاه داده را در پروژه‌های مختلف به سادگی استفاده نمود.

همانطور که مشاهده می‌کنید حاصل این تفکیک‌ها به وجود آمدن لایه‌های تخصصی همچون Entity Framework است که تمامی مسائل مرتبط با پایگاه داده را مدیریت می‌کند.

همین جا می‌بایست به یک نکته اشاره کرد، در بسیاری از وب‌سایت‌ها سوالی مبنی بر بهترین معماری مطرح می‌شود. شاید بتوان این سوال را با سوالی مبنی بر لپ‌تاپ بهتر است و یا تبلت مقایسه کرد. هر دو این سوال‌ها وابسته به نیاز مصرف‌کننده دارند طبیعتاً هر پروژه‌ای مانند هر ساختمانی نیاز به معماری خود دارد و نباید انتظار داشت معماری یک هتل چندین طبقه با معماری یک ساختمان کوچک یکسان باشد زیرا این معماری‌ها باعث افزایش هزینه و ایجاد مسائل پیچیده می‌شوند، در نهایت زمانی این هزینه‌ها مقرون به صرفه و ارزشمند خواهند بود که وابسته به هزینه‌های صورت گرفته سودآوری اتفاق بیافتد.

طبیعتاً شالوده یک هتل با شالوده یک ساختمان کوچک هیچگاه قابل قیاس نیست، پس منطقی نیست برای پروژه‌های کوچک روش‌های پیچیده استفاده شود، اما نکته‌ای که همه در مورد آن اشتباه فکر می‌کنند این است که تصور می‌کنند پروژه‌های کوچک باید به صورت خیلی ساده و در واقع بدون هیچ معماری به صورت درهم‌کد نویسی و طراحی شود، این تفکر نیز طبیعتاً اشتباه است باید در پروژه‌های کوچک نیز از معماری مناسبی استفاده

کرد تا فردا بتوان با کمترین هزینه آن را رشد و توسعه داد زیرا تنها صنعتی که در دنیا هیچگاه به اتمام نمی رسد صنعت توسعه نرم افزار است و ممکن است همین پروژه کوچک تبدیل به یک پروژه بسیار عظیم شود.

۲,۸ معماری MVC و تفکیک دغدغه‌ها

عبارت MVC یا Model View Controller واژه‌ای است که در هنگام کار کردن با فریم ورک‌ها مورد استفاده قرار می‌گیرد.

این عبارت شکلی خاص از ترکیب کدهای برنامه‌نویسی است. در این شکل از برنامه‌نویسی کدهای برنامه به سه بخش تقسیم می‌شوند. با تقسیم و طبقه بندی کدها، برنامه به صورت سه لایه مجزا از هم در می‌آید. برنامه‌نویسی لایه‌ای مزایای زیادی دارد که در محدوده‌ی این نوشته نمی‌گنجد. تنها به همین نکته بسنده می‌کنیم که در برنامه نویسی MVC بخش View وظیفه‌ی نمایش اطلاعات به کاربر را بر عهده دارد. بخش Controller دریافت اطلاعات از کاربر و پردازش آن را به انجام می‌رساند. و بخش Model کار ذخیره و بازیابی اطلاعات را عملی می‌کند.

۲,۹ فریم‌ورک

برنامه‌نویسان فارغ از زبان برنامه‌نویسی که با آن فعالیت می‌کنند با فریم ورک‌های برنامه‌نویسی نیز مواجهند. این فریم ورک‌ها مجموعه‌ای کدهای از پیش آماده، کتابخانه‌های برنامه‌نویسی و قوانین توسعه‌ی نرم‌افزار هستند. فریم‌ورک در اصل محیطی است که برای یک زبان برنامه نویسی خاص ایجاد می شود و باعث آسان تر شدن کار می شود. بطور کلی می توان فریم ورک را مجموعه‌ای از ابزارها، کتابخانه ها، قرارداد ها و بهترین شیوه ها تعریف کرد که فریم ورک ها این امکان را به برنامه نویسان می دهند تا با توابع و کلاس های از پیش تعریف شده، ضمن ایجاد امنیت بیشتر برای برنامه‌ها، از کد کمتری در برنامه های خود استفاده کنند. فریم ورک الزاماً چیزی ساخته شده و بصورت پکیج درآمده برای عموم نیست؛ چرا که ممکن است یک فریم ورک منحصرأ برای شخص یا یک تیم طراحی شده باشد.

۲,۱۰ مدلسازی نرم‌افزار

در فرآیند تحلیل و طراحی سیستم‌های نرم‌افزاری به منظور ایجاد یک نمود تصویری قابل بحث و قابل توضیح، مهندسین نرم‌افزار اقدام به مدلسازی سیستم از جنبه‌های مختلف می‌کنند. برخی از این جنبه‌ها به طور کلی عبارتند از:

- مدلسازی داده‌ای

- مدلسازی ساختاری

- مدلسازی رفتاری

از جمله فواید مدلسازی می‌تواند به این موضوع اشاره کرد که توضیح ویژگی‌های محصول به کارفرما راحت‌تر شده و همچنین فرآیند عیب‌یابی می‌تواند پیش‌تر آغاز شود.

در این مستند از چند نمونه دیاگرام به منظور مدلسازی سامانه استفاده کرده ایم که در بخش مستندات فنی قابل مشاهده می‌باشد.

۲.۱۱ مستندهای متداول پروژه‌های نرم‌افزاری

در پروژه‌های نرم‌افزاری مستندات مختلفی مورد نگارش قرار می‌گیرند که بسیاری از آن‌ها دقیقاً مطابق با مستندات هر نوع پروژه‌ی دیگر هستند، از جمله «طرح پیشنهادی» یا همان پروپوزال. علاوه بر این مستندات پروژه، مستنداتی نیز در حوزه‌ی حرفه‌ای مهندسين نرم‌افزار قرار می‌گیرد که به طور مثال بر مسائل فنی یا کاربری سیستم نرم‌افزاری تمرکز دارند از جمله:

- سند نیازمندی‌ها که بیانگر ویژگی‌ها و قابلیت‌ها سیستم مورد توسعه است.
- سند طراحی/معماری که بیانگر ساختار و روابط اجزای سیستم در داخل و با محیط خارج آن است
- سند فنی که بیانگر کنوانسیون‌ها، قراردادهای الگوریتم‌ها و تشریح API‌ها می‌باشد
- سند راهنمای کاربری که توسط کاربران سامانه و تیم پشتیبانی مورد استفاده قرار می‌گیرد.

۲.۱۲ اصل DRY در توسعه‌ی نرم‌افزار

در حوزه‌ی توسعه‌ی نرم‌افزار، اصول و قواعد بسیاری وجود دارد که گاهی یکی از دیگری مهم‌تر جلوه می‌کند اما یکی از اساسی‌ترین قواعد برنامه‌نویسی، قانون DRY است که مخفف واژگان Don't Repeat Yourself به معنی «دوباره کاری نکن» است!

این قانون توسط دو توسعه‌دهنده به نام‌های Andy Hunt و Dave Thomas ابداع شد که بسیاری از دیزاین پترن‌های معروف برنامه‌نویسی، ریشه در این قانون دارند

برنامه‌نویسی که بتواند تشخیص دهد کدام بخش‌های کد اصطلاحاً Duplicate یا «مشابه» هستند و تمام تلاش خود را به کار بندد تا با استفاده از کلاس‌ها و فانکشن‌های مختلف، میزان استفاده از کدهای تکراری در سراسر

برنامه را به حداقل برساند، در نهایت سورس کد تمیز تری تحویل خواهد داد که در آینده نگهداری چنین پروژه‌ی به مراتب راحت‌تر از سورس کدی است که پر است از کدهای مشابه!

هرچه میزان کدهای دوپلیکیت در سورس کد بیشتر باشد، احتمال ایجاد باگ در آینده به مراتب بیشتر خواهد شد

در فرایند توسعه‌ی نرم افزار، بخش‌های بسیاری از کد را می‌توان دید که تکراری هستند و قانون DRY دقیقاً برای چنین موقعیت‌هایی است.

دوپلیکیت شدن در منطق نرم‌افزار می‌تواند به اشکال مختلفی جلوه کند که از جمله‌ی رایج‌ترین آن‌ها می‌توان به آبجکت‌هایی که از روی کلاس خاصی ساخته می‌شوند اشاره کرد و اینجا است که بسیاری از دیزاین پترن‌ها به داد توسعه دهندگان می‌آیند. در واقع، ابداع دیزاین پترن‌ها یا «الگوهای طراحی» جلوگیری از استفاده از کدهای مشابه است.

علاوه بر دیزاین پترن‌ها، یکسری اصول کدنویسی که تحت عنوان SOLID شناخته می‌شوند نیز بر پایه‌ی اصل DRY هستند به طور مثال، حرف S در ابتدای SOLID به اصطلاح Single Responsibility اشاره دارد. به عبارت دیگر، هر کلاسی که در پروژه‌ی خود ایجاد می‌کنیم فقط و فقط باید مسئول یک کار باشد و در صورت نیاز به اعمال تغییرات در کلاس مد نظر، فقط و فقط باید یک دلیل برای ایجاد آن تغییر وجود داشته باشد نه اینکه کلاس مد نظر در جای جای نرم‌افزار برای کارهای مختلفی استفاده شده باشد و به هر دلیلی، نیاز به اعمال تغییرات در آن کلاس داشته باشیم.

۲,۱۳ حمله‌ی CSRF

حملات CSRF یا XSRF، در این حمله کاربر در صفحه مدیریت سایت خود لاگین است یا مهاجم از طریق یک لینک کاربر را مجبور به ورود به مدیریت سایت می‌کند بدون آنکه کاربر متوجه این موضوع باشد و در نهایت به صفحه مورد نظر دسترسی خواهد یافت. این حمله از طریق کشف آدرس‌هایی که محتوایی را تایید و ارسال میکنند اتفاق می‌افتد (مانند صفحات لاگین). کدهای اجرایی میتواند از قالب یک URL source در یک تصویر ذخیره شود بصورتی که کاربر از رخداد و اجرای URL آگاهی پیدا نکند. عموماً تزریق از طریق image tag در

HTML یا JavaScript انجام می شود در صورتی که در سایت بصورت غیر امن (http) متصل باشیم و یا کوکی ها روی مرورگر ذخیره شده باشند این حمله با احتمال موفقیت بالاتری اجرا خواهد شد.

سرویس PayPal یک مرتبه با آسیب پذیری CSRF روبرو شده است. نمونه دیگر این مشکل در Google رخ داد، در حالی که مهاجم میتواند اطلاعات اکانت Google کاربر را کشف کند و به اکانت قربانی دسترسی یابد. عموماً صفحات ثبت نام عمومی که با ارسال و دریافت اطلاعات طراحی شده اند بیشتر هدف این حمله قرار میگیرند و از متدهایی مانند GET یا POST میتواند استفاده کند.

راهکار رایج برای مقابله این مشکل ایجاد یک رشته کد (توکن) بصورت تصادفی که هم برای کوکی استفاده میشود و هم در آدرس ایجاد شده بصورت پنهان وجود خواهد داشت. در هنگام ارسال فرم بررسی می شود آیا CSRFTOKEN در فرم همان CSRFTOKEN ذخیره شده در کوکی است یا خیر و در صورت یکسان بودن اجازه اجرا و ارسال اطلاعات و یا لاگین را خواهد داد.

راهکارها:

- قرار گرفتن یک Token منحصر به فرد بصورت پنهان در بدنه درخواست HTTP ارسال میشود.
- استفاده از reauthenticate درخواست با استفاده از ابزاری مانند کد CAPTCHA
- استفاده از OWASP's CSRF Guard

OWASP سازمان بین المللی جهت استاندارد کردن ایمن سازی در طراحی و پیاده سازی می باشد.
(Open Web Application Security Project)

۲.۱۴ برنامه نویسی شیء گرا

«برنامه نویسی شیء گرا» (Object-Oriented Programming) یا به اختصار OOP یک الگو یا شیوه تفکر در برنامه نویسی است که برگرفته از دنیای واقعی بوده و از دهه ۱۹۶۰ میلادی مطرح گشته است. به زبانی که از این الگو پشتیبانی کند، «زبان شیء گرا» گفته می شود؛ Simula ۶۷ و Smalltalk نخستین زبان های برنامه نویسی شیء گرا هستند. ایده شیء گرایی در پاسخ به برخی از نیازها که الگوهای موجود پاسخگو آنها نبودند به وجود آمد؛

نیازهایی مانند: توانایی حل تمامی مسائل پیچیده (Complex)، «پنهان سازی داده» (Data Hiding)، «قابلیت استفاده مجدد» (Reusability) بیشتر، وابستگی کمتر به توابع، انعطاف بالا و...

رویکرد برنامه نویسی شی گرا «از پایین به بالا» (Bottom-Up) است؛ یعنی ابتدا واحدهایی کوچک از برنامه ایجاد می شوند و سپس با پیوند این واحدها، واحدهایی بزرگتر و در نهایت شکلی کامل از برنامه به وجود می آید. برنامه نویسی شی گرا در قالب دو مفهوم «کلاس» (Class) و «شی» (Object) ارایه می گردد. هر کلاس واحدی از برنامه است که تعدادی داده و عملیات را در خود نگهداری می کند و هر شی نیز حالتی (State) مشخص از یک کلاس می باشد.

در برنامه نویسی شی گرا، هر برنامه در قالب موجودیت های کوچکی که در واقع همان اشیا هستند و با یکدیگر تعامل دارند در نظر گرفته می شود. برای داشتن این اشیا می بایست ابتدا کلاس های برنامه را تعریف نماییم؛ هر کلاس «رفتار» (Behavior) و «صفات» (Attributes) اشیایی که قرار است از آن ایجاد شوند را تعریف می کند. از یک کلاس می توان هر تعداد که بخواهیم شی ایجاد نماییم. هر شی بیانگر یک «حالت» یا یک «نمونه» (Instance) از کلاس خود است.

۲،۱۵ نگاشت داده ها (Data Mapper و ORM)

ORM (الگوی طراحی یا تهاجمی ضد الگو)

مدیریت دیتا (اطلاعات) در برنامه های شیء گرا امری بسیار ضروری و مهم است. اینکه در یک برنامه ما چگونه بتوانیم اطلاعاتی را که کاربر و یا خودمان در حین اجرای برنامه و هنگامی که برنامه از حالت اجرا خارج شده است، حفظ، مدیریت و تغییر دهیم بسیار حیاتی است.

ORM یا Object Relational Mapper مجموعه ای از تکنیک های برنامه نویسی برای تبدیل داده ها بین سیستم های غیر شی گرا با محیط های برنامه نویسی شی گراست. به عبارت دیگر، برای تطبیق بانک های اطلاعاتی رابطه ای (مثل SQL Server, MySQL و ...) با زبان های برنامه نویسی شی گرا (مثل سی شارپ، جاوا و ...) و استفاده از امکاناتی همچون Inheritance از ORM ها استفاده می کنیم. در واقع با استفاده از ORM ها و استفاده از ابزارهایی که ORM در اختیار توسعه دهنده قرار می دهد، توسعه دهنده ساختار رابطه ای بانک اطلاعاتی خود را به ساختار شی گرا برنامه نویسی خود متصل می نماید و به این ترتیب پیچیدگی های مربوط به ذخیره سازی، بازیابی و ... در بانک اطلاعاتی را به عهده ORM می سپارید.

در پیاده سازی سیستم های بزرگ نرم افزاری معمولاً بخشی از انرژی و وقت تیم به پیاده سازی لایه داده سیستم (DAL) که وظیفه اتصال به بانک اطلاعاتی و ذخیره و بازیابی داده از آن را دارد صرف می شود که با استفاده از یک ORM بخش قابل توجهی از این انرژی را می توان صرف صرفه جویی نمود.

ORM محدود به هیچ زبان خاصی نیست و تنها یک تکنیک و سیستم است که می تواند در زبانهای مختلف پیاده سازی شود. در حقیقت ” یک شیء مجازی از دیتابیس ” داریم که می تواند در زبانهای برنامه نویسی استفاده شود و خروجی دیگر یک آرایه نیست. تعداد این ORM ها خیلی زیاد است و هر کدام تعدادی پایگاه داده رو پشتیبانی میکنند. هر دو بسته ” تجاری و رایگان ” این تکنولوژی وجود دارد.

اولین فریم ورک مناسب و خوبی که برای ORM تولید شد، EOF نام گرفت (Enterprise Objects Framework) که البته خود نیز دارای مشکلاتی بود . در واقع Nhibernate ورژن Open source و رایگان Hibernate (فریم ورک پیاده سازی ORM در Java) در دات نت است.

مزایا

الف) یک ابزار ORM زمان تحویل پروژه را کاهش می دهد

ب) یک ابزار ORM کدی با طراحی بهتر را تولید می کند

ج) استفاده از یک ابزار ORM ، کار برنامه نویسی را ساده تر می کند

۳ معرفی تکنولوژی‌های به کار رفته در توسعه‌ی سیستم

در این بخش با رویکردی عملی به معرفی ابزار و عناصری می‌پردازیم که به کمک آن‌ها و به کارگیری مفاهیم مطرح شده در بخش قبل سامانه را پیاده‌سازی کرده ایم.

۳.۱ ابزار کنترل نسخه‌ی کد منبع گیت

گیت یک نرم‌افزار کنترل نسخه و از مدل نرم‌افزارهای آزاد و متن‌باز برای بازنگری کد منبع توزیع شده و مدیریت منبع کد است که برای دنبال کردن تغییر فایل‌های کامپیوتری و دنبال کردن کارهای انجام شده روی آن‌ها توسط افراد مختلف است. هدف اولیه این نرم‌افزار برای استفاده در پروژه‌های نرم‌افزاری بوده‌است ولی می‌توان از آن تنها برای دنبال کردن تغییر فایل‌ها هم استفاده کرد.

گیت ابتدا برای توسعهٔ لینوکس توسط لینوس توروالدز به وجود آمد و اکنون پروژه‌های فراوانی از آن الهام گرفته‌اند. هر دایرکتوری کاری در گیت یک مخزن کامل با تاریخچهٔ کامل تغییرها و قابلیت بازنگری آن‌ها است و برای کار با آن نیازی به دسترسی به شبکه یا سرور مرکزی وجود ندارد. گیت یک نرم‌افزار آزاد است که تحت عنوان جی‌پی‌ال نسخه ۲ توزیع شده‌است.

۳.۲ سیستم عامل گنو/لینوکس

یک سیستم‌عامل شبه یونیکس است که بخش عمدهٔ آن سازگار با استاندارد پازیکس است. از دید فنی لینوکس تنها نامی است برای هستهٔ سیستم‌عامل و نه کل آن. هسته‌ای که نخستین بار در سال ۱۹۹۱ میلادی توسط لینوس توروالدز منتشر شد. اما به‌طور معمول این نام به سیستم‌عامل‌هایی که از ترکیب‌بندی لینوکس (به عنوان هسته سیستم‌عامل) با نرم‌افزارهای آزاد و متن‌باز بدست می‌آیند اطلاق می‌شود. در حالی که بنیاد نرم‌افزار آزاد

سیستم عامل حاصل از این ترکیب بندی را «گنو/لینوکس» می نامد اما عموماً آن را لینوکس می خوانند. این دوگانگی در نام نهادن منجر به بحث نامگذاری گنو/لینوکس شده است. محیط توسعه (و حتی ترجیحا عملیات) در این پروژه سیستم عامل گنو/لینوکس می باشد که در این مورد به طور خاص از توزیع Ubuntu 18.04 Bionic Beaver استفاده شده است.

۳.۳ زبان پایتون

پایتون از جمله زبان های برنامه نویسی قدرتمندی است که در زمینه علم داده ها، یادگیری ماشینی، خودکارسازی سامانه ها، توسعه وب، واسطه های برنامه نویسی و... به کار گرفته می شود. شاید بتوانیم این گونه بیان کنیم که پایتون در مقایسه با زبان های بزرگ یک زبان نسبتاً جدید به شمار می رود. این زبان برنامه نویسی در سال ۱۹۹۱ به دنیای برنامه نویسی وارد شد و به منظور پر کردن شکاف های موجود در دنیای برنامه نویسی و ارائه راهکاری به منظور نوشتن اسکریپت هایی که فرآیند انجام یکسری از کارهای روتین خسته کننده را به طور خودکار اجرا کنند یا ساخت یک نمونه اولیه از برنامه های کاربردی که در یک یا چند زبان دیگر پیاده سازی شوند، مورد استفاده قرار گرفت. با این حال در چند سال گذشته، پایتون به یکی از ابزارهای تراز اول در زمینه توسعه برنامه های کاربردی، مدیریت زیرساخت ها و تحلیل داده ها تبدیل شده است. امروزه پایتون در زمینه توسعه برنامه های کاربردی تحت وب و مدیریت سیستم ها و تجزیه و تحلیل بزرگ داده ها که رشد انفجاری به خود گرفته اند و همچنین هوش مصنوعی به یکی از بازیگران اصلی دنیای فناوری تبدیل شده است. پایتون این موفقیت چشم گیر و کاربرد گسترده را مدیون یکسری ویژگی های ارزشمندی است که هم در اختیار توسعه دهندگان حرفه ای و هم در اختیار توسعه دهندگان تازه کار قرار داده است. اساسی ترین کاربرد پایتون در ارتباط با اسکریپت نویسی و خودکارسازی است. پایتون تنها یک جایگزین برای اسکریپت های شل یا فایل های دسته ای نیست، به واسطه آنکه از پایتون برای تعامل خودکار با مرورگرهای وب، برنامه های گرافیکی و پیکربندی سیستم ها از طریق ابزارهایی همچون Salt و Ansible می توان استفاده کرد. در حال حاضر، دو نسخه از زبان پایتون در اختیار برنامه نویسان قرار دارد. هریک از این دو نسخه ویژگی های خاص خود را دارند. پایتون نسخه X.۲ نسخه قدیمی تر است و پشتیبانی رسمی و عرضه به روزرسانی ها برای این نسخه تا سال ۲۰۲۰ ادامه خواهد داشت. حتی این احتمال وجود دارد که پس از این تاریخ نیز به صورت غیررسمی از این نسخه پشتیبانی به عمل آید. جدیدترین نسخه ی این زبان نسخه ی ۷/۳ می باشد.

دستور زبان پایتون:

آشنایی با انواع دستورات در زبان برنامه نویسی پایتون

در یک تقسیم بندی می توان دستورهای پایتون را در دو دسته ی دستورهای ساده (Simple Statements) و دستورهای مرکب (Compound Statements) قرار داد که به اختصار به بررسی این دو گروه می پردازیم:

دستورهای ساده:

تا به حال تمام چیزهایی که در زبان پایتون آموخته ایم را در قالب دستورات ساده ای در اختیار مفسر قرار داده ایم تا آن ها را اجرا کند. به طور مثال در دستور زیر:

```
>>> num1 = 4 + 5
>>> message = ""Welcome to
SokanAcademy.com""
>>> num1 += 8
>>> 2 + 3 >= 6
False
```

همان طور که می بینیم بعضی از این دستورها خروجی خاصی ندارند، مثل سه دستور اول و در حقیقت با وارد کردن آن ها مفسر پایتون در پشت صحنه عملیات انتساب را انجام می دهد و ما بعداً می توانیم از نتایج این عملیات استفاده کنیم؛ مثلاً متغیر message که در این عملیات در حافظه ایجاد و به آبجکتی از نوع استرینگ "Welcome to SokanAcademy.com" منتسب شده است را چاپ کنیم. این در حالی است که دستور چهارم یک خروجی قابل مشاهده به ما نشان می دهد. همان طور که در این مثال ها می بینید هر یک از این دستورها را به صورت مستقل از هم در یک خط مثل دستور اول، یا در بیش از یک خط مثل دستور دوم نوشته ایم و اجرا می کنیم. در زبان پایتون اصطلاحاً به هر یک از این دستورهای ساده، یک Logical Line یا «سطر منطقی» گفته می شود.

به عنوان مثالی دیگر دستورات ساده ی زیر را در نظر بگیرید:

```
>>> message = "Welcome to
SyntaxError: EOL while scanning string literal
>>> message = "Welcome to\
SokanAcademy.com"
>>> message = "Welcome to \ # This is a line continuation
character.
SyntaxError: EOL while scanning string literal
```


همان طور که در دستور اول می بینید، وقتی بدون استفاده از کاراکتر \ دکمه ی اینتر را می زنیم تا ادامه ی دستور را در خط بعدی بنویسیم، مفسر پایتون بلافاصله اعلام خطای سینتکسی می کند؛ با این حال زمانی که از کاراکتر \ استفاده می کنیم به راحتی می توانیم به سطر بعدی برویم و ادامه ی دستور خود را وارد کنیم. در دستور سوم هم می بینید که بعد از قرار دادن کاراکتر \ نمی توان آن خط را ادامه داد و توضیح در آن وارد کرد، هرچند که مفسر توضیحات را نادیده می گیرد.

گرچه اکیدا توصیه می شود که دستورات ساده و مستقل از هم در خطوط جداگانه وارد شوند، با این حال برای نوشتن آن ها در یک خط کافی است بین آن ها کاراکتر ; را قرار دهیم..

در این صورت فراموش کردن درج کاراکتر ; (سمی کالن) بین دستورات منجر به بروز خطای سینتکسی می شود.

- دستورهای مرکب:

دستورهای مرکب اجرای سایر دستورها را کنترل می کنند یا بر روند اجرای آن ها تأثیر می گذارند. به طور مثال دستورهای شرطی در این گروه جای می گیرند که در ساده ترین حالت ممکن در صورت برقراری یک شرط خاص مفسر یک سری از دستورات را اجرا می کند، و در صورتی که شرط برقرار نباشد آن دستورات را نادیده می گیرد و اجرا نمی کند

یک دستور مرکب از یک یا چند Clause (کلاز یا بند) تشکیل می شود. یک بند از یک سر بند یا Header و یک بدنه یا Suite تشکیل شده است. هر سر بند با یک کلمه ی کلیدی یکتا آغاز می شود و با کاراکتر دو نقطه : پایان می یابد. پیش از این لیستی از کلمات کلیدی پایتون را به شما معرفی کردیم و گفتیم این کلمات شناسه هایی هستند که به صورت پیش فرض تعریف شده اند و معنای خاصی را به مفسر پایتون می رسانند. مثال هایی از تعریف هدر برای یک بند به صورت زیر است:

```
if x > y:
def move():
```

این هدرها با کلمات کلیدی if و def آغاز شده اند و با : پایان می یابند بدنه، گروهی از دستورات هستند که اجرای آن ها در یک بند کنترل می شود. دستورات بدنه ی یک بند را می توان به دو صورت نوشت:

۱- تمام دستورات بدنه را در همان خطی که سرزند را وارد کردیم بنویسیم و در صورتی که چند دستور وجود داشته باشد آن ها را با سمی کالن ; از هم جدا کنیم. فرم کلی چنین بندی به صورت زیر خواهد بود:

```
Clause Header : Statement1 ; Statement2 ; ...
```

به طور مثال:

```
if x > y: x = 0; y += 1; z = x+y
```

همان طور که می بینید در مثال بالا بدنه از سه دستور ساده تشکیل شده است که در همان خطی که هدر قرار دارد نوشته شده اند و با سمی کالن از هم جدا شده اند.

۲- دستورات بدنه را در سطرهای مجزا که در یک بلوک کد مجزا از بلوک سطر سرزند قرار دارند بنویسیم. بلوک بندی دستورات در بسیاری از زبان های برنامه نویسی هم چون جاوا و سی با استفاده از آکولادهای باز و بسته {} انجام می شود؛ حال این که در پایتون برای بلوک بندی کدها از تورفتگی یا Indentation استفاده می شود. تورفتگی با استفاده از فشردن کلیدهای Space یا Tab و ایجاد فاصله از آغاز سطر مشخص می شود.

استاندارد تورفتگی در پایتون مقداری برابر با فضای خالی ایجاد شده بواسطه ی چهار بار فشردن کلید Space یا یک بار فشردن کلید Tab است. البته در زمان نیاز، IDLE به صورت خودکار تورفتگی هر بلوک را ایجاد می کند.

مثالی از یک دستور مرکب که به این فرم نوشته شده است:

```
if x > y:
    x = 0
    y += 1
    z = x+y
>>>
```

در حالت تعاملی IDLE زمانی که تمام دستورات بدنه را وارد کردیم کافی است دو بار کلید اینتر را فشار دهیم تا مفسر بلوک بدنه را ببندد و دستور مرکب را اجرا کند. اما در حالت اسکریپتی کافی است خودمان اشاره گر را جا به جا کنیم تا به بلوکی که هدر در آن قرار دارد برگردیم و ادامه ی دستورها را وارد کنیم:

```
if x > y:
    x = 0
    y += 1
    z = x + y
age = 5
```

بدنه ی دستور مرکب if با دستور $z = x + y$ پایان یافته است و دستور بعدی از ابتدای سطر بعد آغاز شده است و تورفتگی ندارد. در صورتی که بدنه ی یک بند را به فرم دوم بنویسیم می توانیم از دستورات مرکب تو در تو استفاده کنیم. به طور مثال در قطعه کد زیر می بینید که دستور if دوم در دستور if اول آمده است:

```
if x > y:
    if z > x:
        print("z is the greatest.")
    print("x is greater than y.")
print("We found the greatest.")
```

۳,۴ زبان جاوااسکریپت

یکی از مهم ترین زبان های برنامه نویسی حوزه ی وب محسوب می شود. جاوا اسکریپت اولین بار با اسم LiveScript و توسط شرکت نت اسکپ معرفی شد و بعدها به جاوا اسکریپت تغییر نام داد. جاوا اسکریپت نه از نظر ساختار و نه از نظر مفاهیم شباهتی به زبان جاوا ندارد و این تشابه اسمی، در حد همان کلمات و نام و نشان باقی مانده است.

استاندارد جاوااسکریپت اکمااسکریپت (ECMAScript) است که از سال ۲۰۱۲ تمامی مرورگرهای مدرن استاندارد اکمااسکریپت نسخه ۵,۱ را به صورت کامل پشتیبانی می کنند، همچنین مرورگرهای قدیمی تر نسخه ۳ از اکمااسکریپت را پشتیبانی می کنند. از ماه June سال ۲۰۱۵ اکمااسکریپت ۶ (ES6) یا همان اکمااسکریپت ۲۰۱۵ (ES2015) مورد قبول واقع شده است

به‌طور طبیعی بعد از یادگیری HTML و CSS نوبت یادگیری جاوا اسکریپت می‌رسد. یادگیری جاوا اسکریپت چندان کار سختی نیست؛ ولی نکته مهم در رابطه با یادگیری این زبان، آن است که برخلاف اچ‌تی‌ام‌ال یا سی‌اس‌اس که زبان‌های نشانه‌گذاری محسوب می‌شوند، جاوا اسکریپت یک زبان «برنامه نویسی» است

دستور زبان جاوا اسکریپت:

حساسیت به بزرگی و کوچکی کاراکترها
جاوا اسکریپت زبانی حساس به بزرگی و کوچکی کاراکترهاست. حتماً لازم است که نام سازنده (Constructor) را با حرف بزرگ و نام یک تابع را با یک حرف کوچک شروع کنیم.

فضاهای خالی (فاصله) و سمی کولن
بر خلاف زبان‌هایی مانند C زبان جاوا اسکریپت حساس به فاصله است. با استفاده از سمی کالن گذاری می‌توان خطای ناشی از فاصله گذاری نامناسب را جبران کرد. نمونه‌هایی از خطاهای فاصله گذاری در ادامه رخ داده است:

```
return
a + b;

// Returns undefined. Treated as:
//   return;
//   a + b;
```

اما:

```
a = b + c
(d + e).foo()

// Treated as:
//   a = b + c(d + e).foo();
```

توضیحات

قواعد توضیح گذاری در جاوااسکریپت عیناً مشابه زبان سی ++ است:

```
// a short, one-line comment

/* this is a long, multi-line comment
   about my script. May it one day
   be great. */

/* Comments /* may not be nested */ Syntax error */
```

متغیرها

شناساگرها (Identifiers) در زبان جاوااسکریپت دارای نوع خاصی نیستند و هر نوع مقداری می تواند داخل هر داده ای قرار بگیرد. تعریف داده جدید با استفاده از دستور var انجام می شود. می توان چندین متغیر را به یکباره تعریف کرد. اسم یک متغیر می تواند با یک حرف، با یک underline یا یک علامت دلار \$ شروع شود. کاراکترهای دیگر می تواند اعداد ۰ تا ۹ باشد. با توجه به اینکه جاوااسکریپت حساس به بزرگی-کوچکی حروف است، استفاده از کاراکترهای بزرگ یا کوچک، متغیرهای جدیدی را ایجاد می کند. با شروع جاوااسکریپت نسخه ۱.۵، ISO ۸۸۵۹-۱ یا کاراکترهای یونی کد را می توان در ساختار متغیرها استفاده کرد. در پیاده سازی هایی از جاوااسکریپت از علامت @ می توان به عنوان یک شناساگر استفاده کرد. در حالی که در برخی از ویرایش های جدید از آن چنین امکانی وجود ندارد. مشابه بسیاری از زبان های دیگر، متغیرهای این زبان در محدوده ای که تعریف شده اند دارای اعتبار هستند (یا به اصطلاح lexically scoped).

زمانی که جاوااسکریپت به دنبال شناسایی مقدار یک شناساگر است، به دنبال تعریف آن و مقدار دهی آن در ناحیه مورد نظر می گردد. در صورتی که در آن ناحیه تعریف یا مقداردهی پیدا نکند، به ناحیه بالاتر مراجعه می کند، تا اینکه به ناحیه می رسد که در آن متغیرهای جهانی تعریف می شوند. اگر تعریف متغیر پیدا نشود، جاوااسکریپت یک استثنای (Exception) ReferenceError بر می گرداند. زمانی که یک شناساگر مقداردهی می شود، دوباره همین مراحل تکرار می شود تا نزدیک ترین متغیر پیدا شود و مقداردهی به آن انجام گیرد. در صورتی که متغیری پیدا نشود، متغیری متناسب با نوع مقدار در ناحیه جهانی تعریف می شود. حواستان باشد که جاوااسکریپت در حالت strict در صورتی که متغیری تعریف نشده باشد، در صورت فراخوانی، متغیر جدید

تعریف نمی‌کند.

نوع تعریف نشده

نوع داده‌ی تعریف نشده (`undefined`) مربوط به متغیرهایی است که ایجاد شده اما مقداردهی نشده‌اند. این نوع همچنین در فراخوانی اشیایی که هنوز ایجاد نشده‌اند برگردانده می‌شود. در مورد نوع‌های بولین، مقدار متغیرهایی که تعریف نشده باشند، `false` برگردانده می‌شود. یکی از رفتارهای غیرمترقبه استفاده از نوع تعریف نشده، و عملکرد `false` آن در جایگاه‌های منطقی است.

```
var test; // variable declared but not defined, ...
//... set to value of undefined
var testObj = {};
alert(test); // test variable exists but value not...
//... defined, displays undefined
alert(testObj.myProp); // testObj exists, property does not, ...
//... displays undefined
alert(undefined == null); // unenforced type during check, displays true
alert(undefined === null); // enforce type during check, displays false
```

باید توجه کرد نوع "تعریف نشده" به صورت یک نوع صریح کامل تعریف نشده‌است. لذا استفاده از `(x == undefined)` راه کاملاً درستی برای "تعریف نشده" بودن یک متغیر وجود نیست؛ چرا که در نسخه‌های قبل از ECMAScript ۵ این مجاز است که متغیری به صورت `var undefined = "I'm defined now"`؛ تعریف شود. شیوه‌ای بهتر استفاده کردن از `(typeof x == undefined)` می‌باشد.

```
alert(null == undefined); // unenforced type during check, displays true
alert(null === undefined); // enforce type during check,
```

```
displays false  
alert(typeof null == object'); // true
```

۳.۵ فریم‌ورک جنگو

جنگو یک فریم‌ورک و چهار چوب سطح بالا به زبان پایتون برای وب نرم‌افزاری می‌باشد که باعث سهولت ایجاد وب‌سایت‌هایی دینامیک می‌شود. یک فریم‌ورک سطح بالا بسیاری از موارد برنامه‌نویسی را بصورت خودکار فراهم کرده و در اختیار برنامه‌نویس قرار می‌دهد. همچنین روش‌هایی میانبر و واسط برای اجرای اعمال مختلف را دارا می‌باشد. پس برنامه‌نویس نیازی به نوشتن کدهای اضافی و مشخص را ندارد.

یک وب‌سایت پویا و بسایتی است که شامل کدهای ساده HTML نبوده بلکه بصورت پویا و توسط نرم‌افزار (web application) تگ‌های اچ‌تی‌ام‌ال و در حقیقت صفحات سایت از روی کدها و دستورات نوشته شده توسط برنامه‌نویس ایجاد می‌گردد. برای مثال ممکن است بر اساس درخواست کاربر اطلاعاتی به صورت دینامیک از پایگاه داده دریافت و نمایش داده شود.

ویژگی‌های یک فریم‌ورک خوب عبارتند از :

مکانیسمی برای نگاشت URL درخواست شده به کدهایی که وظیفه پاسخ دادن به آن درخواست را دارند به عبارت دیگر امکان تعیین اینکه کدام کد یا تابع مسئول جواب دادن به آدرس درخواست شده می‌باشد. برای مثال می‌توان از فریم‌ورک خواست تا برای آدرس‌هایی شبیه به `/users/joe/` کدی را اجرا کند که اطلاعات کاربر را نمایش می‌دهد.

تسهیل نمایش، اعتبار سنجی و نمایش مجدد فرم‌های html
فرم‌های اچ‌تی‌ام‌ال یکی از مهمترین روش‌ها برای دریافت اطلاعات از یک کاربر وب می‌باشد. پس یک فریم‌ورک خوب باید دارای روش‌هایی مناسب برای نمایش و کار با این فرم‌ها باشد. (برای مثال امکان مشخص کردن خطاهای رخ داده)

تبدیل اطلاعات وارد شده توسط کاربر به ساختاری مناسب برای مدیریت و استفاده

برای مثال تبدیل اطلاعات وارد شده در یک فرم اچ تی ام ال به ساختاری متناسب با زبان مورد استفاده

امکان جداسازی محتوا از ظاهر نمایشی

در اکثر زبانهای موجود برنامه نویسی کد های اچ تی ام ال و در حقیقت ظاهر یا **template** سایت با کد ها و محتوای سایت آمیخته می گردد که باعث سختی و ایجاد مشکلاتی هنگام تغییرات بعدی و نگهداری می گردد . با استفاده از این روش ظاهر نمایشی سایت بصورت جداگانه در فایلی خاص ذخیره می گردد . اکنون با اعمال تغییر در هر کدام نیازی به ویرایش دیگری نیست و این دو موجودیتی مستقل می باشد .

امکان درونی سازی آسان با لایه اطلاعات ذخیره شده (**storage**)

در حالت معمول هر برنامه نوشته شده با جنگو دارای ۳ لایه مهم می باشد . بخش نمایشی (**template**) ، محتوا یا کد های کنترلی و بخش ذخیره دائمی اطلاعات (برای مثال پایگاه داده) . البته بسته به نوع برنامه و خواست برنامه نویس می توان این لایه ها را ادغام کرده یا نادیده گرفت .

امکان استفاده موثر از سطح بالایی از تجرید و انتزاع

جنگو در موارد زیادی با استفاده از مفهوم انتزاع سهولت بسیاری را فراهم کرده است . برای مثال برای کار با تکنولوژی چون **ftp** یا **http** با یک مفهوم انتزاعی و سطح بالا روبرو خواهید بود که با استفاده از روابط و توابع متعدد برنامه نویسی را بسیار آسان و قدرتمند می کند . البته امکان نفوذ و استفاده از سطوح پایین تر نیز فراهم است .

حذف موارد و مشکلات معمول در برنامه نویسی وب

حذف و رفع موارد و مشکلات معمول در برنامه نویسی سمت وب . برای مثال حذف پسوند فایل از آدرس های وب (**.php** یا **.aspx**)

فریم ورک **django** تمامی موارد فوق را دارا می باشد . این فریم ورک با استفاده از زبان زیبا ، قدرتمند و سطح بالای پایتون نوشته شده است . برای ایجاد وب سایت برنامه نویس کدهای پایتونی می نویسد که امکان استفاده از ویژگی های پایتون و کتابخانه های این زبان را دارد .

۳,۶ فریم ورک بوت استرپ

Bootstrap مجموعه ای از ابزارهای رایگان برای ایجاد صفحات وب و نرم افزارهای تحت وب است که شامل دستورات **HTML**، **CSS** و توابع جاوا اسکریپت جهت تولید و نمایش فرم ها، دکمه ها، تب ها، ستون ها و سایر

المان های مورد نیاز طراحی وب می باشد.

Bootstrap در ابتدا توسط مارک اتو و جاکوب تورنتون و در جهت ایجاد یک چارچوب ظاهری مشخص و یکسان در ابزارهای توییتر طراحی و نوشته شد. قبل از شروع این پروژه نمونه های زیادی با همین رویکرد ایجاد شده بود که همگی با سرنوشتی مشابه و عدم استقبال طراحان وب دنیا مواجه شده بودند. به دلیل وجود مشکلات اساسی در نمونه های دیگر، سازنده اصلی توییتر یا همان مارک اتو تصمیم به ساخت یک سیستم داخلی و قدرتمند برای خود با نام Bootstrap گرفت. در اگوست سال ۲۰۱۱، توییتر Bootstrap را به صورت متن باز و رایگان به عنوان یک فریم ورک طراحی وب سایت در اختیار دیگران قرارداد و در فوریه ۲۰۱۲ به محبوب ترین پروژه در سایت Github تبدیل شده بود.

بوت استرپ از یک سری ماژول های مختلف تشکیل شده است که توسعه سریع تر و آسان تر ظاهر وب سایت را ممکن می سازد.

یکی از مهم ترین ویژگی های آن امکان طراحی واکنش گرا (رِسپانسیو) است که به آسانی وب سایت شمارا در همه دستگاه ها از موبایل، تبلت تا صفحات دسکتاپ به درستی نمایش می دهد.

این فریم ورک از به روز ترین زبان ها مانند HTML5 و CSS3 پشتیبانی می کند.

کدهای Bootstrap، از دو پیش پردازنده مهم css یعنی LESS و SASS استفاده می کند.

ساختار کلی بوت استرپ از یک سری فایل های استایل و اسکریپت تشکیل شده است.

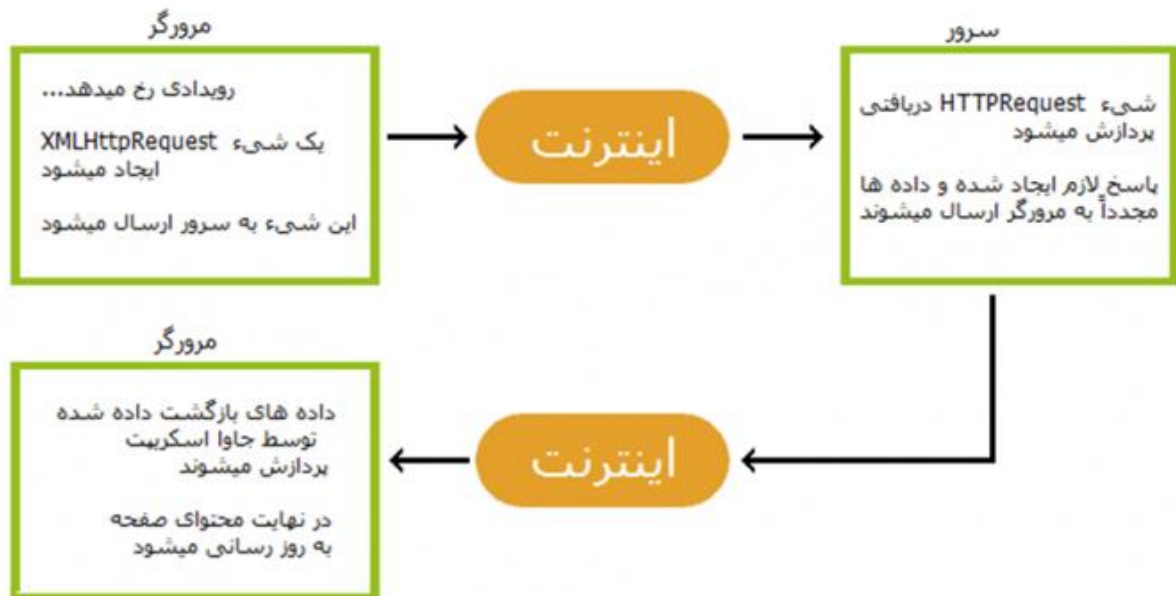
در Bootstrap برای هر امکان کلاسی مشخص وجود دارد؛ مثلاً برای طراحی صفحه واکنش گرا و بخش بندی صفحه (Grid System ۴) کلاس lg-md-sm-xs تعریف شده است. این چهار کلاس مشخص کننده سایز باکس ها در دستگاه های مختلف است.

بوت استرپ ماژولار بوده و توسعه دهندگان می توانند فایل بوت استرپ را به انتخاب خودشان بسازند و حتی به وسیله دستورهای LESS تغییرات عمده تری انجام دهند.

۳,۷ تکنولوژی AJAX

عبارت Ajax به طیف وسیعی از فن آوری های وب اشاره دارد که میتوانند برای پیاده سازی یک برنامه تحت وب بکار روند، فن آوری هایی که در پشت صحنه و در سرور فعالیت می کنند و در نتیجه با وضعیت جاری صفحه وب تداخلی ندارند. بطور کلی می توان گفت فن آوری های HTML (یا XHTML) و CSS مورد استفاده قرار میگیرند

به منظور نمایش محتوا فن آوری Document Object Model (DOM) یا مدل شیئی صفحه)، برای نمایش پویای داده ها و تعامل با آنها XML و JavaScript به منظور یکپارچه سازی این فن آوری ها البته در حال حاضر عمدتاً بجای XML (که فرمتی نسبتاً سنگین است) از JSON برای تبادل داده ها استفاده میشود و حتی امکان استفاده از HTML فرمت بندی شده یا متن ساده نیز به این منظور وجود دارد.



تصویر 2: نمودار تصویری عملکرد AJAX

میتوانیم شکل بالا را در چند مرحله خلاصه کنیم:

- ابتدا یک رویداد client (یعنی یک رویداد در مرورگر) رخ میدهد
- یک شیء از نوع XMLHttpRequest ساخته میشود
- این شیء XMLHttpRequest با اطلاعات لازم پیکربندی میشود
- سپس این شیء یک تقاضای غیر همزمان به سرور وب ارسال میکند
- وب سرور نیز پس از انجام پردازش لازم، نتیجه را که حاوی شیء XML (یا JSON) است برگشت میدهد
- شیء XMLHttpRequest تابع callback (تابعی که پس از انتهای عملیات فراخوانی می شود) را صدا کرده و نتیجه حاصله (شیء XML) را پردازش می کند.

- در نهایت ساختار HTML صفحه به روز رسانی میشود

۳,۸ پایگاه داده Sqlite

SQLite یک برنامه مدیریت بانک اطلاعاتی مبتنی بر زبان استاندارد SQL است. بر خلاف مدل های مرسوم بانک اطلاعاتی که به صورت Client/Server هستند و نیاز به نصب و پیکربندی های خاص خود دارند، SQLite تنها یک برنامه مدیریت بانک اطلاعاتی مستقل است که نیازی به هیچ گونه نصب و پیکربندی ندارد. عبارتی SQLite یک پایگاه داده کوچک (حجمی در حدود ۵۰۰ KB) که به زبان C در قالب یک کتابخانه (Library) نوشته شده است و از پایگاه داده های RDBMS به حساب می آید. این پایگاه داده بدون لیسانس (Public Domain)، آزاد و open source منتشر می شود.

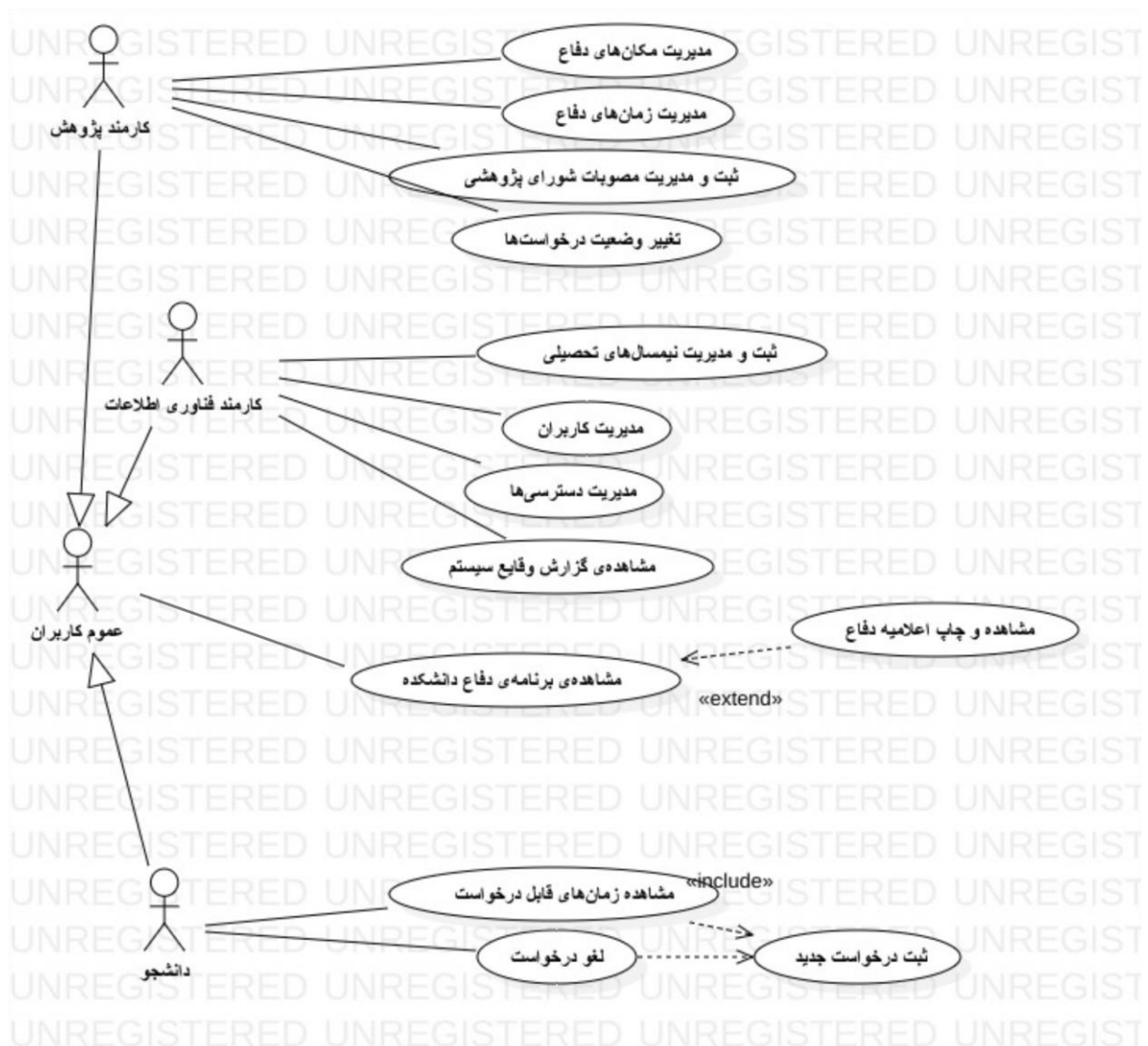
۴ مستندات فنی پروژه

در این فصل به ارائه مدل های بیانگر محصول در سطح کسب و کار مربوطه، در سطح پیاده سازی سیستم و همچنین تشریح عملکرد اجزای سیستم می پردازیم. برای این منظور از دیگرام های متداول تحلیل و طراحی نرم افزار استفاده کرده و همچنین به توضیح قطعات کد سامانه می پردازیم.

۴,۱ مدل سازی سیستم

در این بخش به ارائه نمودارهای سامانه و توضیح آن ها می پردازیم. این نمودارهای شمای کلی سامانه را به ما نشان می دهند.

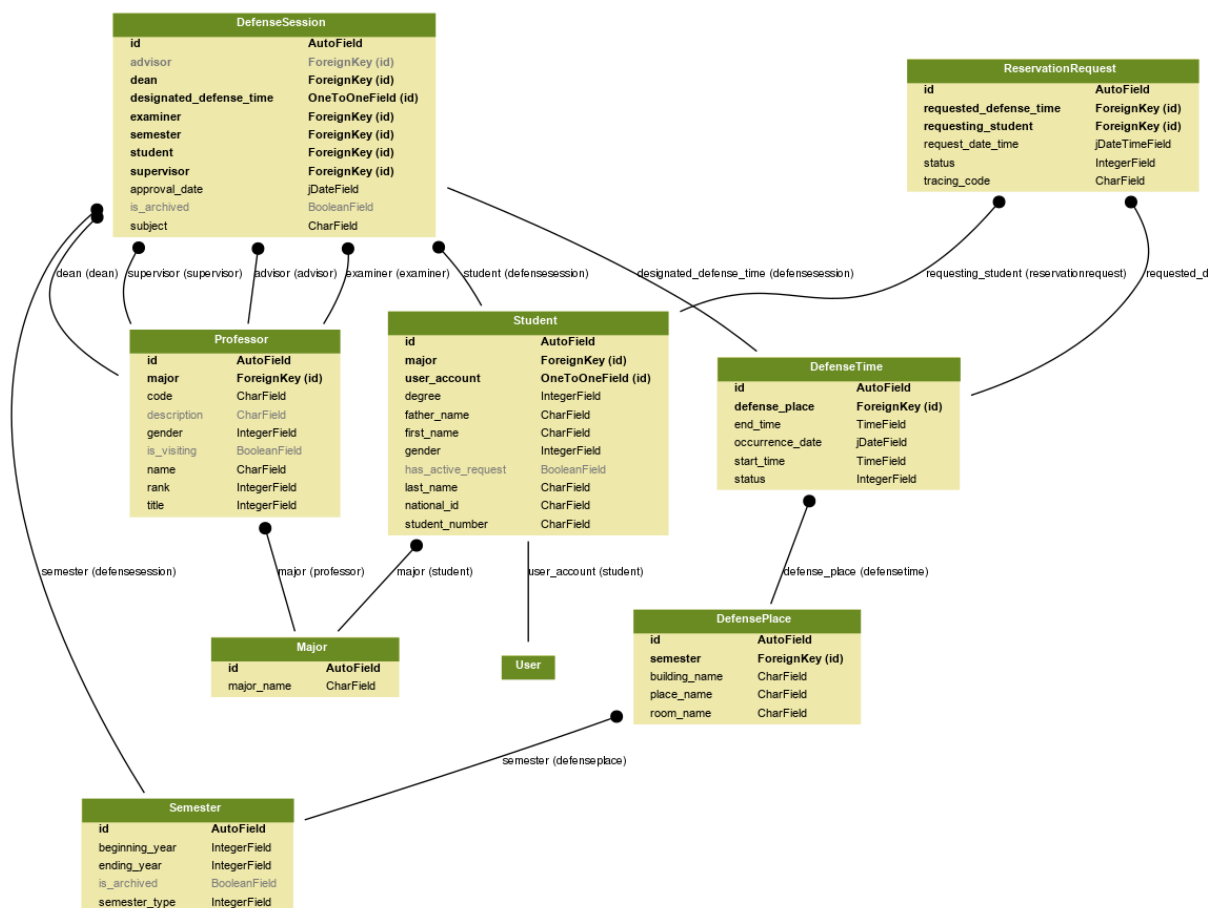
- نمودار موارد کاربرد



تصویر 3 نمودار موارد کاربرد (Use Case)

این نمودار مشخص می‌کند که چه قابلیت‌هایی می‌بایست در سامانه گنجانده شود و چه نوع کاربری باید بتواند به آن دسترسی داشته باشد. همچنین تقدم بعضی از عملیات‌ها نشان داده شده است. بطور مثال همه می‌توانند برنامه‌ی دفاع را مشاهده نمایند و البته به منظور چاپ اعلامیه‌ی مربوط به دفاع مورد نظر لازم است اول مشاهده‌ی برنامه‌ی دفاع (جدول زمان‌بندی شامل مشخصات تمام دفاعیه‌های در دستور کار) اتفاق بیفتد تا بتوان رکورد مورد نظر را چاپ کرد. همچنین ثبت و ربط کاربران تنها از عهده فرد ذی‌صلاح در اداره‌ی فناوری اطلاعات بر می‌آید.

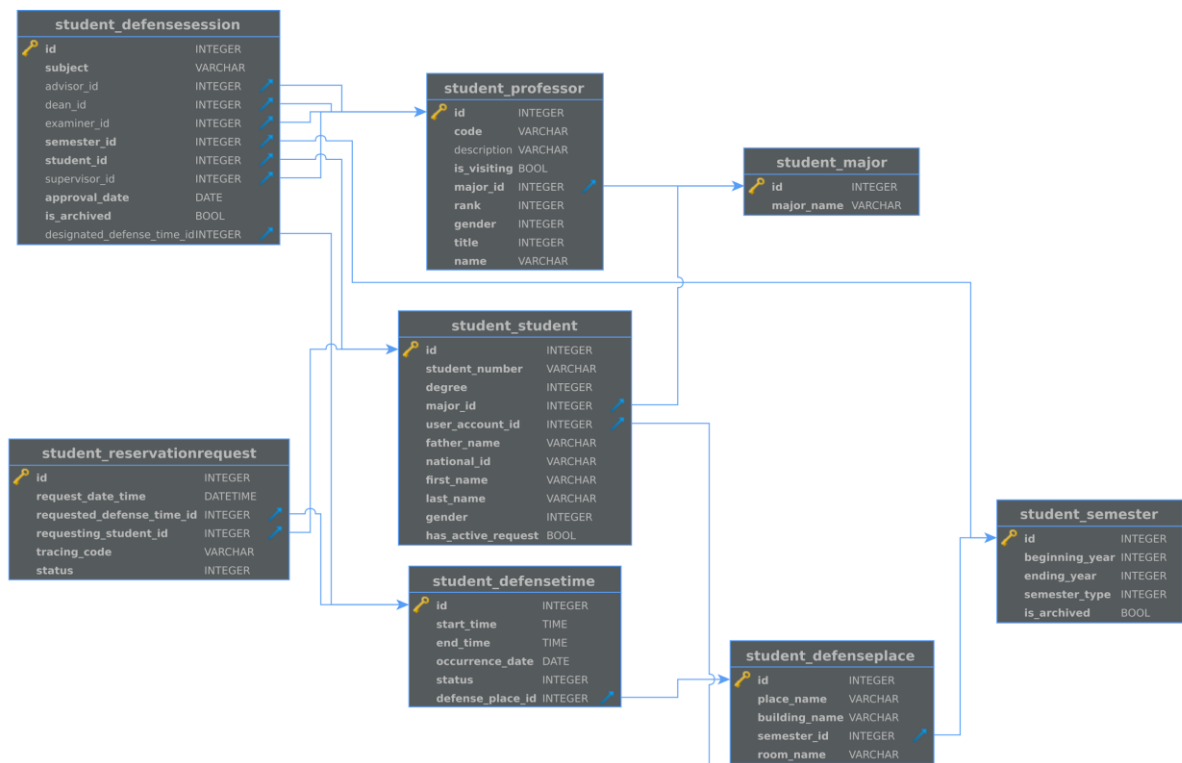
• نمودار کلاس



تصویر 4 نمودار کلاس

البته تصویر فوق دقیقاً یک نمودار کلاس کامل نیست چرا که با وجود بهره‌گیری از شیء‌گرایی در سامانه، هیچ کلاسی به معنای کلاسیک آن موجود نیست و تقریباً تمام منطق‌های آن با استفاده از توابع پیاده شده. نمودار بالا نمایانگر کلاس‌های مدل سامانه است که برای نگاشت کردن جداول در ORM استفاده می‌شوند. در حقیقت این نمودار نمایش دیگری از نمودار ERD منتها در سطح اپلیکیشن است. تمام کلاس‌های فوق از کلاس (models.Model) ارث بری دارند که در این نمودار از نمایش آن صرف نظر شده است.

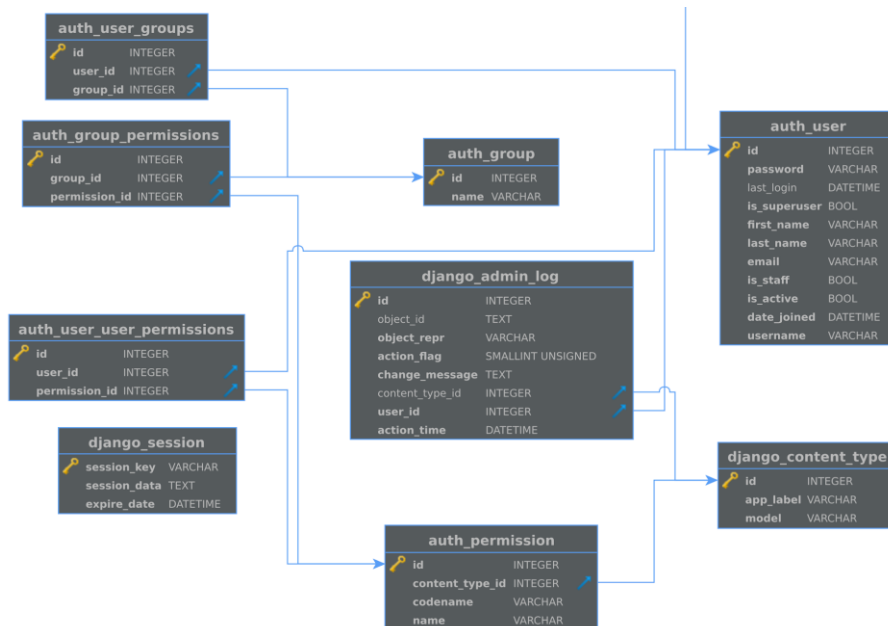
- نمودار موجودیت-بستگی (اصلی)



تصویر 5: دیاگرام موجودیت‌بستگی

این نمودار بیانگر شماتیک جداولی است که ما به واسطه‌ی مدل‌هایی که در قالب تعدادی کلاس در سطح اپلیکیشن به کمک نگاشتگر داده‌ها ساخته ایم و بستگی میان آن‌ها می‌باشد. اگرچه جداول پایگاه داده‌ی ما به این‌ها خلاصه نمی‌شود و شامل تعدادی جدول نیز هست که خود فریم‌ورک برای امور خودش و همچنین برای احراز هویت و تایید مجوز استفاده می‌کند. همانطور که دیده می‌شود این جداول همگی با پیشوند student نام‌گذاری شده‌اند. دلیل این پیشوندگذاری آن است که این موجودیت‌ها مربوط به اپ سامانه‌ی دانشجو می‌باشد که در پروژه از آن با نام «student» نام برده ایم. جدول student_student نیز با موجودیت auth_user که در تصویر زیر آمده بستگی یک به یک دارد.

- نمودار موجودیت بستگی مربوط به فریم‌ورک



تصویر 6 نمودار موجودیت بستگی جداول فریم‌ورک

در این نمودار جداول مربوط به فریم‌ورک جنگو و میان‌افزار auth که امور احراز هویت و اجازه‌سنجی (Authentication & Authorization) را بر عهده دارد قابل مشاهده است. این جداول بدون دخالت ما ایجاد شده، اگرچه می‌توانیم در آن‌ها دست ببریم و به دلخواه خود بهبودشان دهیم یا حتی بخشی از آن‌ها را از جمله auth از سامانه خارج نماییم.

۴.۲ تشریح اجزای سیستم

همانطور که در مقدمه گفته شد، این سامانه یک نرم‌افزار تحت وب است و از الگوی client/server تبعیت می‌کند و به همین دلیل از یک فریم‌ورک سمت سرور و یک فریم‌ورک سمت کلاینت استفاده می‌کند. در این بخش به تشریح ساختار فریم‌ورک سمت سرور (جنگو) می‌پردازیم. در این فریم‌ورک یک سری قوانین و استاندارد وجود دارد که اگرچه در اکثر اوقات می‌توان آن‌ها را رعایت نکرد اما با رعایت آن‌ها کار بسیار آسان شده و سریع‌تر انجام می‌پذیرد.

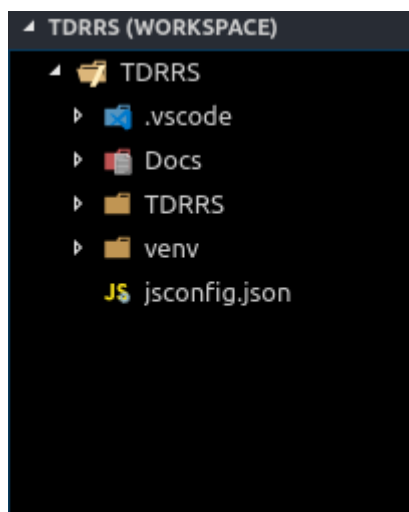
۴.۳ ساختار پروژه‌های جنگو

برای این که بتوانیم ساز و کار این سامانه را مخصوصاً به منظور نگهداری و بارگذاری آن به خوبی درک کنیم، لازم است با ساختار فایل‌های آن آشنا شویم.

• اجزای دایرکتوری پروژه‌ی جنگو

در تصویر زیر دایرکتوری‌ها و فایل‌های گوناگونی به چشم می‌خورد. در محیط توسعه، بالاترین سطح یک پوشه‌ی اصلی داریم که نام آن اهمیتی ندارد (در این جا نام آن با نام پروژه یکسان است) و به طور کلی شامل دو پوشه‌ی مهم می‌شود: ۱- پوشه‌ی محتوی محیط مجازی (virtual environment) و ۲- پوشه‌ی محتوی فایل‌های سامانه. از این پس این پوشه را پوشه‌ی والد می‌نامیم. سایر فایل‌ها و پوشه‌هایی که در ذیل این دایرکتوری مشاهده می‌شود صرفاً در محیط توسعه و فضای کار برنامه‌نویس مورد استفاده قرار گرفته اند و جزئی از پروژه نیستند.

پوشه‌ی محیط مجازی (در این جا venv) در حقیقت بخشی از پروژه نیست بلکه بخشی از محیط اجرای پروژه است و شامل مفسر پایتون، کتابخانه‌های لازم برای اجرای خود پایتون و همچنین پروژه و پکیج‌های مورد نیاز پروژه می‌باشد. از جمله ابزارهایی که حتماً در این پوشه حضور دارد می‌توان به pip اشاره کرد. Pip مدیریت پکیج‌های پایتون را برعهده دارد.



تصویر 7 ساختار دایرکتوری محیط توسعه یا استقرار

پوشه‌ی دیگر که TDRRS نام دارد، دایرکتوری اصلی (ریشه) پروژه می‌باشد. این پوشه به طور متداول و متعارف شامل موارد زیر است:

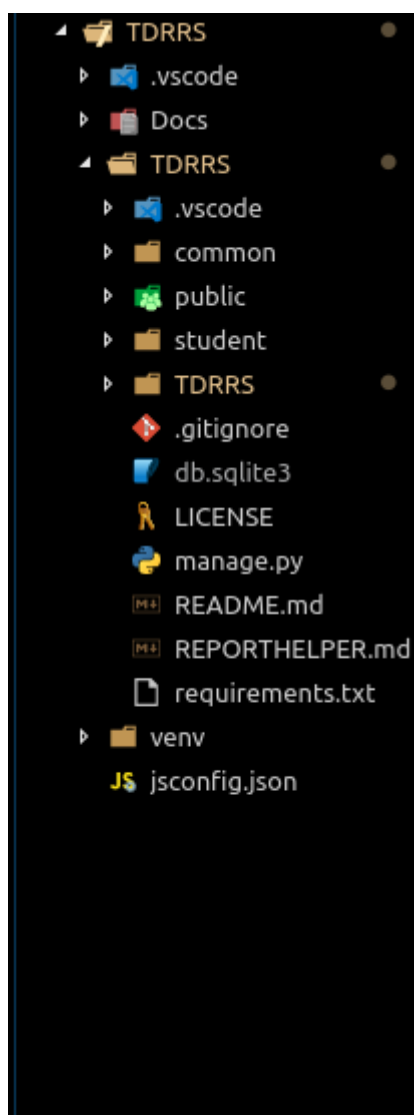
- ۱) دایرکتوری حاوی پیکربندی پروژه همانم با نام پروژه (در اینجا TDRRS)
 - ۲) دایرکتوری مربوط به هر کدام از اپ‌ها (زیر پروژه‌ها)
 - ۳) فایل اجرایی manage.py: این فایل ابزار اصلی ما در محیط توسعه است و اجرا، مهاجرت داده‌ها، ساخت اولین ابرکاربر سامانه و بسیاری دیگر از عملیات سامانه را این اسکریپت برای ما انجام می‌دهد و کافی است آن را با استفاده از مفسر پایتون اجرا کرده و دستورات را به شکل آرگومان به آن پاس بدهیم.
- موارد متعارف دیگر که در این پروژه مشاهده می‌شود عبارتند از:

(۱) فایل requirements.txt: این فایل اطلاعات پکیج‌های مورد‌های مورد استفاده‌ی سامانه را در خود نگه می‌دارد تا در صورت تغییر محیط توسعه یا در هنگام استقرار سامانه به راحتی پکیج‌ها که در حقیقت dependency های سیستم هستند نصب شده و راه‌اندازی شوند.

(۲) فایل .gitignore: این فایل شامل مشخصات فایل‌ها و دایرکتوری‌هایی می‌شود که git نباید تغییرات آن‌ها را دنبال کند.

(۳) فایل LICENSE: این فایل شامل مشخصات لایسنس تجاری و حقوق استفاده کننده از کد می‌باشد

(۴) فایل db.sqlite3: این فایل در حقیقت پایگاه داده‌ی sqlite3 است و تمام جداول و اطلاعات را در خود نگه می‌دارد.



تصویر ۸ ساختار دایرکتوری اصلی پروژه

● ساختار تنظیمات و پیگیری پروژه جنگو

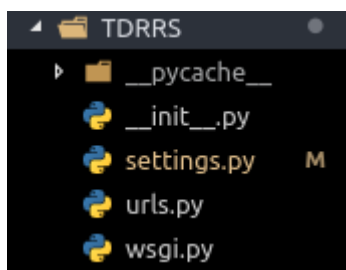
همانطور که گفته شد دایرکتوری هم‌نام که در داخل پوشه‌ی پروژه قرار دارد دربردارنده‌ی پیکربندی سامانه است. محتویات آن در تصویر زیر مشهود است.

تذکر: پوشه‌ی `__pycache__` شامل کد میان‌سطح پایتون است که با یکبار اجرا به منظور اجرای سریع‌تر تولید می‌شود و هیچوقت کاری با آن نداریم. همچنین فایل `__init__.py` صرفاً برای اعلان دایرکتوری حاوی آن به عنوان پکیج پایتون در پوشه حضور دارد. در این جا با ۳ فایل رو به رو هستیم:

(۱) فایل `settings.py` که شامل تمام تنظیمات حیاتی سامانه است

(۲) فایل `urls.py` که نقش `url dispatcher` را دارد.

(۳) فایل `wsgi.py` که برای عملیاتی کردن سامانه و اتصال آن به یک وب سرور عملیاتی استفاده می‌شود.



تصویر 9 ساختار پوشه‌ی مربوط به پیکربندی

حال بررسی فایل `settings.py` خالی از لطف نیست.

```
import os
import locale

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

LOGIN_URL = '/student/login/'

SECRET_KEY = 'clK-@)99e3j5v=85x13%4t)(lSk6pdgnnditiy)g9qz34l9$e0'

DEBUG = True

ALLOWED_HOSTS = []

PREREQ_APPS = [
    'bootstrap_admin',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
```

```

        'django.contrib.staticfiles',
        'django_jalali',
        'django_spaghetti',
        'django_extensions',
    ]

PROJECT_APPS = [
    'student',
    'public',
    'common'
]

INSTALLED_APPS = PREREQ_APPS + PROJECT_APPS

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'TDRRS.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'TDRRS.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

```

```

    }
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'fa-ir'
TIME_ZONE = 'Asia/Tehran'

USE_I18N = True

USE_L10N = True

USE_TZ = True

STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static")
]

SPAGHETTI_SAUCE = {
    'apps': ['student'],
    'show_fields': False,
    'exclude': {'auth': ['user']},
}

```

در این فایل با یک سری متغیر روبه‌رو هستیم که مقادیر مهمی را در خود نگاه می‌دارند. در این جا می‌توان پکیج‌ها، اپ‌ها، زبان سامانه، ساختار ساعت سامانه، پایگاه‌داده‌ی مورد استفاده، وضعیت توسعه و دیباگ سامانه و یک سری مقادیر پیش‌فرض دیگر را اعلان کرد.

● پیکربندی یک «اپ» در جنگو

یک پروژه‌ی متعارف و مبتنی بر اصول در جنگو به این گونه است که باید به ازای هر دسته از قابلیت‌های خود یک اصطلاح `app` داشته باشد. مثلاً یک سامانه‌ی فروشگاهی برای کنترل احراز هویت یک اپ، برای بخش انبار یک اپ، برای مشاهده‌ی کالا یک اپ، برای ثبت سفارش و ویرایش آن یک اپ و برای پرداخت یک اپ دیگر داشته باشد.

پروژه‌ی TDRRS شامل ۳ دایرکتوری اپ با نام‌های «student», «public» و «common» می‌باشد. البته `common` واقعاً یک اپ نیست اما به عنوان یک اپ در سامانه معرفی شده و با آن به عنوان اپ برخورد می‌شود. گفتنی است اپ `student` شامل منطق و مدل‌های مربوط به سامانه‌ی دانشجو می‌باشد و اپ `public` شامل منطق بخش عمومی سامانه که شامل مشاهده‌ی برنامه‌ی زمان‌بندی می‌باشد. دایرکتوری یک اپ از فایل‌های متعددی تشکیل شده است که عبارتند از:

(۱) فایل `admin.py` که شامل یک سری پارامترها و تنظیمات رابط مدیریت می‌باشد و جداولی که باید در بخش مدیریت وبسایت در دسترس قرار بگیرد در این قسمت معرفی می‌کنیم.

(۲) فایل `apps.py` برای تنظیمات و پیکربندی خاص اپ می‌باشد.

(۳) فایل `models.py` که برای تعریف مدل‌ها (در حقیقت موجودیت‌هایی که قرار است در پایگاه داده‌ها به صورت جدول تعریف شوند) به صورت کلاس‌های پایتون استفاده می‌شود.

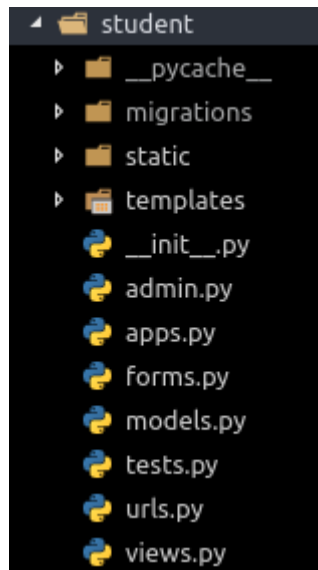
(۴) فایل `tests.py` برای نوشتن آزمون نرم‌افزار (یعنی تعدادی `test case`) خاص این اپ استفاده می‌شود و در متدولوژی TDD بسیار کاربرد دارد.

(۵) فایل `urls.py` در حقیقت `url dispatcher` خاص این اپ است و به این صورت کار می‌کند که `url dispatcher` اصلی آدرسی با ساختار `student/yyyy/yyyy/zzzz/` را دریافت می‌کند و از آن جا که با `student` شروع شده ادامه‌ی پردازش آن را به `url dispatcher` اپ `student` منتقل می‌کند.

(۶) فایل `views.py` شامل توابع (یا کلاس‌هایی) می‌باشد که دربرگیرنده‌ی منطق سامانه هستند و درخواست‌های کاربر را بر اساس آدرس وارد شده پردازش می‌کند. در حقیقت عملکردی مانند `controller` دارند اما با انتزاع بیشتر.

(۷) پوشه‌ی `static` دربردارنده‌ی فایل‌های تصویری، کد `js` و استایل‌های `CSS` می‌باشد.

(۸) پوشه‌ی `templates` دربردارنده‌ی قالب‌های خام `html` است که بعداً توسط `view`ها پردازش شده و به کاربر ارائه می‌شود.



تصویر 10 ساختار یک app

۴,۴ نمونه کدهای سمت سرور

- قطعه کد یک view ساده:

```
from django.shortcuts import render

def get_home(request):
    return render(request, 'public/index.html')
```

این کد به این صورت کار می‌کند که اگر درخواستی به آن بیاید، فایل قالب index.html موجود در اپ public را پردازش کرده و برای کاربر ارسال می‌کند و مرورگر آن را نمایش می‌دهد.

- قطعه کد مربوط به pagination (یا صفحه‌بندی)

```
from django.shortcuts import render
from student.models import *
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger

def get_schedule(request):
    queried_schedules=DefenseSession.objects.filter(is_archived=False)
    page = request.GET.get('page', 1)
    paginator = Paginator(queried_schedules, 5)
    try:
        page_schedules = paginator.page(page)
    except PageNotAnInteger:
        page_schedules = paginator.page(1)
    except EmptyPage:
        page_schedules = paginator.page(paginator.num_pages)
```

```

return
render(request, 'public/defense_schedule.html', {'schedules': page_schedules})

```

این کد به این صورت کار می‌کند که با توجه به درخواست آمده (آبجکت request) یک جستار در پایگاه داده انجام داده و با توجه به مقدار داخل query string صفحه‌بندی را با استفاده از paginator انجام می‌دهد و صفحه‌ی مورد نظر را به کاربر بازمی‌گرداند.

- قطعه کد مربوط به یک نمونه جستار (query)

```

def get_announcement(request, dsid):
    session=DefenseSession.objects.get(pk=dsid)
    return render(request, 'public/announcement.html', {'session':session})

```

این کد به این شکل کار می‌کند که علاوه بر شیء درخواست، یک مقدار عددی نیز از url dispatcher دریافت می‌کند. این مقدار عددی در حقیقت شناسه‌ی جلسه‌ی دفاع است. در خط دوم می‌بینیم که یک جستار با استفاده از ORM جنگو بر روی اشیاء مدل DefenseSession که نگاشتی بر جدول «جلسات دفاع» است می‌زند و رکوردی که کلید اصلی آن برابر با مقدار عددی dsid است را گرفته و در متغیری با نام session می‌ریزد. این متغیر به شکل یک شیء دیکشنری به تابع render پاس داده می‌شود تا در قالب announcement.html به منظور پردازش در دسترس قرار گیرد. بعداً این شیء در بدنه‌ی announcement به زبان DTL پردازش شده و محتویات فیلدهای آن در صفحه‌ی پردازش شده مشاهده خواهد شد.

- محتویات فایل urls.py اصلی پروژه (URL Dispatcher اصلی)

```

from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.get_dashboard, name='dashboard'),
    url(r'^dashboard', views.get_dashboard, name='dashboard'),
    url(r'^login', views.do_login, name='login'),
    url(r'^defense-times', views.defense_times, name='defense_times'),
    url(r'^logout$', views.do_logout, name='logout'),
    url(r'^change-password', views.do_change_password,
name='change_password'),
    url(r'^cancel-reservation', views.do_submit_cancellation,
name='cancel_reservation'),
]

```

این فایل url dispatcher اپ student است و آدرس‌هایی که با فرمت student/yyy/ به سرور بیاید به این فایل منتقل می‌شوند. به طور مثال اگر مرورگر کاربر درخواستی به آدرس زیر بفرستد

۱۲۷.۰.۰.۱:۸۰۰۰/student/defense-times

خواهیم دید که که url چهارم انتخاب شده و کنترل را به views.defense_times می‌دهد.

- migration نمونه کد

```
# -*- coding: utf-8 -*-
# Generated by Django 1.11 on 2018-08-۱۸ ۱۰:۰۴
from __future__ import unicode_literals

from django.db import migrations, models

class Migration(migrations.Migration):

    dependencies = [
        ('student', '0016_auto_20180809_0054'),
    ]

    operations = [
        migrations.AddField(
            model_name='semester',
            name='is_archived',
            field=models.BooleanField(default=False, verbose_name='
            است؟'),
            preserve_default=False,
        ),
    ]
```

این کد که توسط خود جنگو با دستور makemigrations تولید شده است به ORM می‌گوید به جدولی به نام semester ستونی به نام is_archived از جنس boolean اضافه کند و به طور پیش‌فرض مقدار آن ستون را در رکوردهایی که از قبل موجود هستند False قرار دهد.

- قطعه کد با نمونه دکوراتور

```
@login_required
def do_logout(request):
    logout(request)
    return HttpResponseRedirect('/')
```

@login_required یک تزئین‌کننده (decorator) است که منطق پایش ورود به سیستم را به view زیر خود اضافه می‌کند. در حقیقت کنترل می‌کند کسی که این تابع را فراخوانی می‌کند حتماً وارد سیستم شده باشد.


```
from django.db import models
from django.contrib import auth
from django.contrib.auth.models import User
from django_jalali.db import models as jmodels
import jdatetime
# CHOICES
DEGREE_CHOICES=[(1,u'کارشناسی ارشد'), (2,u'دکتری')]
GENDER_CHOICES=[(0,u'سرکار خانم'),(1,u'جناب آقای')]
class Student(models.Model):
    first_name=models.CharField(verbose_name='نام',max_length=30)
    last_name=models.CharField(verbose_name='نام خانوادگی',max_length=30)
    student_number=models.CharField(max_length=12,verbose_name='شماره دانشجویی')
    major=models.ForeignKey('Major',verbose_name='رشته‌ی تحصیلی')
    degree=models.IntegerField(choices=DEGREE_CHOICES,verbose_name='مقطع تحصیلی')
    father_name=models.CharField(verbose_name='نام پدر',max_length=30)
    gender=models.IntegerField(choices=GENDER_CHOICES,verbose_name='جنسیت')
    national_id=models.CharField(verbose_name='شماره ملی',max_length=10)
    has_active_request=models.BooleanField(default=False,verbose_name='درخواست فعال دارد')
    #...
    user_account=models.OneToOneField(User,verbose_name='حساب کاربری',
related_name='student')
    class Meta:
        verbose_name=u'دانشجو'
        verbose_name_plural=u'دانشجویان'
    def get_info_from_user(self):
        return u"{}"
    {}.format(self.user_account.first_name,self.user_account.last_name)
    @property
    def info_from_self(self):
        return u"{} {}".format(self.first_name,self.last_name)
    @property
    def full_name(self):
        return u"{} {}"
    {}.format(self.get_gender_display(),self.first_name,self.last_name)
    def __str__(self):
        return u"{} {} رشته‌ی {}".format(self.full_name,self.get_degree_display(),self.major)
```

این کد مربوط به تعریف موجودیت دانشجو (student) است. در حقیقت یک کلاس پایتون به نام student داریم که از کلاس models.Model ارث‌بری دارد. کلاس Model شامل تمامی منطق‌هایی است که ORM نیاز دارد تا با این مدل کار کند. همچنین ماژول models شامل تعداد کلاس است که هر کدام از آن‌ها متناظر با یک جنس داده در ستون‌های جداول پایگاه داده هستند. به طور مثال first_name که نمونه‌ای کلاس CharField است یک ستون از جنس کاراکتر خواهد بود که با توجه به پارامتر max_length که به آن پاس داده شده است، سائز این ستون در پایگاه داده نیز ۳۰ کاراکتر خواهد بود. ForeignKey نیز مشخص می‌کند که این موجودیت به موجودیت دیگری که نامش را پاس می‌دهیم کلید خارجی (ارتباط یک به چند) دارد.

در ادامه با یک کلاس به نام Meta رو به رو می‌شویم که در درون کلاس Student تعریف شده است. این کلاس دربرگیرنده‌ی اطلاعات اضافی است که در سطح اپلیکیشن به آن‌ها نیاز داریم.

همچنین تعدادی تابع تعریف کرده ایم که کار با این مدل و یک سری منطق ابتدایی را برایمان راحت‌تر کند. به طور مثال تابع `__str__` که یک تابع استاندارد در شیء‌گرایی پایتون است برای تعیین نمود متنی شیء استفاده می‌شود. در حقیقت تعیین می‌کنیم اگر یک نمونه از کلاس Student را print کردیم به جای آن که مشخصات شیء را نمایش دهد یک مقدار به شکل رشته نمایش دهد مثل: علی رضایی دانشجوی کارشناسی ارشد رشته‌ی مهندسی عمران

دکوراتور `@property` باعث می‌شود تا بتوان از تابع به صورت یک فیلد عادی استفاده کرد.

۴,۵ نمونه کدهای سمت کلاینت

- یک نمونه کد قالب پایه

```
{% load staticfiles %}
<!DOCTYPE html>
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width,initial-scale=1,minimum-
scale=1,maximum-scale=7">

<meta name="language" content="en-EN" />
```

```

<meta name="author" content="Ashkan Taravati" />
<title>{% block title %} {% endblock %}</title>

<link rel="stylesheet" type="text/css" href="{% static 'common/styles/main.css'
%}">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
<link href="https://cdn.rawgit.com/rastikerdar/vazir-font/v18.0.1/dist/font-
face.css" rel="stylesheet" type="text/css" />
<!--bootstrap-->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.css"
integrity="sha384-
Smlep5jCw/wG7hdkwQ/Z5nLiefveQRIY9nfy6xoR1uRYBtpZgI6339F5dgvm/e9B"
crossorigin="anonymous">

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js"
integrity="sha384-
o+RDsa0aLu++PJvFqy8fFScvbHFLtbvScb8AjopnFD+iEQ7wo/CG0xlczd+2O/em"
crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+96540ql8C347SDESXU"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js"></
script>
<script src="//code.jquery.com/jquery-1.11.1.min.js" integrity="sha256-VVV0wRhhIghjZP4KXzJenWhhO/hFkWf0SUHt/K/ZpYa"
crossorigin="anonymous"></script>

<link rel="stylesheet" type="text/css" href="{% static 'common/js/main.css'
%}"> {% block stylesheet %}{% endblock %}

</head>

<body>

<header>
<div class="row mrl-0">

```

```

<div class="col-md-12">

    <h4><a class="nav-link">
        سامانه رزرو اتاق دفاع
        
    </a></h4>
</div>
</div>
</header>
<!-- HEADER END-->
<nav class="navbar navbar-expand-lg navbar-light" style="background-
color: #5ca9ff !important;">

    <button class="navbar-toggle rtlDirection text-right d-xl-none d-lg-none" data-
toggle="collapse" data-target=".navHeaderCollapse">
        <i class="fa fa-bars fa-2x" style="color: aliceblue;"></i>
        <span class="icon-bar"> </span>
        <span class="icon-bar"> </span>
        <span class="icon-bar"> </span>
        <span class="icon-bar"> </span>
    </button>
    <div class="collapse navbar-collapse rightToLeft navHeaderCollapse"
id="navbarSupportedContent" style="direction: rtl !important; ">
        <ul class="nav navbar-nav navbar-right text-right">
            <li class="nav-item ">
                </li>
            <li class="nav-item ">
                <a class="nav-link" href="{% url 'home' %}">
                    <i class="fa fa-home"></i> صفحه اصلی
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="{% url 'dashboard' %}">
                    <i class="fa fa-graduation-cap"></i> سامانه دانشجو </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/admin">
                    <i class="fa fa-user"></i>

```

```

        </a> ورود پرسنل
    </li>
    <li class="nav-item">
        <a class="nav-link" href="">
            <i class="fa fa-question"></i> راهنما </a>
        </li>
    </ul>
</div>
</nav>
<!-- LOGO HEADER END-->

<div class="content-wrapper">
    <div class="container col-10" >
        <div class="title">
            {% block heading %}{% endblock heading %}
        </div>
        <div >
            {% block content %}{% endblock %}
        </div>
    </div>
</div>

<!-- CONTENT-WRAPPER SECTION END-->
<footer>
    <div >
        <div class="row mrl-0">
            <div class="col-md-12">
                دانشگاه آزاد اسلامی واحد تهران مرکزی | توسعه یافته توسط گروه مهندسی کامپیوتر
            </div>
        </div>
    </div>
</footer>

<!-- FOOTER SECTION END-->
<!-- JAVASCRIPT AT THE BOTTOM TO REDUCE THE LOADING TIME -->
<!-- CORE JQUERY SCRIPTS -->

```

```

{% comment %} <script src="assets/js/jquery-1.۸.۸.۰۰۰۰۰/۰۰۰۰۰۰۰ {%.۰۰۰۰۰۰۰۰۰۰ %}
<!-- BOOTSTRAP SCRIPTS -->
{% comment %} <script src="assets/js/bootstrap.js"></script> {%
endcomment %}
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.js"></script>
<script src="{% static 'student/main.js' %}"></script>
{% block javascript %}{% endblock %}
</body>

</html>

```

این کد قالب اصلی است که سایر قالب‌ها می‌توانند در درون آن قرار گیرند. بیشتر بخش‌های این قالب مانند یک فایل html عادی است اما در بعضی قسمت‌ها از دستورات DTL استفاده شده است.

• کد مربوط به درخواست AJAX

```

function submitIdAJAX(sender, endpoint) {
    let did = $(sender).data('id');
    $.ajax({
        url: endpoint,
        data: {
            id: did
        },
        dataType: 'JSON',
        method: 'POST',
        success: function (result) {
            console.log('success:' + result.msg);
            ;debugger

            let noticeBox = $('#notice');
            noticeBox.removeClass('alert-danger');
            noticeBox.addClass('alert-info');
            noticeBox.removeClass('d-none');
            noticeBox.html(result.msg);
        },
        error: function () {
            console.log('ERROR:submitReservation failed');
            let noticeBox = $('#notice');
            noticeBox.removeClass('alert-info');
            noticeBox.addClass('alert-danger');
            noticeBox.removeClass('d-none');
        }
    });
}

```

```

        noticeBox.html('<strong>خطا</strong>');
    }
})
}

```

این کد جاوااسکریپت به این شکل کار می‌کند که مقدار صفت `data-id` دکمه‌ی فراخوانده‌ی خود را می‌گیرد و به آدرس مشخص شده (که بیانگر `view` مورد نظر است) با استفاده از توابع موجود در کتابخانه‌ی `jQuery` یک درخواست `POST` ارسال می‌کند و با توجه به نتیجه‌ی درخواست که به شکل یک شیء `JSON` باز می‌گردد، اگر موفقیت‌آمیز باشد باکس پیغام را به رنگ آبی و پیام موفقیت را منعکس کرده و در صورت شکست، باکس را قرمز کرده و پیام خطا را نمایش می‌دهد.

- کد نمونه قالب

```

{% extends "public/public_layout.html" %}
{% block title %}
تیتراصفحه
{% endblock title %}
محتوای آزمایشی
{% block content %}
{% endblock %}

```

این کد یک مثال ساده و آزمایشی است که در صورت پردازش به محتویات فایل `public_layout.html` اضافه می‌شود.

- کد نمونه دستورات `DTL` شبه برنامه‌نویسی در قالب

```

<table class="table table-hover " >
    <thead class=" row-blue-bg">
        <tr class="filters">
            <th >مشخصه</th>
            <th >زمان</th>
            <th >مکان</th>
            <th >موضوع</th>
            <th >رشته</th>

```

```

        <th>توضیحات</th>

        <th></th>
    </tr>
</thead>
<tbody>
    {% for schedule in schedules %}
    {% if schedule.designated_defense_time %}

    <tr>
        {% comment %} <td>{{forloop.counter}}</td> {% endcomment
    %}

        <td>{{schedule.id}}</td>
        <td>{{schedule.designated_defense_time}}</td>
        <td>{{schedule.designated_defense_time.defense_place}}</td>
        <td>{{schedule.subject}}</td>
        <td>{{schedule.major}}</td>
        <td>{{schedule.info}}</td>
        <td><a href="{% url 'defense_announcement' dsid=schedule.id
    %}">مشاهده جزئیات</a></td>

    </tr>
    {% endif %}
    {% endfor %}
</tbody>
</table>

```

در این دستورات یک جدول به صورت پویا ایجاد کرده ایم، به این صورت که به ازای هر یک از عناصر واکنشی شده از جستاری که از view به سمت قالب پاس داده شده، مقادیر موجود در یک آبجکت schedule را که یک رکورد از جدول «جلسات دفاع» است در مکان مناسب خود در میان تگ‌های html قرار داده‌ایم تا به مناسب‌ترین شکل نمایش داده شود. تقریباً مشابه آن است که در یک حلقه‌ی for یکایک عناصر یک آرایه را چاپ کنیم.

۴.۶ روش راه‌اندازی سامانه در محیط توسعه

به منظور راه‌اندازی پروژه در محیط توسعه می‌بایست ابتدا محیط مجازی را فعال کنیم. برای این منظور ابتدا دستور زیر را (با فرض این که در پوشه‌ی والد باشیم) در ترمینال وارد می‌کنیم:

```
source venv/bin/activate
```

این دستور با فعال کردن محیط مجازی، یک سری از متغیرهای محیطی را بازتعریف می‌کند. مثلاً باعث می‌شود سیستم از مفسری که در سیستم عامل موجود است و انبارهای اصلی سیستم استفاده نکند و از موارد موجود در پوشه‌ی محیط مجازی استفاده کند.

در صورتی که اولین بار است سامانه را راه‌اندازی می‌کنیم دو کار لازم است انجام شود.

(۱) نصب پکیج‌های لازم برای اجرای سامانه

برای این منظور دستورات زیر را در ترمینال وارد می‌کنیم

```
cd TDRSS
pip install -r requirements.txt
```

دستور اول دایرکتوری کنونی را به پوشه‌ی پروژه تغییر می‌دهد و دستور دوم فایل requirements.txt را که شامل نام و نسخه‌های مورد نظر پکیج‌های مورد نیاز است به عنوان ورودی به دستور نصب ابزار مدیریت پکیج پایتون می‌دهد تا یکی یکی آن‌ها را دانلود و نصب کند.

(۲) مهاجرت و ساخت پایگاه داده

در این مرحله با وارد کردن دستور زیر، مدل‌ها به شکل جداول در یک پایگاه داده‌ی sqlite ایجاد می‌شوند.

```
python manage.py makemigrations student
python manage.py migrate
```

دستور اول یک سری فایل migration ایجاد می‌کند و دستور دوم یکایک آن فایل‌ها را به ترتیب اجرا و اعمال می‌کند.

حال می‌توانیم سامانه را اجرا کنیم. پس دستور زیر را وارد می‌کنیم:

```
python manage.py runserver
```

این دستور سرور توسعه (Django Development Server) را فعال می‌کند. پیغام موفقیت آن به شکل زیر خواهد بود:

```
Performing system checks...

System check identified no issues (0 silenced).
```

```
August 22, 2018 - 15:38:40
Django version 1.11, using settings 'TDRRS.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

با توجه به این پیام، حال سامانه در آدرس محلی <http://127.0.0.1:8000/> قابل دسترسی است. البته می‌توان صراحتاً پورت دیگری را مشخص کرد اما پورت پیش‌فرض سرور توسعه‌ی جنگو ۸۰۰۰ می‌باشد. توجه داریم که این سرور برای محیط عملیاتی مناسب نیست.

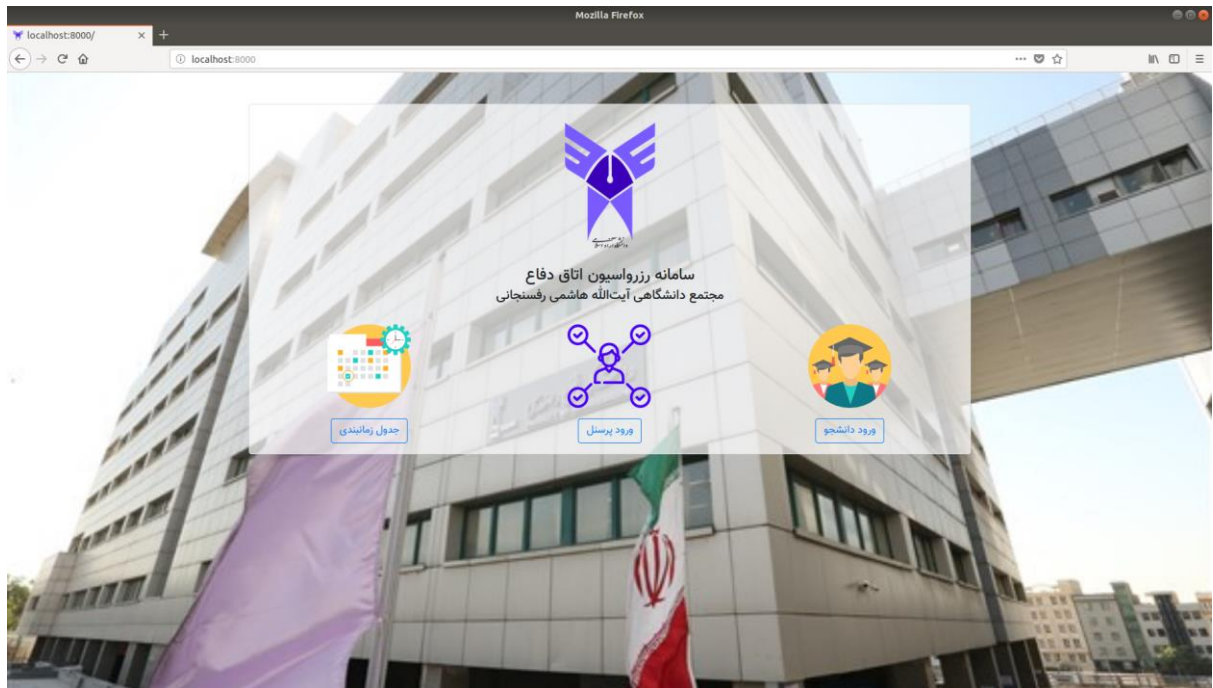
اگرچه سیستم کار می‌کند اما برای ورود به پنل مدیریت لازم است یک ابرکاربر در پایگاه داده وارد کنیم. برای این منظور بار دیگر از `manage.py` استفاده می‌کنیم. ابتدا `Ctrl+C` را فشار داده و دستور زیر را وارد می‌کنیم:

```
python manage.py createsuperuser
Username: ali
Email address: ali@domain.ir
Password:
```

جنگو با پرسیدن اطلاعاتی کاربری با دسترسی کامل برایمان ایجاد می‌کند که بعداً با مراجعه به پنل ادمین می‌توان وارد مدیریت سامانه شد.

۵ راهنمای کاربری سامانه

پس از ورود به سامانه با این صفحه که صفحه اصلی آن می باشد مواجه می شویم:



تصویر 11 صفحه اصلی سامانه

۵.۱ راهنمای بخش جدول زمان بندی

این بخش قابل استفاده ی عموم کاربران می باشد و محدودیتی برای ورود به آن وجود ندارد. برای ورود به این بخش می بایست بر روی دکمه ی «جدول زمان بندی» کلیک نماییم. با ورود به قسمت «جدول زمان بندی» با صفحه ی زیر رو به رو می شویم:

برنامه جلسات دفاع

سامانه رزرو اتاق دفاع

صفحه اصلی سامانه دانشجو ورود پرسنل راهنما

برنامه جلسات دفاع

جستجو کنید

مشخصه	زمان	مکان	موضوع	رشته	توضیحات
2	روز 1397-05-31 از 16:41 تا 18:00	اتاق دفاع شهید باکری- ساختمان دوم فنی مهندسی- ۲۵۴۴	بررسی تاثیر المان‌های گرافیکی سه بعدی در بازخورد کاربر عادی	مهندسی کامپیوتر	دانشجو: نسترن فتوحی، استاد راهنما: مهرشید جوانبخت، استاد مشاور: نیما افشار، استاد داور: بابک وزیری
3	روز 1397-05-21 از 06:00 تا 07:00	اتاق دفاع شهید باکری- ساختمان دوم فنی مهندسی- ۲۵۴۴	تهویه هوشمند حساس به سیالات موجود در محیط	مهندسی مکانیک	دانشجو: علیرضا جوشقانی، استاد راهنما: محمد مهدی متولی، استاد مشاور: ندارد، استاد داور: محسن جهانشاهی

تصویر 12 جدول زمانبندی جلسات

در این صفحه فهرست تمام جلسات دفاعیه‌ی آتی که زمان آن‌ها قطعی شده است قابل مشاهده است. برای مشاهده‌ی جزئیات دفاعیه‌ی مورد نظر میتوانید با کلیک بر روی پیوند «مشاهده‌ی جزئیات» وارد صفحه‌ی جزئیات آن شوید.

جزئیات دفاع | بررسی تاثیر المان‌های گرافیکی سه بعدی در بازخورد کاربر عادی

سامانه رزرو اتاق دفاع

چاپ بازگشت

پایان نامه برای دریافت درجه دکتری

در رشته مهندسی کامپیوتر

عنوان

بررسی تاثیر المان‌های گرافیکی سه بعدی در بازخورد کاربر عادی

ارائه کننده

سرکار خانم نسترن فتوحی دانشجوی دکتری رشته‌ی مهندسی کامپیوتر

اساتید راهنما

سرکار خانم دکتر مهرشید جوانبخت مربی

اساتید مشاور

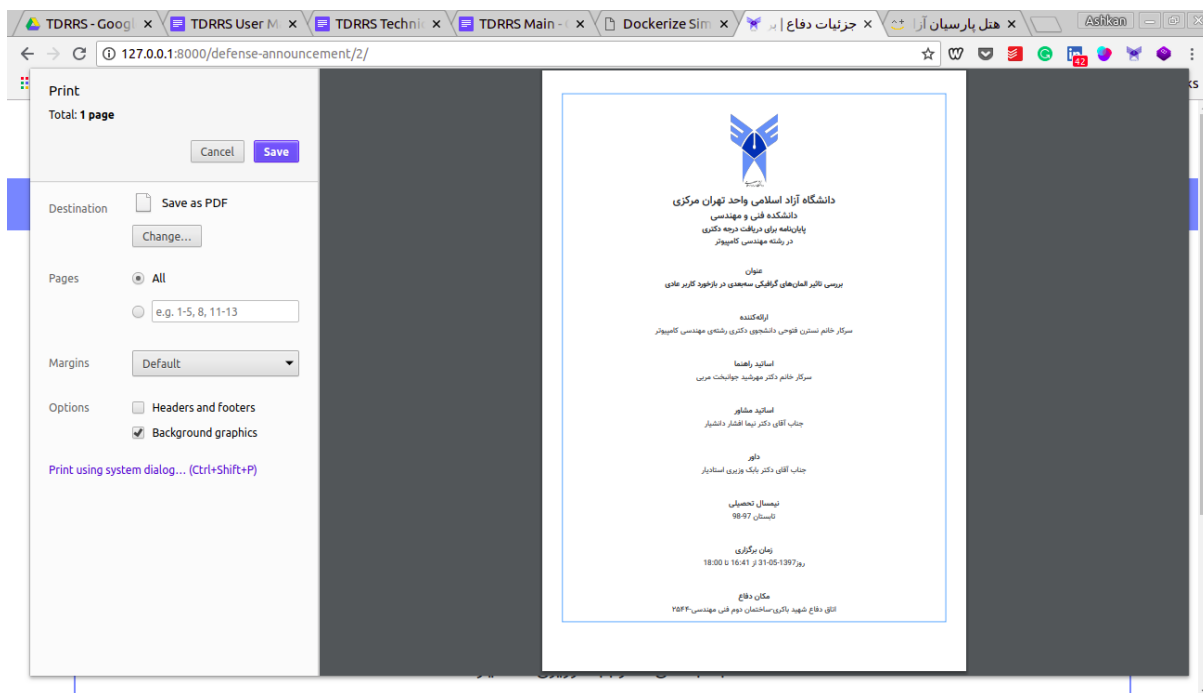
جناب آقای دکتر نیما افشار دانشیار

داور

جناب آقای دکتر بابک وزیری استادیار

تصویر 13 / علامیه‌ی دفاع

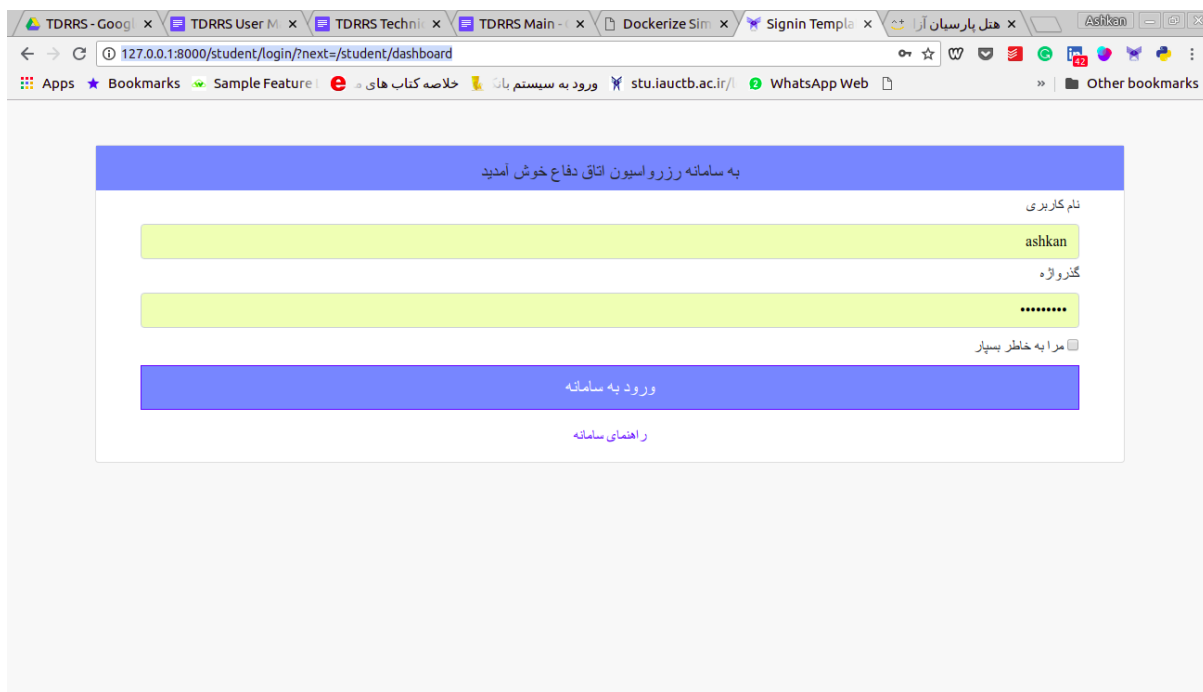
این صفحه دارای قابلیت چاپ می‌باشد به این معنا که با دادن دستور چاپ به مرورگر (Ctrl+P) یا فشردن دکمه‌ی «چاپ» در نوار ابزار بالا، صفحه تبدیل به یک اعلامیه‌ی استاندارد آماده‌ی چاپ می‌شود.



تصویر 14 چاپ اعلامیه

۵.۲ راهنمای بخش دانشجو

با کلیک بر روی دکمه‌ی «ورود دانشجو» از ما خواسته می‌شود تا وارد سیستم شویم.



تصویر 15 صفحه‌ی ورود

با وارد کردن اطلاعات کاربری صحیح به صفحه‌ی داشبورد سامانه‌ی دانشجو هدایت می‌شویم که به شکل زیر است

The screenshot shows a web browser with multiple tabs. The active tab is '127.0.0.1:8000', displaying the 'سامانه رزرو اتاق دفاع' (Defense Room Reservation System) student dashboard. The dashboard has a blue header with navigation links: 'صفحه اصلی' (Home), 'زمان‌های قابل رزرو' (Reservable times), 'جدول زمانبندی' (Timetable), and 'راهنما' (Help). Below the header, there's a status bar indicating 'وقت بخیر جناب آقای طراوتی!' (Good afternoon Mr. Torati!). The main content area is divided into two sections: 'اطلاعات شخصی' (Personal Information) and 'اطلاعات پایان‌نامه' (Thesis Information). The 'اطلاعات شخصی' section contains a table with student details. The 'اطلاعات پایان‌نامه' section contains a table with thesis-related information.

نام	نام خانوادگی	شماره دانشجویی	رشته	کد ملی	نام پدر
سید اشکان	طراوتی	۹۳۰۲۲۵۸۲۰	مهندسی کامپیوتر	۰۰۱۸۵۳۵۲۴۰	سید جمال الدین

موضوع پایان‌نامه	مدیر گروه	استاد مشاور	داور	نیمسال تحصیلی	تاریخ تصویب
سامانه‌های اطلاعاتی با یادگیر BRE	جناب آقای دکتر محسن جهانشاهی دانشیار	ندارد	جناب آقای دکتر بابک وزیری استادیار	تابستان ۹۸-۹۷	۱۳۹۷-۰۱-۲۷

تصویر ۱۶ داشبورد دانشجو

This screenshot shows the same dashboard as before, but with the 'پیگیری درخواست' (Request Tracking) section expanded. It displays a table with a list of requests, including their status, date, time, and a unique code.

زمان درخواست شده	تاریخ ثبت درخواست	کد رهگیری	وضعیت	عملیات
روز 1397-05-21 از 16:38 تا 16:38	01:30 1397-05-11	PvEVvxTqWPqL	لغو از طرف دانشگاه	
روز 1397-04-19 از 14:58 تا 14:58	23:58 1397-05-05	RCI9vtsBqATK	لغو از طرف دانشجو	
روز 1397-05-31 از 16:41 تا 18:00	14:52 1397-05-10	mjE2wryciiYZ	لغو از طرف دانشجو	
روز 1397-05-31 از 16:41 تا 18:00	18:37 1397-05-15	3Qllke8yoYmZ	لغو از طرف دانشجو	
روز 1397-04-19 از 14:58 تا 14:58	18:38 1397-05-15	O3NR4wkzcL1H	لغو از طرف دانشجو	
روز 1397-05-16 از 16:39 تا 16:39	20:50 1397-05-17	BmAxcgSG46VI	لغو از طرف دانشجو	
روز 1397-05-07 از 16:39 تا 16:39	21:04 1397-05-17	eV2hgzo13Jy2	لغو از طرف دانشجو	
روز 1397-05-31 از 16:41 تا 18:00	21:04 1397-05-17	b7M9Pfa8mrhZ	لغو از طرف دانشجو	
روز 1397-04-19 از 14:58 تا 14:58	00:44 1397-05-18	g2Ltkqiy6sgG	لغو از طرف دانشجو	

تصویر ۱۷ داشبورد دانشجو - پیگیری درخواست

این صفحه از سه قسمت تشکیل شده است:

- ۱) مشاهده‌ی اطلاعات هویتی دانشجو
- ۲) مشاهده‌ی مشخصات پروپوزال و دفاعیه‌ی جاری دانشجو

۳) فهرست درخواست‌هایی که دانشجو تا کنون به منظور هماهنگی زمان و مکان تشکیل جلسه دفاع خود ثبت کرده است

در بخش درخواست‌ها می‌توان با کلیک روی دکمه سطل آتش درخواست مربوطه را لغو کرد. البته این در شرایطی است که درخواست پیش‌تر لغو نشده باشد

یکی از بخش‌های دیگر سامانه دانشجو صفحه‌ی «زمان‌های قابل رزرو» می‌باشد در این صفحه فهرستی از زمان-مکان‌ها (یعنی چه زمانی، در چه تاریخی، در کدام اتاق دفاع) که دانشجوی وارد شده با توجه به رشته‌ی و سایر قیود در ترم جاری می‌تواند رزرو کند قابل مشاهده است. دانشجو می‌تواند با فشردن دکمه‌ی ثبت در مقابل آیتم مورد نظر آن را به رزرو خود در بیاورد.

در صورت موفقیت پیام موفقیت در صفحه مشاهده خواهد شد و در صورت صرف نظر همانطور که در بالا گفته شد می‌تواند عمل نماید.

کد	تاریخ برگزاری	روز هفته	ساعت شروع	ساعت پایان	مکان دفاع	عملیات
1	1397-04-19	سه‌شنبه	14:58	14:58	اتاق دفاع اصلی-ساختمان یک فنی مهندسی-۵۱۰	ثبت
5	1397-05-07	یکشنبه	16:39	16:39	اتاق دفاع اصلی-ساختمان یک فنی مهندسی-۵۱۰	ثبت
7	1397-05-16	سه‌شنبه	16:39	16:39	اتاق دفاع شهید باکری-ساختمان دوم فنی مهندسی-۲۵۴۴	ثبت
2	1397-05-31	چهارشنبه	16:41	18:00	اتاق دفاع شهید باکری-ساختمان دوم فنی مهندسی-۲۵۴۴	ثبت

تصویر ۸ صفحه‌ی زمان‌ها قابل رزرو

روز 1397-05-07 از 16:39 تا 16:39	21:04 1397-05-17	eV2hgzt3Jy2	لغو از طرف دانشجو
روز 1397-05-31 از 16:41 تا 18:00	21:04 1397-05-17	b7M9Pfa8mrhZ	لغو از طرف دانشجو
روز 1397-04-19 از 14:58 تا 14:58	00:44 1397-05-18	g2Ltkqiy6sgG	لغو از طرف دانشجو
روز 1397-05-31 از 16:41 تا 18:00	08:25 1397-05-25	PWnNDA7MuN67	لغو از طرف دانشجو
روز 1397-04-19 از 14:58 تا 14:58	19:18 1397-05-31	LTVCSUPvKHen	ثبت شد

تصویر ۱۹ لغو رزرو

۵.۳ راهنمای بخش مدیریت

این بخش امکانات زیادی از قبیل مدیریت کاربران، مدیریت دسترسی‌ها، انجام عملیات CRUD بر روی جداول پایگاه داده‌ها و پایش آخرین رویدادهای سامانه را دارا می‌باشد که مجوز استفاده از همه‌ی این امکانات برای کاربر از نوع کارمند پژوهش توجیه نداشته و مناسب راهبران سامانه در اداره‌ی فناوری اطلاعات می‌باشد. فلذا می‌بایست دسترسی‌های ایشان و حتی سایر نقش‌های کاربری در صورت نیاز به درستی تعیین گردد.

Recent Actions

Apps	عمل
بررسی اصالت و اجازه‌ها	کاربرها
بررسی اصالت و اجازه‌ها	گروه‌ها
دانشپورد دانشجو	اساتید
دانشپورد دانشجو	جلسات دفاع مصوب شورا
دانشپورد دانشجو	دانشجویان
دانشپورد دانشجو	درخواست‌های رزرو
دانشپورد دانشجو	رشته‌های تحصیلی
دانشپورد دانشجو	زمان‌های دفاع
دانشپورد دانشجو	محل‌های دفاع
دانشپورد دانشجو	نیمسال‌های تحصیلی

(Recent Actions) (My Actions)

جلسه دفاع مصوب شورا

تصویر 20 دانشپورد مدیریت

django admin

بررسی اصالت و اجازه‌ها

دانشپورد دانشجو

جلسه دفاع مصوب شورا

22 آرت 2018، ساعت 17:41

نویسنده‌ی هوشمند حساس به سیالات موجود در محیط توسط جناب آقای علیرضا جوشقانی

دانشجوی دکتری رشته‌ی مهندسی مکانیک مصوب 19-04-1397

جلسه دفاع مصوب شورا

22 آرت 2018، ساعت 17:38

بررسی تاثیر آلان‌های گرافیکی ساده‌ی در بازخورد کاربر عادی توسط سرکار خانم نسرن

فخری دانشجوی دکتری رشته‌ی مهندسی کامپیوتر مصوب 01-05-1397

دانشجو

22 آرت 2018، ساعت 17:34

سرکار خانم نسرن فخری دانشجوی دکتری رشته‌ی مهندسی کامپیوتر

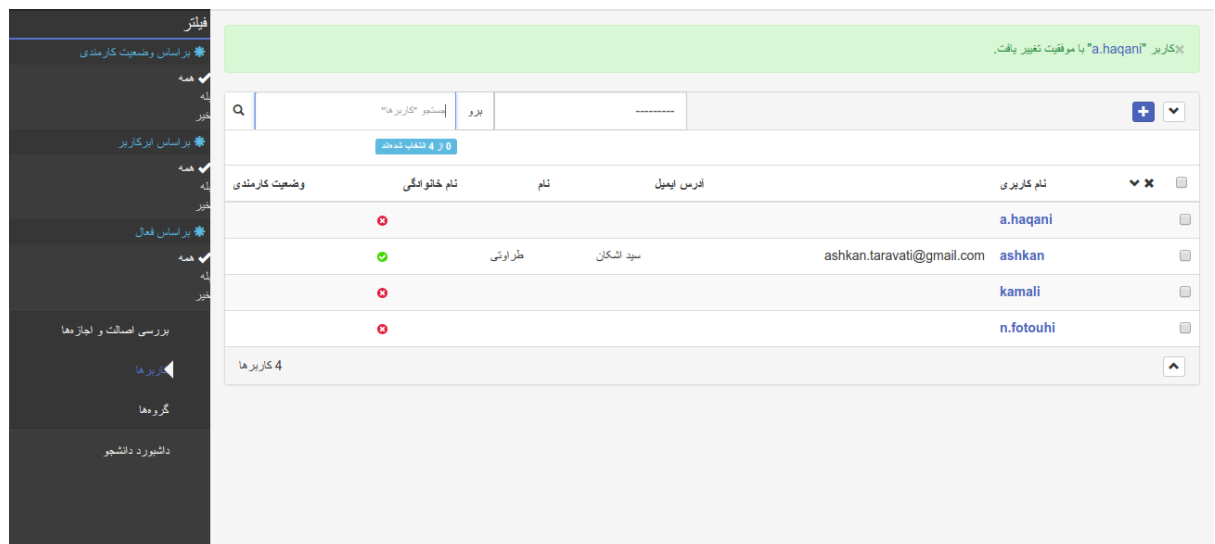
دانشجو

22 آرت 2018، ساعت 17:34

جناب آقای علیرضا جوشقانی دانشجوی دکتری رشته‌ی مهندسی مکانیک

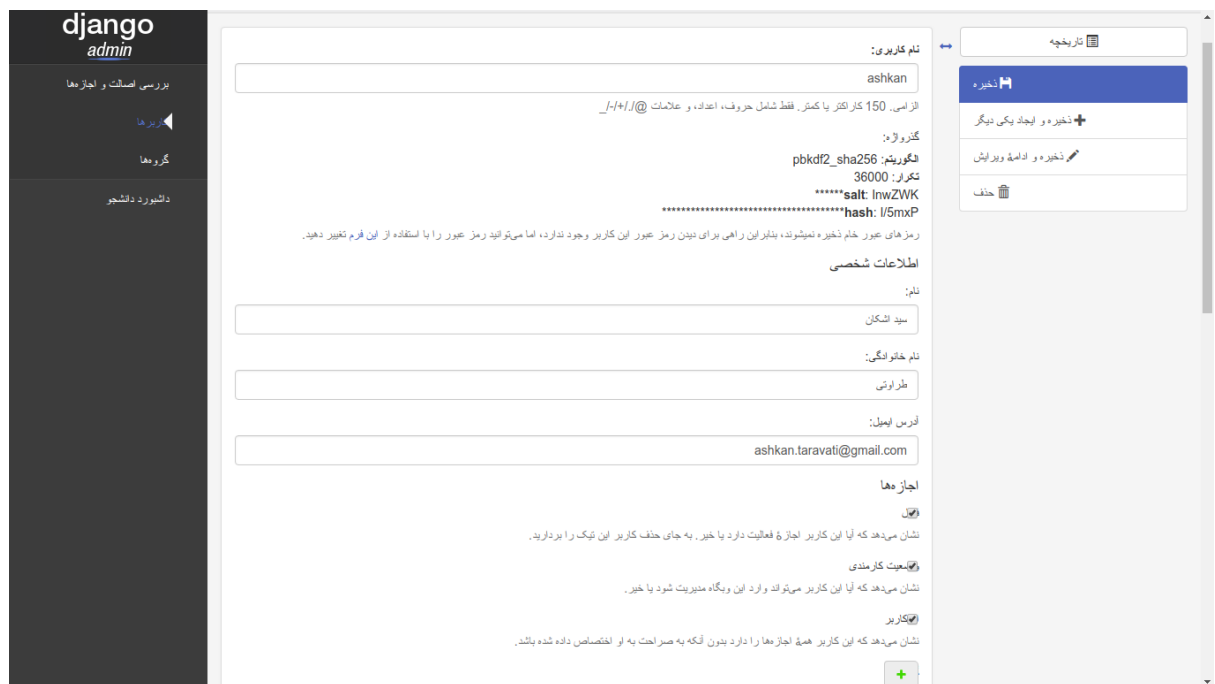
تصویر 21 رویدادهای اخیر

بخش بررسی اصالت و اجازه‌ها شامل دو بخش کاربرها و گروه‌ها می‌باشد در بخش کاربرها می‌توان به مدیریت کاربران سامانه و ویرایش دسترسی‌ها و سایر اطلاعات آن‌ها پرداخت



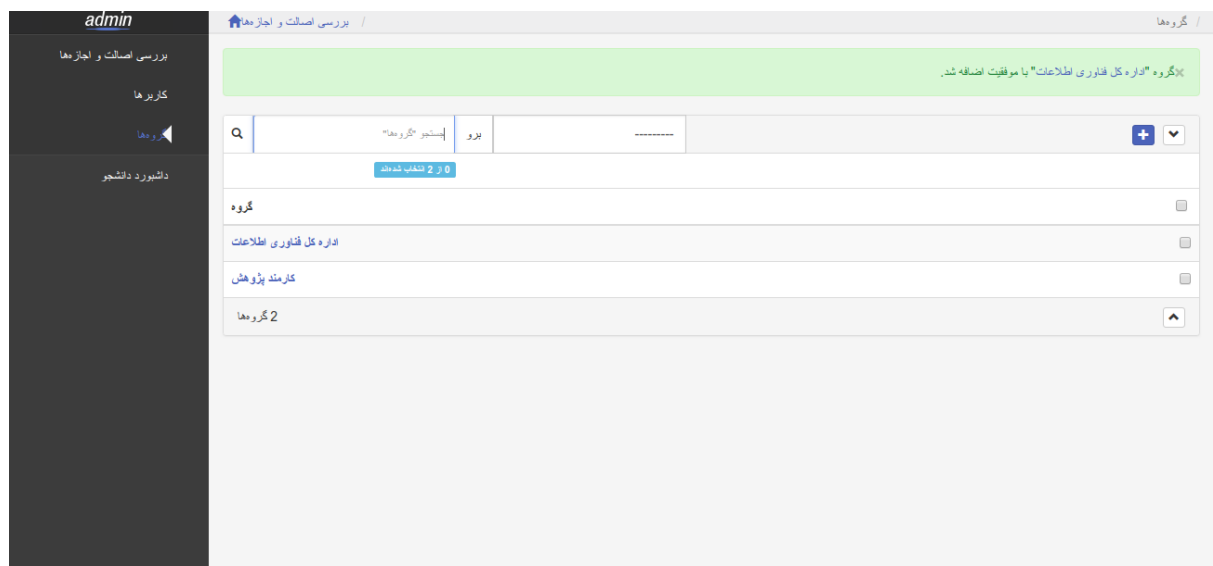
تصویر 22 مدیریت کاربران

به طور مثال صفحه‌ی ویرایش یکی از کاربران به این شکل است:



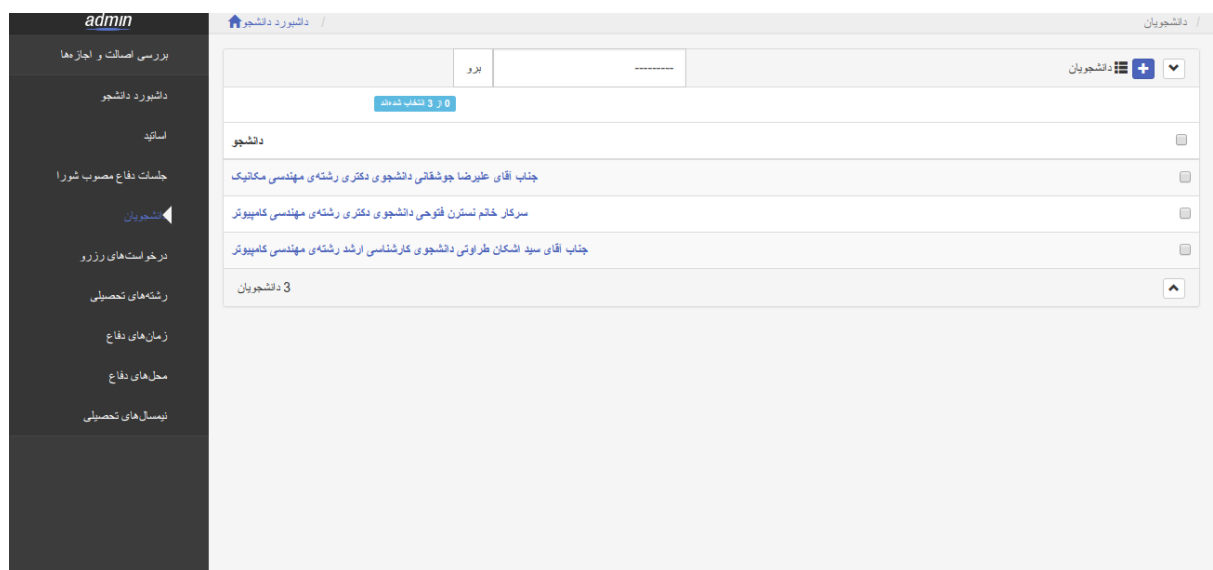
تصویر 23 ویرایش کاربر

در بخش گروه‌ها می‌توان به ازای نقش‌های مختلف در سامانه گروه‌هایی تعریف نمود. هر گروه دارای مجموعه‌ای از دسترسی‌ها (اجازه‌ها) می‌باشد که به راحتی می‌توان با افزودن یک کاربر به یک گروه دسترسی‌های مربوط به آن گروه را به کاربر نسبت داد.



تصویر 24 گروه‌ها (نقش‌ها)

در دسته‌ی آیتم‌های داشبورد دانشجو فهرستی از جداول مربوط به سامانه‌ی دانشجو قابل رؤیت است و کاربر می‌تواند بر حسب دسترسی‌های خود اقدام به ویرایش، ثبت، مشاهده و حذف رکوردهایی از جداول نماید. نمونه‌هایی از صفحات این بخش در ادامه مشاهده می‌شود:



تصویر 25 مدیریت دانشجویان

- مهندسی نرم افزار، راجر اس پرسمن، ترجمه دکتر محمد مهدی سالخورد حقیقی، انتشارات خراسان، ویرایش هفتم، ۷۸۴ صفحه
- کتاب مفاهیم بنیادی پایگاه داده ها، محمد تقی روحانی رانکوهی، انتشارات جلوه، ویراست چهارم، ۸۰۸ صفحه
- UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Martin Fowler, Addison-Wesley, 2004, 175 pages
- The New Relational Database Dictionary: Terms, Concepts, and Examples, By C.J. Date, "O'Reilly Media, Inc.", 2015, ISBN 1491951710, 9781491951712, 450 pages
- Two Scoops of Django 1.11: Best Practices for the Django Web Framework, Daniel Roy Greenfield, 4th edition, Two Scoops Press, 2017, ISBN: 0692915729, 9780692915721, 519 pages
- Computer Networking: A Top-Down Approach, James F. Kurose, Keith W. Ross, 5th edition, Pearson Education, 2010, ISBN 0131365487, 9780131365483, 888 pages

وبسایت های مفید:

- <https://docs.djangoproject.com/en/1.11/>
- <https://readthedocs.org/projects/django/>
- <https://www.mspsoft.com>
- <https://www.djangoproject.com/>
- <https://www.w3schools.com/bootstrap4/>
- <https://whatis.techtarget.com>
- <https://www.javascript.com/>
- <https://www.python.org/>
- <https://getbootstrap.com/>
- <https://searchnetworking.techtarget.com>
- <https://www.tutorialspoint.com>
- <http://www.uml.org/>



درخواست تصویب پروژه کارشناسی

مشخصات دانشجویان

نام و نام خانوادگی	شماره دانشجویی	تعداد واحد گذرانده	رشته - گرایش	امضا
سیدانگن طرابی	۹۳۰۲۵۸۲۰	۱۱۵	مهندسی کامپیوتر - گرایش شبکه های کامپیوتر	

تاریخ شروع پروژه	نیمسال ثبت نام	نیمسال دوم ۹۶-۹۷
۱۳۹۶/۱۲/۲۵		

عنوان پروژه
سازمان رزرو هیئت رئیسه های دفاع پایان نامه ها

شرح مختصری از پروژه

این نام از ار برپستروب به دانشجویان تحصیلات تکمیلی، گروه های آموزشی و دفتر پژوهشی دانشکده احراز می دهد تا در تعامل با یکدیگر به آسانی اتاق دفاع را برای صحنی دفاع و حضور استاد راهنما، استاد مشاور و داور رزرو نمایند.

کاربر این سازمان دانشجویی و پژوهشی می باشد.

استاد پروژه

نام و نام خانوادگی	تاریخ و امضاء

جلسه شورای آموزشی گروه برق در ساعت روز مورخ در محل با حضور هیئت منتخب زیر تشکیل و پس از بررسی پروپوزال فوق رای نهایی به شرح ذیل ارائه گردید:

پروپوزال مورد قبول میباشد ☐ پروپوزال مورد قبول نمی باشد ☐

پروپوزال با اصلاحات مورد قبول میباشد ☐

هیات منتخب:

نام و نام خانوادگی	تاریخ و امضاء	نام و نام خانوادگی	تاریخ و امضاء
1		4	
2		5	
3		6	

تاریخ و امضاء:

مدیر گروه آموزشی: