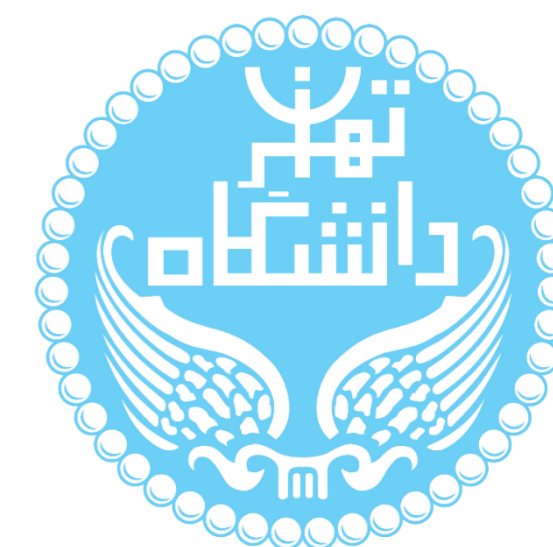


# طراحی و پیاده سازی آزمون خودکار برای نرم افزار ITender در شرکت آریانا



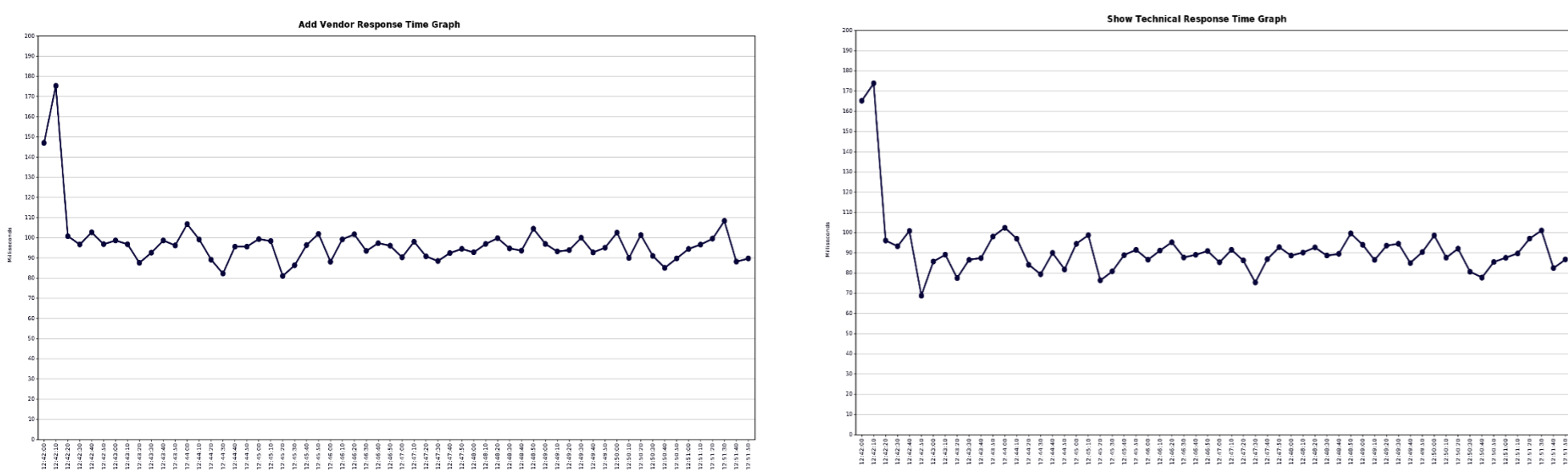
دانشجو: علیرضا شریف پور  
استاد راهنما: دکتر احسان خامس پناه  
دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران



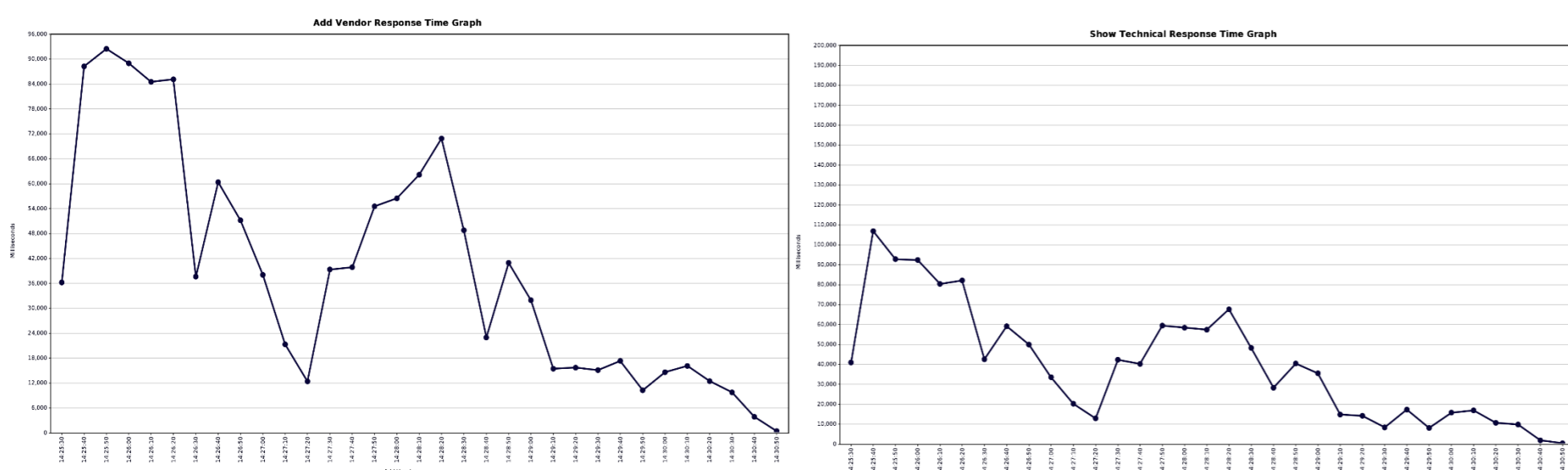
## نتایج

برای یک Controller و یک Model مهم آزمون های واحدی به خوبی پوشش داده شد و برخی الگوهای غلط تصحیح شد.  
همچنین سیستم در حالت پایدار (بار متعادل) کارآمدی قابل قبول دارد اما پس از آزمون استرس (بار زیاد) حد مشخصی از میزان کارآمدی سیستم به دست آمد.

نمونه نتایج در حالت پایدار



نمونه نتایج در حالت استرس



## جمع بندی

در این تحقیق، ابتدا به مفاهیم آزمون واحد و آزمون کارایی پرداخته شد و سپس کتابخانه RSpec و نرم افزار Apache JMeter برای اجرای آزمون ها انتخاب شدند. نتایج آزمون واحد نشان داد که الگوهای کد نیازمند اصلاحاتی برای بهبود قابلیت توسعه و خوانایی هستند. پیشنهاد هایی برای اصلاح الگوی کد شامل استفاده از اصول Clean Coding و ایجاد تست های بیشتر برای پوشش بیشتر کد ارائه شدند. از سوی دیگر، در مورد آزمون کارایی، بهبود عملکرد سیستم از طریق بهینه سازی الگوریتم ها و مدیریت منابع مطرح شد. به طور کلی، اصلاحات الگوی کد و بهبود عملکرد سیستم از جمله پیشنهادات این تحقیق برای بهبود کیفیت نرم افزار بودند.

## مراجع اصلی

- "RSpec Documentation," RelishApp. [Online]. Available: [www.relishapp.com/rspec/](http://www.relishapp.com/rspec/)
- "Apache JMeter Documentation," Apache JMeter. [Online]. Available: [www.jmeter.apache.org/usermanual/index.html](http://www.jmeter.apache.org/usermanual/index.html)
- "Ruby on Rails Documentation," Ruby on Rails. [Online]. Available: [www.rubyonrails.org/documentation](http://www.rubyonrails.org/documentation)

## مقدمه

در این پروژه، ما بر روی پیاده سازی آزمون های واحد و کارایی برای نرم افزار ITender در شرکت آریانا کار کردیم. برای آزمون های واحد از کتابخانه RSpec در زبان Ruby و برای آزمون های کارایی از نرم افزار Apache Jmeter استفاده کردیم. نتایج آزمون های کارایی نشان داد که قواعد و اصولی مانند Clean Code و SOLID رعایت نشده بودند و کد نیاز به بهبود داشت. از تحلیل نتایج آزمون های کارایی نیز برای شناسایی نقاط ضعف نرم افزار و افزایش حد تحمل بار نرم افزار استفاده کردیم. در پایان، به اعمال برخی قواعد و بهبود های کارایی در کد پرداختیم.

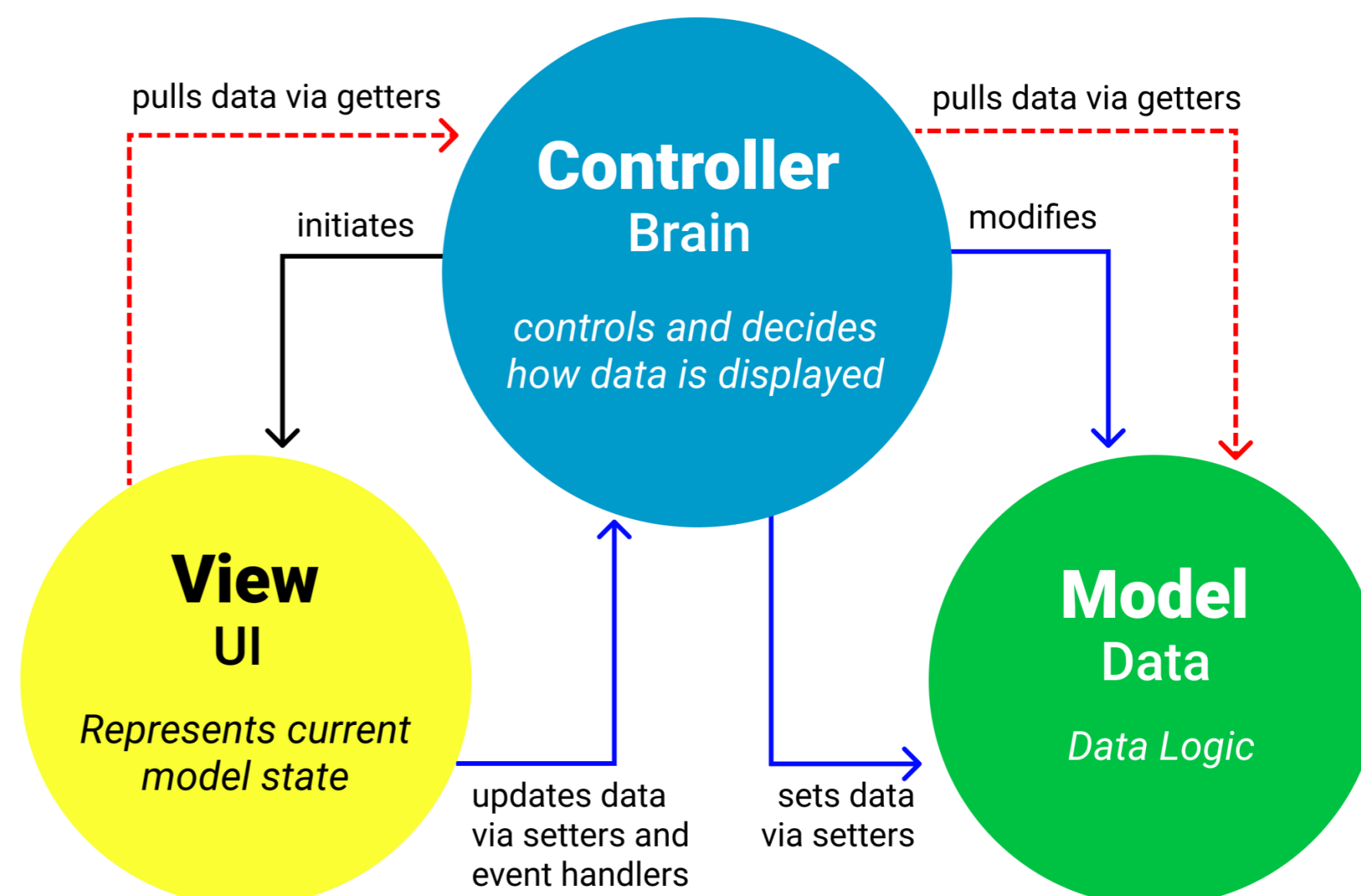
اهداف:

- پیاده سازی آزمون های واحد و کارایی برای نرم افزار ITender
- بهبود کد و افزایش حد تحمل بار نرم افزار

## روش

در پروژه ی ITender از معماری MVC استفاده شده است:

### MVC Architecture Pattern



از کتابخانه هایی چون RSpec, ShouldaMatchers و FactoryBot برای نوشتن آزمون های واحد و بررسی درستی ها و از نرم افزار Apache JMeter برای دستیابی به نتایج قابل بررسی درباره مشخصه های کارایی سیستم استفاده شد.

استفاده از Mocks و Stubs برای جدا کردن وابستگی ها، از ابزارهایی مانند Database Cleaner برای مدیریت پایگاه داده، و تقسیم مسئولیت بخش های مختلف آزمون های واحد به اجزاء مشخص به منظور سازماندهی بهتر کد آزمون ها به اتمام رسید.

