# سند طراحی سیستم

# معرفی

## هدف سیستم

هدف از سامانه کلید آسمان این میباشد که بروکراسی و کاغذبازی را در مسابقه سفیران مسجد به حداقل برساند.برای اینکه هدف به صورت شفاف بیان شود به شرح مختصری در مورد مسابقه سفیران مسجد میپردازیم.مسابقه سفیران مسجد در مساجد و مدارس برگزار میگردد و این مسابقه توسط مسئول فرهنگی مساجد و مربی پرورشی مدارس و با همکاری والدین و نوجوانان برگزار میگردد.هدف از این مسابقه گرایش بیشتر نوجوانان به دین اسلام و آشنایی آنان با مفاهیم دینی میباشد.هم چنین در نظر داریم که این سامانه به گونه ای عمل کند که مسئولین فرهنگی مساجد و مربیان پرورشی به صورتی ساده و راحت و به دور از هرگونه اخلال با این سامانه در تماس باشند.

## اهداف طراحی

هدف از طراحی این سامانه یک User Interface کاربرپسند میباشد.در ضمن سامانه به گونه ای طراحی خواهد شد که کاربر با کم‌ترین سردرگمی با سامانه ارتباط برقرار کند.لازمه نیل به این هدف این میباشد که سامانه از یک راهنمای خوب بهره مند باشد.این راهنما باید به گونه ای طراحی شود که کاربر بتواند به راحتی از این راهنما بهره‌مند شود.از این نظر راهنمای کاربران به صورت اختصاصی شده طراحی میگردد.

## تعاریف، مخفف ها و علامت های اختصاری

کلید آسمان:سامانه فرهنگی و مذهبی است که مسابقه سفیران مسجد را به صورت آنلاین شبیه سازی میکند.

## منابع

1-مساجد شهر تهران(مسئولین فرهنگی مساجد)

2-مدارس شهر تهران(مربیان پرورشی)

3-مصاحبه با نوجوانان و والدین آنها

4-متصدیان برگزاری مسابقه سفیران مسجد

## مرور

در دید کلی در نظر داریم در راستای ارتقای فرهنگ عمومی و اسلامی نوجوانان به طراحی و پیاده‌سازی سامانه کلید آسمان بپردازیم.به طوریکه تیم کلید آسمان برای نیل به این هدف تمام همت و تلاش خود را به عمل می‌آورد.بدیهی است در این راه از تمامی امکانات موجود بهره خواهیم گرفت.امید است در پایان این راه این سامانه به طور کامل طراحی و پیاده سازی شود و این سامانه در اختیار مسئولین فرهنگی قرار گیرد و به نحو احسنت از آن استفاده گردد.

# معماری نرم افزار فعلی

با توجه به اینکه سامانه کلید آسمان نمونه مشابه ندارد لذا مهندسی مجدد در این زمینه دخیل نمی باشد.

# معماری نرم افزار پیشنهادی

## نمای کلی

در این پروژه از معماری سه لایه که حالت خاصی از معماری چند لایه می‌باشد، استفاده شده است. در این قسمت به شرحی از معماری 3 لایه می‌پردازیم:

این معماری به 3 لایه جداگانه تقسیم می‌شود. این لایه‌ها عبارت‌اند از:

**1.Presentation Layer**

**2.Business Logic Layer**

**3.Data Access Layer**

لایه Presentation چیزی نیست به جز بخشی از نرم‌افزار که با کاربر برنامه شما ارتباط برقرار می‌کند (واسط برنامه شماست )نمایش داده‌ها به کاربر نهایی و اجازه به آنان برای ارتباط داشتن با داده‌ها، اصلی‌ترین وظیفه این لایه است. در واقع هر آنچه را که کاربر نهایی استفاده‌کننده از سیستم مشاهده می‌کند از قبیل فرم‌ها، کنترل‌های روی فرم‌ها، تصاویر، منوهای برنامه و... در این لایه قرار می‌گیرند. کاربر سیستم فقط با این لایه در ارتباط بوده و هیچ ارتباطی با دیگر لایه‌ها ندارد و در واقع درخواست خود را از طریق لایه نمایش به لایه‌های زیرین انتقال می‌دهد. وظیفه لایه نمایش این است که اطلاعات لازم را از کاربر گرفته و در صورت لزوم با اعتبارسنجی‌هایی که باید در این لایه انجام گیرد مثل: کنترل طول فیلدها، کنترل اجباری بودن بعضی فیلدها و... را انجام می‌دهد و این اطلاعات را برای هر گونه پردازش لازم دیگری به لایه بعدی ارسال می‌نماید و در واقع هیچ اثری از منطق اصلی برنامه و اتصال به بانک اطلاعاتی در این لایه دیده نمی‌شود. در بیشتر موارد داده‌هایی که توسط کاربر وارد می‌شوند نیاز به اعتبارسنجی یا پردازش اضافی دارند. این مسئولیت لایه Business Logic است. در واقع کلیه درخواست‌هایی که در اثر تعامل کاربر با لایه نمایش ایجاد شده است به این لایه منتقل‌شده و تمام پردازش‌های لازم بر اساس منطق اصلی برنامه در این لایه انجام‌شده و نتیجه این پردازش مجدداً به لایه نمایش منتقل‌شده و برای کاربر به نمایش درمی‌آید. گاهی اوقات درخواست کاربر به گونه‌ای است که لایه منطق تجاری برای انجام آن نیاز دارد که با لایه داده یعنی لایه زیرین خود ارتباط داشته باشد در نهایت داده‌های برنامه شما نیاز به ذخیره و بازیابی از طریق یک پایگاه داده دارند این وظیفه توسط لایه دسترسی به داده انجام می‌شود. این لایه که به آن لایه بانک اطلاعاتی نیز گفته می‌شود وظیفه مدیریت اطلاعات موجود در بانک اطلاعاتی یا همان Database را بر عهده دارد و بر اساس درخواست‌هایی که از لایه بالایی خود دریافت می‌کند عملیاتی از قبیل: حذف، اضافه، اصلاح، خواندن اطلاعات و... را بر روی بانک اطلاعاتی انجام داده و نتیجه عمل را به لایه بالایی خود ارسال می‌کند. باید توجه داشت که ارتباط با بانک اطلاعاتی فقط از طریق لایه داده انجام می‌گیرد.

**به طور خلاصه، فرآیند مورد نظر ما این‌گونه کار می‌کند:**

-کاربر برای داده‌های برنامه درخواستی ارسال می‌کند.

-لایه Data Access داده‌های مورد نظر را بازیابی می‌کند و از طریق لایه Business Logic آن‌ها را به لایه نمایش می‌فرستد. بعضی مواقع لایه دسترسی به داده‌ها، این داده‌ها را مستقیماً به لایه نمایش ارسال می‌کند.

-لایه نمایش اطلاعاتی که باید نمایش داده شوند را از طریق لایه Business Logic دریافت می‌کند.

-کاربر داده‌ها را تغییر می‌دهد و عمل مناسب در مورد آن‌ها را اجرا می‌کند(مثل اضافه یا به‌روز کردن داده‌ها)

-لایه Business Logic صحت داده‌های واردشده توسط کاربر را بررسی می‌کند(داده‌ها را اعتبارسنجی می‌کند )

-اگر داده‌ها معتبر باشند آن‌ها را برای به‌روزرسانی در بانک اطلاعاتی به دست لایه دسترسی به داده می‌سپارد.

مزایا و معایب معماری سه لایه:

* برنامه‌ها به چند بخش منطقی جدا از هم تقسیم می‌شوند و اتصال میان UI (رابط کاربری)، پردازش‌ها و بانک اطلاعاتی کم می‌شود.
* تغییر در بانک اطلاعاتی یا روال‌های دسترسی به داده‌ها تأثیری در لایه نمایش یا برنامه کلاینت نخواهد گذاشت.
* برنامه کلاینت با عبارات SQL آمیخته نخواهد شد.
* نام جداول و ستون‌ها به طور موثری از برنامه کلاینت حذف می‌شوند.
* برنامه کلاینت نمی‌فهمد که داده‌ها از کجا آمده‌اند (چیزی که به آن Location Transparency) گفته می‌شود.
* تغییر یا گسترش برنامه بسیار ساده‌تر خواهد شد، بدون نیاز به تغییر یا کامپایل مجدد برنامه کلاینت.
* نکته منفی در معماری چند لایه این است که شما باید تعداد زیادی بخش‌ها و کلاس‌های از هم جدا در نرم‌افزار بسازید. برای سیستم‌های کوچک استفاده از معماری 3 لایه زمان بر و هزینه بر می‌باشد و در واقع هزینه توسعه و نگهداری سیستم بالا می‌رود.

اما به هر حال مزایای این روش بیشتر و برتر از معایب آن است.

در این قسمت شمایی از معماری فریم‌ورک Yii آورده شده است.



## تجزیه به زیرسیستم ها

در این قسمت از Work Breakdown Structure استفاده میشود.در WBS سامانه کلید آسمان 3 قسمت اصلی وجود دارد:1-Deliverables 2-System 3-Work Package

1. سامانه کلید آسمان(%100)
   1. مدیریت پروژه(%10)
      1. برنامه‌ریزی
      2. مدیریت هزینه و زمان
      3. مدیریت Scope (محدوده کاری)
      4. مدیریت Task ها
      5. مدیریت ارتباطات(ارتباط با کارفرما و مشاور کارفرما)
      6. مدیریت نیروی انسانی
      7. مدیریت ریسکها
      8. مدیریت بهره وری
      9. مدیریت کیفیت
   2. مهندسی سیستم(%15)
      1. برنامه ریزی فنی
      2. مدیریت فنی
      3. تعریف نیازمندی های کار فنی
      4. تعریف نیازمندی های سیستم
      5. معماری سیستم و طراحی سطح بالا
   3. نرم افزار(%40)
      1. طراحی نرم افزار
         1. مشخص کردن نیازهای نرم افزار
         2. تعریف فازهای کار و مشخص کردن بسته های کاری
         3. مشخص نمودن نمونه اولیه نرم افزار
         4. مشخص نمودن جزئی طراحی نرم افزار
      2. توسعه و ساخت نرم افزار
         1. کدنویسی نرم افزار
         2. خطایابی کد نرم افزار
      3. تست فاز
         1. برنامه ریزی تست
         2. آماده سازی فاز برای تست
         3. مشخص کردن نوع تست
         4. ثبت نتایج تست
   4. سخت افزار(5%)
      1. تعریف سخت افزارهای موردنیاز
      2. انتخاب سخت افزارهای موردنیاز
      3. تهیه سخت افزارهای موردنیاز
      4. تست سخت افزارهای تهیه شده
   5. مدیریت Deliverables(بسته های تحویل)(%5)
      1. پیگیری بسته های تحویل(Tracking)
      2. بسته بندی بسته های تحویل و تحویل به کارفرما
   6. تست سامانه(%15)
      1. تست زیرسیستم ها و ماژول ها
      2. تست ملحقات سیستم
      3. تست تایید کارفرما
      4. بسته بندی و معیارهای موفقیت تست
   7. پشتیبانی سامانه(%5)
      1. مدیریت پیکربندی سامانه
      2. تضمین کیفیت سامانه
      3. تحویل نرم افزارهای توسعه
      4. آموزش تیم فنی کارفرما
   8. نصب سامانه(%5)
      1. برنامه ریزی نصب
      2. راهنمای کاربر
      3. مدیریت و هماهنگی برای نصب سامانه
      4. نظارت بر نصب سامانه
      5. تست نصب و تایید نهایی کارفرما

## نگاشت سخت افزار/ نرم افزار

## مدیریت پایدار داده ها

برای ذخیره داده‌های پایدار از پایگاه داده MySQL استفاده میشود.از فایل‌ها پشتیبان گرفته میشود ولی نیاز به وجود کپی از فایل‌های پایگاه داده در استفاده از سامانه نمیباشد.پشتیبان به دلیل حفاظت از اطلاعات در مقابل خطرات فیزیکی و امنیتی گرفته میشود.در ضمن نیاز به Shard میباشد ولی این کار توسط DBMS انجام میشود و این فایلهای خرد بعد از انجام شدن Query پاک میگردد.

## امنیت و سطح دسترسی

در سامانه کلید آسمان برای Authentication و Authorization در فازهای ابتدایی از مکانیزم Access Control Filter استفاده میشود.در فاز انتهایی برای کنترل دسترسی افراد از مکانیزم RBAC یا Rule-Based Access Control استفاده خواهد شد.دو مکانیزم استفاده شده در پروژه در فریم ورک Yii پیاده‌سازی شده است.

## مکانیزم کلی کنترل نرم افزار

## شرایط حدی

با توجه به اینکه سامانه کلید آسمان تحت وب میباشد شروع این سامانه بدین صورت میباشد که تا زمانی که تمام سرویس‌های سیستم لود نشود سامانه به صورت آنلاین نخواهد بود.به محض لود شدن سامانه کاربران میتوانند از سامانه استفاده نمایند.در شروع کار کاربر با سامانه اولین کارخواستی که فراخوانی میشود کارخواست ورود میباشد.این کارخواست مشخص میکند که در حال حاضر کاربر وارد سیستم شده است یا خیر.سپس با توجه به این اطلاعات کارخواست های دیگر فراخوانی میشود.

پایان سامانه در یک سامانه آنلاین برای سرویس‌دهنده بی‌معنی میباشد.در صورتی که سامانه به صورت دستی از حالت آنلاین خارج شود،ابتدا تمام کاربرانی که در اکانت خود وارد شده‌اند توسط سامانه از اکانت خود خارج میشوند.در واقع آخرین کارخواستی که فراخوانی میشود،کارخواست خروج میباشد و سپس تمام سرویس‌ها بسته میشوند.

هنگام وقوع خطا در سامانه هنگامیکه در اصطلاح خطای سیستمی(System Error) رخ دهد سعی بر این خواهد بود که این خطا به یک خطای کشنده(Fatal Error) تبدیل نشود.این مکانیزم به صورتی تقریبا کامل در فریم‌ورک Yii پیاده‌سازی شده است.طبیعتا فریم‌ورک خطای سیستمی را که مربوط به قسمت نرم‌افزاری سیستم باشد را تحت پوشش قرار میدهد.ار خطای سیستمی که مربوط به سخت‌افزار باشد،رفع آن توسط سیستم‌عامل موجود بر روی سرور انجام میشود.اکثر خطاهای سیستمی که مربوط به سخت‌افزار میباشد به یک خطای کشنده تبدیل میشود.

هنگامی که خطا از سوی کاربر رخ دهد که به آن User Error نیز گفته میشود،با پیغام‌های مناسب از کاربر خواسته میشود که مجددا تلاش نماید.مثلا در صورت تکمیل نبودن یک فیلد اجباری از یک فرم بار دیگر فرم به کاربر نمایش داده میشود.در صورت تلاش برای دسترسی به سطحی که کاربر مجاز نیست،به کاربر پیغام مناسبی در این مورد نمایش داده میشود.

# فهرست سرویس های زیرسیستم ها

ورود:ورود کاربران به حساب کاربری خود

خروج:خروج کاربران از حساب کاربری خود

ثبت‌نام:ایجاد حساب کاربری برای مسئولین فرهنگی مساجد

با توجه به اینکه معماری سیستم MVC میباشد،Interfacing بدین صورت میباشد که اطلاعات توسط بخش Model جمع‌آوری میشود و سپس بخش Controller اطلاعات را به بخش View منتقل مینماید.

البته اگر از Widget های فریم‌ورک استفاده شود ارتباط مستقیم View و Model نیز فراهم میشود.