

# طراحی و پیاده‌سازی رابط کاربری مدیریت هسته معاملات بورس



دانشجو: امین ادیب‌نیا  
استاد راهنما: دکتر رامتین خسروی  
دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران



## نتایج

پس از آنکه با الگوی معماری Flux آشنا شدیم، ابتدا به پیاده‌سازی دو نمونه اولیه برای مشاهده عملکرد این معماری پرداختیم. نمونه اولیه با استفاده از الگوی معماری Flux و نمونه دوم بدون استفاده از این معماری بود.

پس از آن، معماری اصلی سامانه مدیریت هسته معاملات بورس را با استفاده از الگوی معماری Flux طراحی و پیاده‌سازی کردیم.

ویژگی‌های پیاده‌سازی:

- Responsiveness
- Keyboard Navigation
- ارتباط با سمت سرور با پروتکل REST
- استفاده از Router

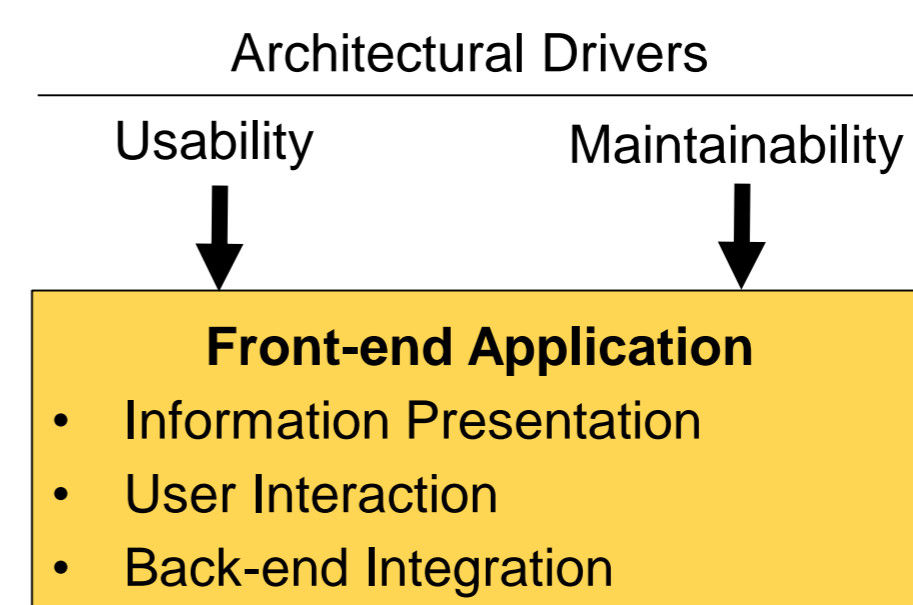
در شکل زیر، نتیجه پیاده‌سازی رابط کاربری مدیریت هسته معاملات بورس را مشاهده می‌کنیم.

نمادها	سفرارها	کارگزاری‌ها	top 5		
نماد	الگویتیم	ISIN	مجموع سهام	قیمت	ویرایش
نماد ۱	بله	۹۴۷۳۲	۱۰۰۰۰	۵%	ویرایش
نماد ۲	بله	۸۲۸۳۷	۲۰۰۰۰	۱۱%	ویرایش
نماد ۳	خیر	۳۸۵۶۱	۲۵۵۰۰۰	۹%	ویرایش
نماد ۴	خیر	۲۹۴۸۶	۳۰۵۰۰۰	۳%	ویرایش
نماد ۵	بله	۹۶۸۱۲	۸۲۳۰۰	۴%	ویرایش

نتیجه پیاده‌سازی رابط کاربری

## مقدمه

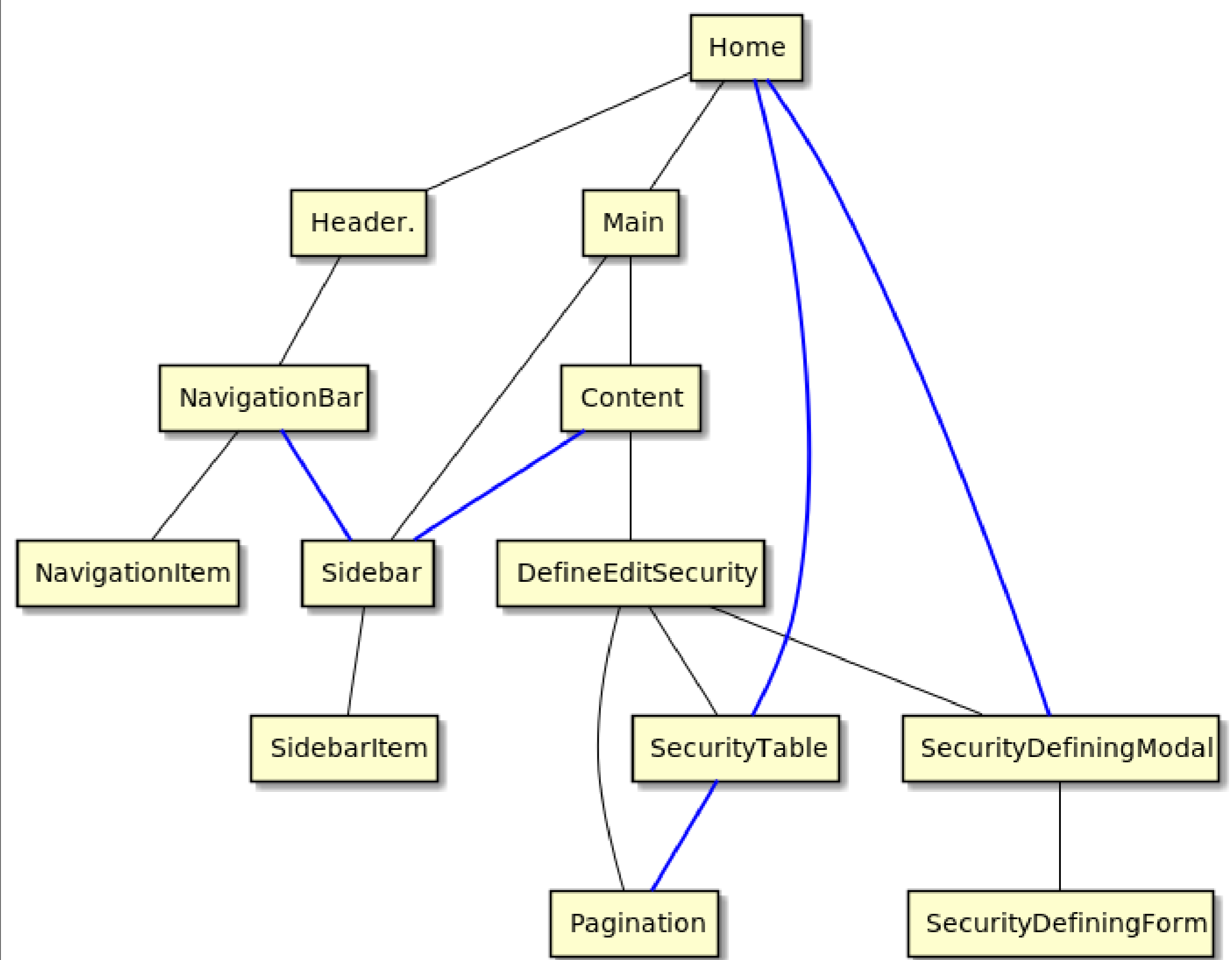
در سمت کاربر نرم‌افزارهای وب، از یک طرف، طراحی ما باید متناسب با مدل کاربری کاربر باشد و پیشران اصلی ما کاربردپذیری است. از طرف دیگر، برای نرم‌افزارهای با رابط کاربری بزرگ و پیچیده، قابلیت نگهداری هم از اهمیت ویژه‌ای برخوردار است. با توجه به این دو نکته، این خود پیچیدگی است که چطور در عین حفظ کاربردپذیری، رابط کاربری ما به گونه‌ای باشد که طراحی مؤلفه‌ای بیشنیه و وابستگی بین مؤلفه‌ها کمینه باشد.



نیازمندی‌های سمت کاربر

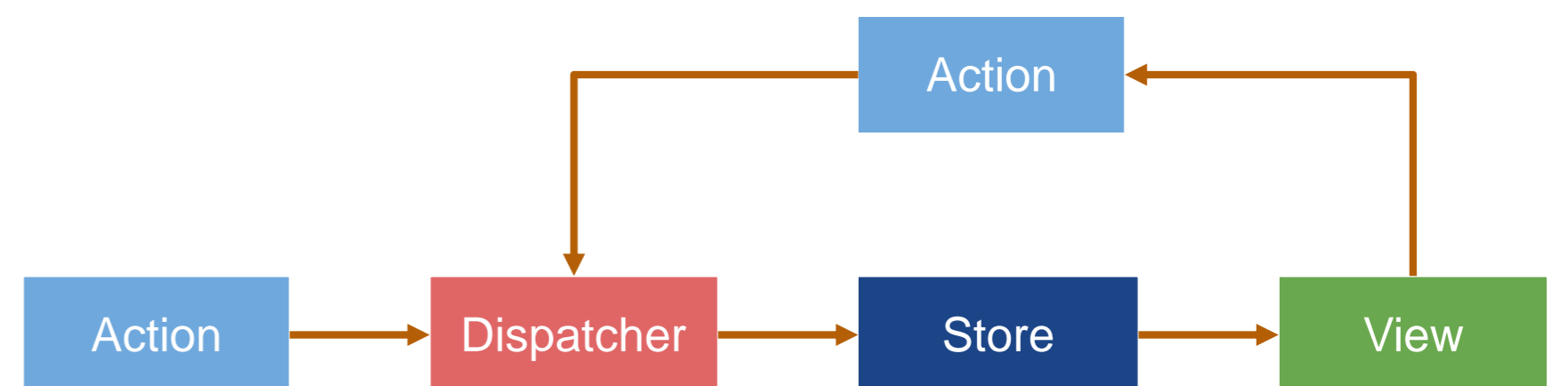
## مدل پیشنهادی

در شکل زیر قسمتی از وابستگی داده‌ای مؤلفه‌ها را مشاهده می‌کنید.



وابستگی داده‌ای مؤلفه‌ها

برای رفع مشکل وابستگی داده‌ای بین مؤلفه‌ها، برای نگهداری حالت برنامه‌ها در سمت کاربر از الگوی معماری Flux استفاده می‌کنیم.



نمای جریان داده بین مؤلفه‌های معماری

- **Action**: متغیرهایی که باعث می‌شوند حالت برنامه عوض شود.
- **Dispatcher**: وظیفه انتقال Action ها را به Store دارد.
- **Store**: محل نگهداری تمام حالت برنامه است.
- **View**: آن چیزی است که کاربر می‌بیند.

## جمع بندی

برای پیاده‌سازی نرم‌افزارهای وب پویا و تعاملی باید در سمت کاربر وضعیت کاربر را در حالت نگه داشت تا با توجه به حالت، محتوای مورد نظر کاربر را به او نمایش داد. اگر صفحات ما از حدی پیچیده‌تر و بزرگ‌تر شوند، نگهداری حالت می‌تواند سخت باشد. باید سعی بر حل این مشکل کنیم.

معماری‌های مختلفی برای حل این مشکل آمده است که از جمله آن‌ها الگوی معماری Flux است. با استفاده از این الگوی معماری می‌توانیم مشکلات مربوط به حالت را تا حد خوبی حل کنیم. این معماری برای آن نوع از پروژه‌هایی که حجم داده زیاد و ساختار پیچیده‌ای در سمت کاربر دارد، مناسب است و قابلیت نگهداری برنامه‌ها را بالا می‌برد.

## مراجع اصلی

1. M. Bertoli, "React Design Patterns and Best Practices," January 13, 2013.
2. R. Unger and C. Chandler, "A Project Guide To UX Design," March 13, 2011.
3. مستندات نیازمندی‌های سامانه هسته معاملات بورس