



Athena

مسابقه ساز

۹۵۱۸۱۱۱۹۳۰۱۶

رافی دیلانچیان

۹۵۱۸۱۱۱۹۳۰۳۳

محمد حسین کمیلی

استاد درس: دکتر زهرا مصلحی

مهندسی نرم افزار

بهار ۱۳۹۹

فهرست مطالب

۱ مقدمه..... ۶

۱.۱ مقدمه و توضیح کلی درباره پروژه..... ۶

۱.۲ اهداف پروژه و ضرورت انجام آن..... ۷

۱.۳ توجیه اقتصادی پروژه..... ۷

۱.۴ تاثیر این پروژه در آینده کاری..... ۹

۱.۵ نحوه تبادل دانش میان توسعه دهندگان و مشتری در خلال

انجام پروژه..... ۹

۱.۶ مدل فرآیند..... ۹

۲ برنامه ریزی..... ۱۰

۲.۱ برآورد کلی پروژه از لحاظ انسانی و مالی..... ۱۰

۲.۲ زمانبندی و تحلیل ریسک های موجود..... ۱۰

۳ تحلیل و طراحی..... ۱۱

۳.۱ نیازسنجی..... ۱۱

۳.۲ نیازهای اولیه برنامه..... ۱۱

۳.۲.۱ Functional..... ۱۱

۳.۲.۲ Non Functional..... ۱۲

۳.۳ روش استفاده شده در جمع آوری نیازها..... ۱۴

۳.۴ مدل سازی نیازها..... ۱۵

۳.۴.۱ Fishbone..... ۱۵

۳.۴.۲ Usecase Diagram..... ۱۶

۳.۴.۳ Calss Diagram..... ۱۷

۳.۴.۴ Collaboration Diagrams..... ۱۸

۱۹.....	CRC Cards	۳.۴.۵
۲۰.....	ERD Models	۳.۴.۶
۲۱.....	Sequence Diagrams	۳.۴.۷
۲۲.....	طراحی UI/UX (صفحات و دیالوگ‌ها)	۳.۵
۲۳.....	UI State Diagram طرح	۳.۶
۲۴.....	الگوهای طراحی (Design Pattern)	۳.۷
۲۴.....	Bridge	۳.۷.۱
۲۴.....	Observer	۳.۷.۲
۲۵.....	Builder	۳.۷.۳
۲۵.....	Command	۳.۷.۴
۲۵.....	Servant	۳.۷.۵
۲۶.....	Singleton	۳.۷.۶

۱ مقدمه

۱.۱ مقدمه و توضیح کلی درباره پروژه

Athena نرم‌افزاری است که برای ساخت و اجرای مسابقات علمی - آموزشی و تفریحی در مدارس و برنامه‌های اجرایی از جمله مسابقات و game show های تلویزیونی مورد استفاده قرار می‌گیرد.

طراحان مسابقات علمی - آموزشی می‌توانند به کمک Athena سوالات متنی و محتوای چندرسانه‌ای مورد استفاده در مسابقه را با قالب های کاربردی و جذاب به راحتی نمایش دهند.

قابل ذکر است که ریسپانسیو بودن ال برنامه این امکان را فراهم میکند که به راحتی و بدون هیچ‌گونه تغییرات خاصی بتوان مسابقات را بر روی هر صفحه نمایشی با اندازه‌های متفاوت اجرا کرد.

این نرم افزار از انواع سوالات مانند سوالات متنی و تصویری، چند گزینه‌ای، تشریحی، حدس تصویر، حدس صوت، کارت شانس و ... پشتیبانی می‌کند.

۱.۲ اهداف پروژه و ضرورت انجام آن

Athena با بهبود سطح کیفیت اجرایی مسابقات در قالبی زیبا، پر کاربرد و آسان می تواند با حفظ جنبه‌ی علمی-آموزشی جنبه‌ای تفریحی داشته و استفاده از آن در مدارس و آموزشگاههای علمی میتواند سطح علمی و علاقه‌مندی دانش‌آموزان و دانشجویان را بالا ببرد.

۱.۳ توجیه اقتصادی پروژه

جامعه‌ی هدف این نرم‌افزار آموزشگاه ها و مدارس می‌باشد، لذا فروش آن نیازمند ویزیت از این ارگان ها می‌باشد.

فروش آن به صورت license based صورت می‌پذیرد، به این شکل که مشتریان می‌توانند با دریافت کد فعال‌سازی، نرم افزار را برای مدت زمان مشخص خریداری کنند، و همچنین در صورت نیاز آن را برای بازه زمان بیشتری تمدید کنند. مشتریان می‌توانند در مدت زمان استفاده، از به‌روز رسانی های نرم افزاری بهره‌مند شوند.

فروش یا تمدید لایسنس این نرم افزار به صاحب پروژه این امکان را می‌دهد که پشتیبانی از نرم افزار را بعد از فروش نیز ادامه دهد.

از طرفی مهمترین مزیت استفاده از Athena در آموزشگاه ها، مدارس و ... صرفه جویی در زمان لازم برای طراحی و ایجاد امتحان های کلاسی، آزمونک ها و منابع کمک آموزشی چندرسانه ای و ... میباشد .

همچنین Athena به دلیل داشتن رابط کاربری زیبا و آسان قابلیت استفاده برای تمام دانش آموزان، دانشجویان و معلمان را فراهم می‌کند .

بدین شکل استفاده کنندگان از این نرم افزار از آموزش به شکل نوین همراه با چالش و سرگرمی های علمی-آموزشی بهره مند می‌شوند. استفاده از این نرم افزار در game show می‌تواند موجب افزایش سرعت، دقت، راحتی و کیفیت مسابقه شود و در نهایت افزایش مخاطبان مسابقه را به همراه دارد.

تفاوت‌های Athena با محصولات مشابه

با بررسی و مشاهده نرم افزار های مشابه موجود در بازار و همچنین نیازسنجی های صورت گرفته می‌توان به یک مقایسه‌ی نسبی میان این محصول با دیگر محصولات رسید:

Athena	دیگر نرم‌افزار ها	پاورپوینت	ویژگی
✓	✓	✗	تغییر و افزودن راحت سوالات
✓	✗	✗	گرافیک بالا
✓	✓	✓	پشتیبانی از سوالات چندگزینه‌ای
✓	✓	✓	پشتیبانی از سوالات حدس تصویر
✓	✗	✗	پشتیبانی از سوالات حدس صوت
✓	✗	✗	تغییر زبان و پوسته
✓	✓	✗	امتیازدهی
✓	✗	✗	شخصی‌سازی پوسته

۱.۴ تاثیر این پروژه در آینده کاری

با در نظر گرفتن مخاطبان Athena، انجام این پروژه علاوه بر فراهم آوردن تجارب فراوان در زمینه های پیاده سازی و تولید نرم افزار هایی از این دست باعث ایجاد ارتباطاتی سودمند برای پروژه های آینده و همچنین داشتن پروژه نمونه و تولید ارزش و اعتبار برای آینده توسعه دهندگان می باشد.

۱.۵ نحوه تبادل دانش میان توسعه دهندگان و مشتری در خلال انجام پروژه

با توجه به انتخاب Scrum به عنوان مدل فرآیند در پروژه، مشتری در صورت نیاز هر روز در جلسات ۱۵ دقیقه ای با عنوان scrum meeting از نزدیک در پروسه تکمیل و توسعه نرم افزار حضور دارد و تبادل دانش از این طریق و همچنین از طریق مصاحبه های صورت گرفته انجام می پذیرد.

۱.۶ مدل فرآیند

Agile process model : Scrum

مدل فرآیند در نظر گرفته شده برای استفاده، یکی از انواع مدل فرآیندهای چابک با عنوان scrum است .

دلیل استفاده از scrum، راحتی مدیریت، ایجاد اولویت بین وظایف موجود، ارتباط موثر ذینفعان از طریق جلسه کوتاه روزانه و همچنین فراگیری و راحتی استفاده از این روش است.

۲ برنامه‌ریزی

۲.۱ برآورد کلی پروژه از لحاظ انسانی و مالی

- با توجه به ملزومات پروژه، برآورد اولیه، ۴ نفر متخصص با وظایف زیر است :
- دو نفر front-end و back-end
- یک نفر طراحی UI/UX
- یک نفر جهت فراهم آوردن محتوای علمی-آموزشی برای نسخه اولیه نرم افزارو

به وضوح برآورد مالی پروژه وابسته به حقوق و منابع مورد نیاز توسعه دهندگان می باشد. اما انتظار می رود که اعضای تیم، با توجه به تخصص و مهارت آن ها در زمینه‌ی کاری خود، به صورت تمام وقت استخدام شده و حقوق آنها با توجه به تعرفه‌های اداره کار مشخص شود.

۲.۲ زمانبندی و تحلیل ریسک‌های موجود

با بررسی‌های انجام شده تخمین زمانی که برای ددلاین این پروژه صورت گرفته است به این صورت می باشد:

توسعه دهندگان این پروژه به صورت تمام وقت طی مدت ۳ تا ۵ ماه (۵۰۰ تا ۶۰۰ ساعت) به توسعه نرم افزار مشغولند.

از طرفی مشکلاتی که ممکن است در خلال انجام پروژه پیش بیاید غیر قابل انکار است. از جمله ناتوانی در اتمام پروژه در مدت زمان تخمینی، و یا کاهش نیاز بازار به این نرم افزار با عرضه‌ی محصولات مشابه رقبا.

۳ تحلیل و طراحی

۳.۱ نیازسنجی

در بررسی های اولیه نیاز به وجود نرم افزاری که بتواند مفاهیم علمی - آموزشی را در قالبی متفاوت و چالشی با استفاده از محتوای چندرسانه ای به مخاطب منتقل کند، همچنین با استفاده از روش هایی که در ادامه ذکر میشود، نرم افزاری که مدیریت سوالها و شرکت کننده ها برای برگزارکنندگان مسابقات با قالب game show را آسان تر، جذاب تر و سریع تر کند حس میشد.

۳.۲ نیازهای اولیه برنامه

۳.۲.۱ Functional

۱. بارگذاری سوالات با توجه به نوع سوالی که وجود دارد
۲. تغییر سوالات از درون برنامه
۳. نمایش سوالات متفاوت با توجه به نوع سوال (متنی، تصویری، توضیحی و ...)
۴. بارگذاری تصویر و صوت (و همچنین حذف آنها)
۵. تغییر اندازه و نسبت تصویر
۶. تعریف، تغییر و حذف تیم
۷. نمایش تیم ها و امتیازها در حین بازی
۸. اضافه و کم کردن امتیاز از تیم
۹. تغییر پوسته و شخصی سازی
۱۰. تغییر رنگ ها

۱۱. تغییر رنگ و تصویر پس زمینه
۱۲. تغییر زبان
۱۳. تغییر اندازه و فونت
۱۴. تهیه فایل پشتیبان از پایگاه داده بازی
۱۵. بارگذاری بازی های دیگر و سوالات دیگر
۱۶. حذف و شروع مجدد بازی
۱۷. دریافت بروزرسانی
۱۸. اضافه کردن سوالات ترکیبی (تصویر و متن و ... به صورت همزمان)
۱۹. داشتن edit mode برای تغییر در سوالات و همچنین پوسته
۲۰. اضافه کردن تایمر به هر سوال به صورت مجزا یا گروهی
۲۱. اضافه یا حذف یا تغییر سوالات هر بخش به صورت دسته‌ای
۲۲. نمایش نتایج امتیازها در صفحه مجزا به صورت نمودار

Non Functional ۳.۲.۲

۱. کاملا آفلاین باشد
۲. پشتیبانی از تمام سیستم عامل های Windows, MacOS, GNU/Linux
۳. ظاهر زیبا با آیکون ها و ویژگی های جذاب
۴. قابلیت شخصی سازی
۵. پشتیبانی از تعداد نامحدود سوال
۶. بارگذاری سریع بازی و اعمال سریع تغییرات
۷. حاوی صداهای درون برنامه ای باشد
۸. responsive بودن style های برنامه برای هر رزولوشن
۹. دارای UI/UX مناسب به طوری که کار کردن با آن نیاز به آموزش خاصی نداشته باشد
۱۰. داده های پایگاه داده همیشه به روز باشد

۳.۳ روش استفاده شده در جمع آوری نیازها

مشاهده نحوه برگزاری game show و همچنین برگزاری رویدادهایی مشابه باعث احساس نیاز به نرم افزاری برای مدیریت و تسهیل فرآیند فوق بود.

با مطالعه سیستم های موجود برای مدیریت مسابقات game show و همچنین یک منظره بودن استفاده از این دست نرم افزارها با وجود داشتن پتانسیل استفاده در موارد مشابه , نیاز به وجود نرم افزاری با طراحی جذاب تر و کاراتر نه تنها برای استفاده در game show ها بلکه به عنوان یک سیستم آموزش مبتنی بر محتوای چندرسانه ای را پررنگ تر کرد.

همچنین انجام مصاحبه با چندی از برگزارکنندگان و معلمان که مخاطبین اصلی چنین نرم افزاری محسوب می شوند , این نیاز را بیش از پیش آشکار کرد.

همانطور که ذکر شد بخش قابل توجهی از نیازسنجی از طریق مصاحبه، و بخشی از آن با بررسی کارایی و نقاط ضعف و قوت نرم افزارهای مشابه موجود صورت گرفته است.

۳.۴ مدل سازی نیازها

Fishbone ۳.۴.۱

Usecase ୩.୪.୨

Diagram

Calss Diagram ۳.۴.۳

Collaboration ۳.۴.۴

Diagrams

CRC Cards ୩.୪.୫

ERD Models ۳.۴.۶

Sequesnce ୩.୪.୧

Diagrams

۳.۵ طراحی UI/UX (صفحات و دیالوگ‌ها)

٣.٦ طرح UI State Diagram

۳.۷ الگوهای طراحی (Design Pattern)

Bridge ۳.۷.۱

برای ذخیره‌سازی سوالات و جواب‌های آن‌ها از Bridge استفاده شده است. در این Design Pattern سعی می‌شود که پیاده‌سازی‌ها از Abstraction ها مجزا باشند تا بتوانند قابل جایگزینی و تعویض باشند. به همین منظور برای ذخیره‌سازی از این Pattern استفاده شده است تا بتوان عمل ذخیره‌سازی را به صورت‌های مختلف انجام داد. نتیجه این Design Pattern این است که می‌توان تمام داده‌ها اعم از تنظیمات مسابقه، امتیاز تیم‌ها، سوالات و جواب‌ها را در دیتابیس‌های مختلف و حتی در فایل ذخیره کرد.

Observer ۳.۷.۲

این Design Pattern ما را قادر می‌سازد تا قسمت‌هایی از برنامه دائما در حال گوش‌دادن به تغییرات وضعیت یا State بخش دیگری از برنامه باشند و متناسب با این تغییرات واکنش متناسب نشان دهند. برای طراحی بسیاری از بخش‌های الی و ظاهر برنامه از این Design Pattern استفاده شده است. به طول مثال بخشی از برنامه یک برد نشان می‌دهد که امتیاز تمام تیم‌های مسابقه را به صورت نمودار و همین طور به صورت عدد نشان می‌دهد. به محض این که امتیاز یک تیم تغییر کند، برد خود به خود به روز شده و آخرین تغییرات را نشان می‌دهد. برای پیاده‌سازی چنین ویژگی از Observer استفاده شده است و برد در حال Observe کردن امتیازات تیم‌هاست و اصطلاحا به این تغییرات Subscribe شده است و به محض تغییر امتیازات نمودارها دوباره به روز شده و رسم می‌شوند.

Builder ۳.۷.۳

این Design Pattern ما را قادر می‌سازد تا جدا و خارج از یک نمایش خاص از Object بتوانیم در موارد نیاز آن را بسازیم و از آن استفاده کنیم. برای ساخت سوال در بازی از این Design Pattern استفاده شده است تا کاربر بتواند به راحتی خارج از صفحه نمایش دهنده سوال، یک سوال بسازد، تغییرات لازم را بدهد و آن را ذخیره کند و بعداً در صفحه نمایش دهنده سوال آن را ببیند و از آن استفاده کند.

Command ۳.۷.۴

این Design Pattern ما را قادر می‌سازد تا بتوانیم اجرای یک دستور را در قالب یک Object توصیف کرده و به مدت مورد نظر ارسال کنیم. در این برنامه برای ارسال اجرای بسیاری از دستورات از این Design Pattern استفاده شده است. به طور مثال برای ارسال دستور گرفتن نسخه پشتیبان از کل مسابقه یک Object ساخته می‌شود. این Object حاوی تمام جزئیات لازم برای اجرای این دستور است، مانند محل ذخیره فایل پشتیبان و نام آن. سپس متدی که مسئول تهیه نسخه پشتیبان است با استفاده از این Object و این Design Pattern این دستور را اجرا می‌کند. برای بقیه عملیات‌های ذخیره‌سازی، پاک کردن و ریست کردن کل مسابقه و دیگر عملیات نیز از این Design Pattern استفاده شده است.

Servant ۳.۷.۵

به کلاس‌هایی که دارای متدهای کمکی یا Helper باشند و دارای شی خاصی نباشند Servant گفته می‌شود. برای بسیاری از عملیات از این Design Pattern استفاده شده است. مثلاً برای نوشتن فایل‌های پشتیبان بر روی دیسک، خواندن فایل‌ها و فشردن فایل‌های مسابقه از این کلاس‌ها و متدها استفاده شده است.

Singleton ۳.۷.۶

این Design Pattern ما را قادر می‌سازد تا از یک Object تنها یک instance داشته باشیم و بتوانیم در سرتاسر برنامه از همان یکی استفاده کنیم. در این برنامه از این Design Pattern برای دسترسی به آبجکت Game استفاده شده است. این آبجکت حاوی تمام اطلاعات و دیتای مسابقه، سوالات، جواب‌ها، تنظیمات و امتیازات است که باید تنها یک instance داشته و همه جا به همین یکی ارجاع داده شود.

پایان

کلیه حقوق این نرم افزار محفوظ است

Athena Game Show Maker ©