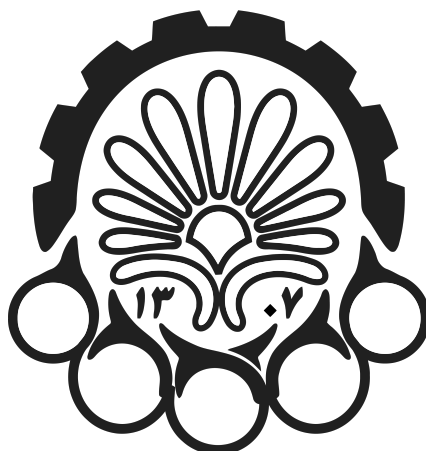


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
درس مدارهای منطقی

پروژه نهایی

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

نیم سال دوم ۰۳ - ۰۲

استاد:

دکتر مرتضی صاحب الزمانی

مهلت ارسال و ارائه:

۱۷ تیر

مسئول پروژه:

مدرسین آزمایشگاه منطقی

طراحان پروژه:

رضا آدینه پور، محمد مهدی نعمتی

فهرست مطالب

۲	نکات قابل توجه
۳	مقدمه
۳	یعقوب برقی که بود؟
۴	اهداف پروژه
۴	توضیحات کلی
۵	حالت مدیر
۷	بخش‌های اصلی
۷	فاز صفر
۸	راه‌اندازی مخزن GitHub
۸	طراحی FSM برای منطق پروژه
۸	فاز اول
۹	بانک
۱۰	کیف پول
۱۱	پشتیبانی
۱۱	مزایده
۱۱	خرید و فروش فایل
۱۲	ارسال خرید
۱۲	توکن
۱۳	بخش‌های امتیازی
۱۳	وضعیت کاربران
۱۳	چت همزمان
۱۳	پایگاه داده
۱۳	REST
۱۴	بانک
۱۴	امنیت
۱۴	Peer-to-Peer



نکات قابل توجه

- ددلاین تحویل فاز اول پروژه، در روزهای شنبه، ۱۹ و دوشنبه، ۲۱ خرداد^۱ در کلاس آزمایشگاه است. همچنین تحویل فاز دوم پروژه، یکشنبه ۱۷ تیر خواهد بود.
- انجام پروژه به صورت گروهی ست و گروه‌های شما همان گروه‌های آزمایشگاه است. در صورت وجود مشکل موجه، با هماهنگی با مدرسین آزمایشگاه می‌توانید گروه خود را تغییر دهید.
- دانشجویان ملزم به تحویل حضوری هر دو فاز پروژه هستند. در صورتی که گروهی در روز تحویل حاضر نشود نمره‌ای به ایشان تعلق نخواهد گرفت.
- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره ایشان برای آن فاز از پروژه "صفر" لحاظ می‌گردد. مشارکت، با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه و سوالات پرسیده شده در روز تحویل و با صلاح دید مدرس مشخص می‌شود.
- ساختار مازولار کد ها حتما باید حفظ شود در صورت عدم رعایت، منفی نمره آن قسمت برای اعضای گروه ثبت خواهد شد.
- ددلاین فاز اول و دوم در آخرین زمان ممکن قبل از تحویل نمرات قرار داده شده است. لذا در هر دو فاز مهلت تاخیر وجود ندارد.
- در صورت کشف تقلب از هریک از تیم‌ها، برای بار اول منفی نمره آن فاز برای آن تیم ثبت می‌شود و برای بار دوم، نمره منفی کل پروژه برای تیم لحاظ خواهد شد که معادل مردود شدن در درس است.

^۱ هر گروه در زمان کلاس خودش

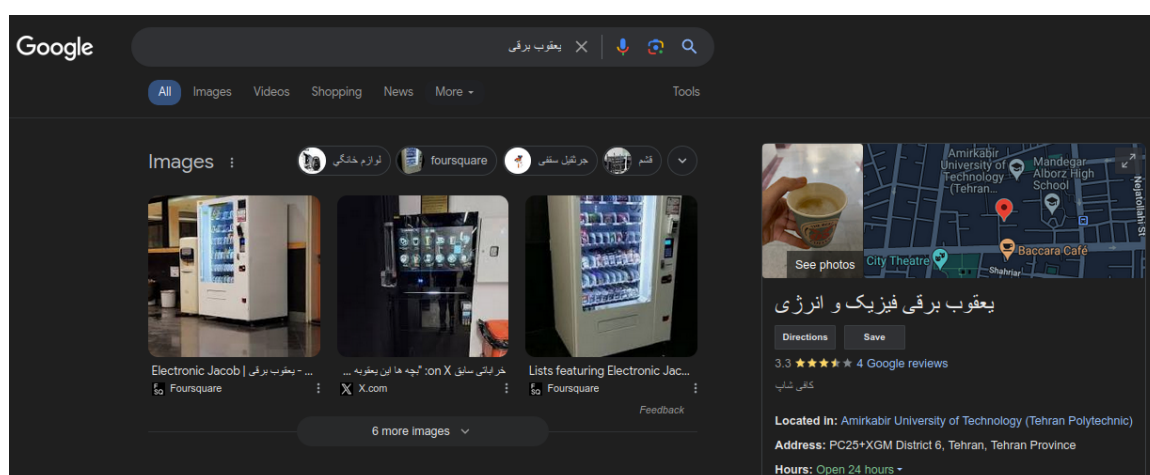


مقدمه

یعقوب^۲ برقی که بود؟

هرامیرکبیری میدونه!

که اسم جهانی یعقوب برقی^۳ مخصوص دستگاه‌های فروش خودکار است که اگر با همین نام اون رو در گوگل سرچ کنید متوجه جهانی بودن این اسم می‌شوید.



شکل ۱: این نام جهانیست!

احتمالا همگی یعقوب برقی دانشکده رو دیدین و از اون استفاده کردین. در این پروژه قصد داریم کمی بیشتر با یعقوب برقی آشنا بشیم و بخش کوچکی از سیستم کنترلی اون رو طراحی، شبیه‌سازی و پیاده‌سازی کنیم. بعد از انجام این پروژه، خریدهای شما از یعقوب برقی، فقط یک خرید ساده نیست!

^۲Electronic Jacob

^۳برگرفته از یعقوب آقا، دکه وسط صحن دانشگاه



اهداف پروژه

- هدف این پروژه، طراحی و شبیه‌سازی سیستم کنترلی دستگاه فروش خودکار است که در فاز صفر آن صرفاً FSM های کلی پروژه طراحی و مخزن^۴ گیت‌هاب پروژه راه اندازی می‌شود.
- در فاز اول، می‌بایست FSM های کشیده شده در فاز صفر تکمیل شوند، طوری که تمام حالات پوشش داده شود و همچنین کد طراحی انجام شده به زبان Verilog نوشته شود و بر روی FPGA پیاده‌سازی شود.
- آشنایی با سیستم مدیریت نسخه Git و کار تیمی بر روی پروژه بر بستر یک مخزن Github، یکی دیگر از اهداف پروژه است. در این مورد توصیه می‌شود تغییرات خود را در دوره‌های کوتاه مدت commit کنید.

توضیحات کلی

پروژه در ۲ حالت^۵ کاری طراحی شده است.

- حالت مدیر^۶
- حالت کاربر^۷

بنابر این پروژه باید دارای ۳ ماژول اصلی `main.v` و `user.v`، `admin.v` باشد. زیر ماژول های هر کدام در ادامه توضیح داده خواهد شد.

Repository^۴
Mode^۵
Admin^۶
User^۷



حالت مدیر (۱۰) نمره

سیستم پس از روشن شدن به صورت پیش فرض در حالت کاربر روشن می شود. با نگه داشتن ۵ ثانیه ای یکی از Push button های موجود بر روی بُرد وارد حالت مدیر می شویم. هر دستگاه پسوردی دارد که تنها مدیر دستگاه با وارد کردن آن می تواند وارد این حالت شود. پسورد دستگاه به صورت زیر در نظر گرفته شود:

Admin_pass: 02

مقدار پسورد باید جایی در حافظه رجیستر شود و در زمان عوض کردن حالت کاری سیستم، پسورد وارد شده توسط شخص با مقدار از قبل ذخیره شده مقایسه شود. شخص می تواند برای وارد کردن پسورد از Dip switch های بُرد استفاده کند. پس از وارد شدن به حالت مدیر، مدیر می تواند اجناس دستگاه را مطابق با شماره خانه های یعقوب برقی دانشکده (شکل «۲» شارژ کند



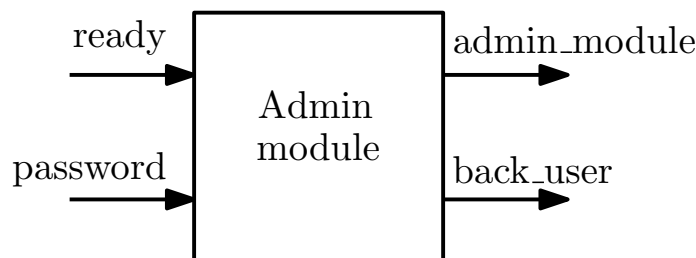
شکل ۲: یعقوب برقی دانشکده



همچنان برای انتخاب جایگاه مورد نظر از Dip switch ها استفاده کنید و برای مشخص کردن تعداد اجناس قرار داده شده در جایگاه انتخاب شده از ۲ Push button بُرد استفاده کنید. یکی از Push button ها بالا شمار و دیگری پایین شمار فرض شود. برای راحتی فرض‌های زیر انجام شود:

- موجودی تمامی جایگاه ها در ابتدای کار صفر است
- هر جایگاه حداکثر ۱۰ ظرفیت دارد

برای خروج از حالت مدیر می‌بایست شماره جایگاه دستگاه را بر روی صفر قرار دهید. در صورت فراموشی مدیر برای خروج از حالت مدیر، برای حفظ امنیت، دستگاه می‌بایست به صورت خودکار پس از ۱ دقیقه از حالت مدیر خارج شده و به حالت کاربر برگردد. ماژول طراحی شده در این قسمت باید به صورت زیر باشد:



شکل ۳: ماژول admin.v

(خروجی‌های admin_mode و back_user حالت کاری سیستم را نشان می‌دهند که باید به دو LED متصل شوند.)

بخش‌های اصلی

فاز صفر (۵ + ۱۰) نمره

در این فاز شما باید مقدمات پروژه را حاضر کنید. این مقدمات شامل ابزارهای مورد استفاده در پروژه و همچنین طراحی FSM های پروژه به صورت کلی است. مراحل این فاز عبارت اند از:



۱. راه اندازی مخزن GitHub

دانشکده مهندسی کامپیوتر

پروژه مدار منطقی

نیمسال ۰۳ - ۰۲

۲. طراحی FSM برای منطق پروژه

در بخش بعد، هر یک از این موارد شرح داده شده اند.



راهنمای مخزن GitHub (۵) نمره

همانطور که می‌دانید برای پروژه لازم است با گروهتان بر روی یک مخزن (repository) گیت فعالیت کنید. برای ساختن این مخزن، کافیست وارد [این لینک](#) شوید. ابتدا با لیستی مواجه می‌شوید که شماره دانشجویی تمام افراد در آن موجود است. شماره دانشجویی خود را بیابید و بر روی آن کلیک کنید. در صفحه‌ی بعد شما باید تیم خود را انتخاب کنید. چنانچه نفر اول گروه خود (سازنده‌ی مخزن) هستید، باید یک تیم بسازید. تنها شماره‌ی گروه پروژه خود را در قسمت نام تیم وارد کنید و تیم را بسازید. نفرات بعدی گروه شما، باید تیم‌شان را از لیست تیم‌های موجود انتخاب کنند و نیازی به ایجاد تیم ندارند. پس از این مراحل مخزن شما آماده خواهد شد و لینک آن در اختیارتان قرار خواهد گرفت. پس از آماده شدن این مخزن، هر یک از اعضای پروژه باید نام و شماره دانشجویی خود را به فایل README.md اضافه کند.

طراحی FSM برای منطق پروژه (۱۰) نمره

مستند فاز اول پروژه به زودی منتشر خواهد شد. برای این بخش، شما لازم است مستند فاز اول پروژه را به طور کامل و دقیق مطالعه نمایید. سپس یک UML مناسب برای تمام بخش‌های منطق پروژه طراحی کنید. فایل ارسالی شما باید یک فایل zip، حاوی تصاویر مربوط به UML باشد. نکته: برای رسم uml می‌توانید از هر ابزار دلخواهی استفاده کنید. مستند آموزشی UML نیز در پیوست این فاز موجود است.

فاز اول

قبل از هر چیز نیاز داریم که نسبت به شبکه و مفاهیم آن شهود بدست بیاوریم. سپس با استفاده از این شهود، ابزارها و کتابخانه‌های جاوا، بر بستر شبکه کد بنویسیم. (البته با توجه به تمرین شبکه، تا حدی با این مفاهیم کار کرده‌اید.)



برای آشنایی با شبکه و کد زدن، **داک شبکه** را مطالعه کنید.
دانشکده مهندسی کامپیوتر پروژه مدار منطقی
در این فاز شما باید فروشگاه خود را به دو بخش کلاینت و سرور تقسیم کنید. سرور شما ابتدا اجرا می شود و یک تا چند کلاینت به آن متصل شده و با آن ارتباط برقرار می کنند (کلاینت و سرور تنها از طریق شبکه با هم در ارتباط خواهند بود). با توجه به تعریف ها و مفاهیمی که از این معماری در داک بالا گفته شد، باید بتوانید کلاس های مربوط سرور و کلاینت را تفکیک کنید.

بانک

در این فاز به پروژه شما یک بانک افزوده می شود. این بانک به صورت یک سرور جدا (از سرور و کلاینت فروشگاه) اجرا می شود. در داک زیر شیوه دقیق ارتباط با بانک، واسطه های کاربری (API) و امکاناتی که در اختیار شما قرار می دهد گفته شده است. فایل اجرایی سرور این بانک بطور آماده به شما داده می شود (برای استخراج کد از این فایل تلاش نکنید)، اما پیاده سازی آن امتیازی است و امتیاز قابل توجهی نیز دارد. در صورتی که می خواهید بانک را خودتان پیاده سازی کنید، باید واسطه های کاربریتان (API) مطابق داک زیر باشد.

فایل اجرایی سرور بانک آماده (به همراه یک نمونه کلاینت) در اختیار شما قرار می گیرد. نحوه استفاده از بانک و API آن را می توانید در **داک بانک** مشاهده کنید.



کیف پول

در این فاز به **فروشگاه** شما، کیف پول اضافه می‌شود. به این شکل که برای هر حساب (فروشنده یا خریدار) یک کیف پول تعریف می‌شود که کاربر می‌تواند با استفاده از حساب بانکی خود (که به صورت جداگانه در بانک قرار دارد) آن را شارژ و یا از کیف پول خود مبلغی را برداشت کند. توجه کنید که خریدار برای خرید کردن باید حق انتخاب داشته باشد که مستقیماً از حساب بانکی خود یا کیف پول خود خرید کند (مدیران فروشگاه حسابی مشترک در بانک تحت عنوان حساب فروشگاه دارند که تمامی اعتبار کیف پول‌های کاربران به درون آن ریخته می‌شود).

۱. خرید مستقیم با استفاده از حساب بانکی: در صورت انتخاب این گزینه، مبلغ پرداختی از حساب کاربر (در بانک) به حساب فروشگاه (در بانک) انتقال می‌یابد (با استفاده از API بانک این کار انجام گیرد) و سپس کیف پول فروشنده با کسر کارمزد، شارژ می‌شود.

■ مدیر باید بتواند کارمزد را تعیین کند. (مثلاً ۵ درصد)

۲. خرید با استفاده از اعتبار کیف پول: در صورت انتخاب این گزینه، کل مبلغ از کیف پول خریدار کم شده و با کسر کارمزد، کیف پول فروشنده شارژ می‌شود.

■ مثلاً اگر کارمزد ۵ درصد باشد، اگر خریدار کالایی با قیمت ۱۰۰ واحد بخرد، از کیف پولش ۱۰۰ واحد کم شده و به کیف پول فروشنده ۹۵ واحد اضافه می‌شود.

هر فروشنده و خریدار باید بتواند بعد از login کردن عملیات زیر را با کیف پول خود انجام دهد:

۱. شارژ کیف پول

۲. برداشت از کیف پول

■ فقط فروشنده می‌تواند این کار را انجام دهد.



پشتیبانی

در فروشگاه باید نقشی تحت عنوان پشتیبان داشته باشیم. ساخت اکانت پشتیبان همانند اکانت **مدیر** باید توسط مدیر انجام گیرد. سپس این افراد به حساب کاربری خود ورود می‌کنند. (login) هر خریدار می‌تواند از بین پشتیبان‌هایی که آنلاین هستند، یک نفر را انتخاب کرده و با او صحبت کند. پیاده سازی چت همزمان یک پشتیبان با چند خریدار، از موارد امتیازی است.

مزایده

یک فروشنده می‌تواند یکی از محصولاتش را برای مزایده بگذارد و زمانی را برای این مزایده تعیین کند که پس از آن، مزایده خاتمه یابد.

۱. باید لیستی از مزایده‌ها برای خریداران وجود داشته باشد.
۲. خریدار با انتخاب یکی از مزایده‌ها وارد آن می‌شود و در ابتدا مبلغی را پیشنهاد می‌دهد که کمتر از شارژ کیف پولش است.
۳. خرید در مزایده تنها با کیف پول انجام می‌شود. (روال انتقال پول مثل خرید عادی است.)
۴. همه‌ی افراد شرکت کننده در مزایده، باید بتوانند با هم صحبت کنند. (چت چندنفره)
۵. خریدار تنها می‌تواند مبلغ پیشنهادی خود را افزایش دهد.

خرید و فروش فایل

محصولی که فروشنده‌ها برای فروش می‌گذارند می‌تواند یک فایل باشد. در این صورت همانند محصولات عادی فروشنده آن را برای فروش می‌گذارد و آن را به سرور آپلود می‌کند، خریدار پس از خرید، آن را از سرور فروشگاه دانلود می‌کند.



ارسال خرید

بعد از خرید کردن توسط خریدار، باید لاگ خرید و آدرس خریدار در حساب مدیران فروشگاه قابل مشاهده باشد. مدیران باید بتوانند خریدهای انجام شده را مشاهده کنند و وضعیت تحویل آنها را به ارسال شده تغییر دهند. خریداران نیز در بخش تاریخچه خرید در حساب خود باید بتوانند وضعیت سفارشهای خود را ببینند (ارسال شده یا در انتظار ارسال).

- اگر محصول خریداری شده فایل بود، همین اطلاعات باید در حساب مدیران و خریداران قابل مشاهده باشد اما دیگر وضعیت ارسال و آدرس خریدار وجود نخواهد داشت.

توکن

در محتوای مربوط به شبکه (داک شبکه)، توکن توضیح داده شد. پیاده سازی توکن برای ارتباط کاربران فروشگاه (با سرور فروشگاه) نیز از موارد اجباری این فاز است.



بخش‌های امتیازی

از این قسمت به بعد وارد بخش‌های امتیازی پروژه می‌شویم. دقت کنید که این بخش‌های هم بسیار آموزنده و کاربردی هستند و هم می‌تواند به نمره پروژه شما به اندازه قابل توجهی اضافه کند. (:

وضعیت کاربران

مدیران فروشگاه بتوانند وضعیت (آنلاین یا آفلاین) بودن کاربران را مشاهده کنند. به این صورت که لیستی از تمام کاربران و وضعیت آن‌ها وجود داشته باشد.

چت همزمان

همان طور که در بخش پشتیبان نیز گفته شد، یک پشتیبان بتواند همزمان با چند خریدار صحبت کند. توجه کنید که باید قسمت گرافیکی این موضوع هم رعایت شود و پشتیبان بتواند به سهولت بین چت‌ها جابجا شود.

پایگاه داده

برای ذخیره اطلاعات حساب‌ها در سرور، می‌توانید از پایگاه داده استفاده کنید (کاری که در برنامه‌های واقعی و عملی به جای ذخیره سازی خام در فایل انجام می‌شود).
برای این منظور در **داک پایگاه داده** مفاهیم مربوط به پایگاه داده و کار با آن در جاوا توضیح داده شد.

REST

برای ارتباط بین کلاینت و سرور، می‌توانید از معماری نرم افزاری REST استفاده کنید. REST به عنوان عضوی از دنیای وب شناخته می‌شود و مزایا و کاربردهای بسیاری دارد.



بانک

از موارد مهم امتیازی، پیاده سازی بانک است. شما می توانید بانک گفته شده را خودتان پیاده سازی کنید. دقت کنید که بانکی که شما پیاده می کنید باید API آن مشابه API بانک آماده ای که ما در اختیارتان قرار می دهیم باشد. (به دستورات گفته شده در **داک بانک** مراجعه کنید)

* در صورت مشابه بودن کد بانک شما به بانکی که ما در اختیارتان قرار می دهیم، برای شما **تقلب** منظور می گردد.

امنیت

از موارد امتیازی دیگر پیاده سازی کنترل های امنیتی و تامین حداقلی امنیت نرم افزار است. برای این منظور محتوایی آماده شده است که بطور کلی مفهوم امنیت را بیان می کند و آن چه ما به عنوان فروشگاه و بانک امن (در صورت پیاده سازی بانک توسط خودتان) از شما می خواهیم را تعریف می کند و همچنین در امن سازی آن به شما کمک خواهد.

* دقت کنید، بانکی را که ما در اختیار شما قرار می دهیم، لزوماً همه ی موارد **داک امنیت** را رعایت نکرده است.

Peer-to-Peer

مورد امتیازی بعدی انتقال P2P فایل پس از خرید در فروشگاه است. به این صورت که اگر خریداری درخواست خرید یک فایل از فروشنده را داشت، به جای استفاده از سرور به عنوان میزبان میانی (middle host)، این فایل مستقیماً از فروشنده به خریدار منتقل شود. برای آشنایی با مفاهیم Peer-To-Peer و پیاده سازی آن به **داک P2P** مراجعه کنید.