

توضیحات مقدماتی

در این پیوست، مراحل مختلف انجام پروژه مهندسی نرم افزار برای توسعه‌ی افزونه به صورت جامع و دقیق تشریح خواهد شد. هدف این است که دانشجویان با تمام بخش‌های کلیدی پروژه آشنا شوند و بتوانند آن‌ها را به درستی اجرا کنند. این بخش‌ها شامل تحلیل نیازمندی‌ها، طراحی، پیاده‌سازی، تست و ارزیابی نهایی است.

در هر مرحله، نیازمندی‌ها، ابزارها و روش‌های استاندارد توضیح داده شده‌اند تا دانشجویان بتوانند با اطمینان و دانش کافی پروژه خود را پیش ببرند. همچنین، حداقل‌ها و استانداردهای ضروری برای موفقیت پروژه معرفی خواهند شد تا دانشجویان با اطمینان و کارآمدی پروژه‌های خود را انجام دهند. توجه داشته باشید راه‌حل‌های ذکر شده صرفاً پیشنهاد‌های موجود برای پیشبرد پروژه می‌باشد و انتخاب راه‌حل مناسب بر عهده شما می‌باشد و با توجه به نیاز و ماهیت پروژه راه‌حل‌ها مشخص خواهند شد.

ساختار این پیوست شامل مراحل زیر است:

- نیازمندی‌ها و تحلیل^۱
- طراحی^۲
- پیاده‌سازی^۳
- آزمون^۴
- استقرار^۵
- برنامه‌ریزی^۶
- چالش‌ها

requirements and analysis^۱
 design^۲
 implementation^۳
 test^۴
 deployment^۵
 planning^۶

نیازمندی ها و تحلیل

نیازمندی ها

در این بخش، باید قالب های مختلف مستندسازی را بررسی کرده و با توجه به پروژه ی خود قالب مناسب را انتخاب نمایید. به عنوان مثال:

(۱) شناسایی نیازمندی ها

- تمامی نیازمندی های کارکردی و غیر کارکردی افزونه را شناسایی کنید.
- این نیازمندی ها باید با توجه به اهداف سامانه و انتظارات کاربران استخراج و مستند شوند.
- نیازمندی ها را در قالب مناسب مستند کنید به عنوان مثال داستان کاربر^۷

تحلیل

هدف از تحلیل این است که نیازمندی ها و ساختار کلی سیستم به خوبی درک شوند و بر اساس آن، طراحی و پیاده سازی انجام گیرد.

انتظارات از بخش تحلیل شامل :

- شناسایی و توضیح دقیق اجزای مختلف سامانه
- نمایش نحوه تعامل اجزا و فرآیندهای سامانه
- ارایه مستنداتی که به درک بهتر ساختار و نیازمندی های سامانه کمک کند

توضیح افزونه :

- هدف از توسعه افزونه را شرح دهید.
- مسائلی که قرار است توسط افزونه حل شود را بیان کنید.
- ویژگی های کلیدی افزونه برای کاربران را مشخص کنید.

مثال های مستندسازی :

- مدل مفهومی : برای نمایش ارتباطات مفهومی بین اجزای سامانه
- تحلیل جریان داده : برای نمایش جریان اطلاعات در سامانه و نحوه پردازش آن ها

- نمودار های ارتباطی : برای نمایش روابط بین اجزا و مولفه های مختلف سامانه
- نمودار های کلاس شیءگرا : اگر توسعه بر پایه شیءگرایی باشد
- جداول نیازمندی ها : شامل لیست کامل و مستند نیازمندی ها با توضیحات دقیق

طراحی

در این بخش، انتظار می‌رود که دانشجویان معماری کلی افزونه را به صورت جامع و دقیق تشریح کنند. این بخش باید شامل توضیحات واضح از ساختار کلی افزونه و اجزای اصلی آن باشد، به گونه‌ای که ارتباط و تعامل میان بخش‌های مختلف افزونه به خوبی نمایش داده شود.

موارد مورد نیاز در فاز طراحی :

- نمودارهای معماری : برای نمایش ساختار کلی سیستم و ارتباطات بین اجزای اصلی
- نمودارهای کلاس : برای نمایش کلاس‌ها و روابط بین آن‌ها در سیستم‌های شیء‌گرا
- نمودارهای ترتیبی : برای نمایش تعاملات پویا بین اجزای مختلف در سناریوهای خاص
- نمودارهای فعالیت : برای نمایش جریان کار و فرآیندهای داخلی افزونه
- نمودارهای مؤلفه‌ها : برای نمایش ساختار اجزای مختلف سیستم و نحوه تعامل آن‌ها
- نمودارهای توالی : برای نمایش ترتیب عملیات و تعاملات در سیستم
- نمودارهای جریان داده (DFD) : برای برنامه‌های غیر شیء‌گرا، نمایش جریان اطلاعات و نحوه پردازش داده
- نمودارهای موجودیت-ارتباط (ERD) : برای مدل‌سازی داده‌ها و ارتباط بین موجودیت‌ها در برنامه‌های غیر شیء‌گرا
- فلوچارت‌ها : برای نمایش الگوریتم‌ها و منطق تصمیم‌گیری در بخش‌های مختلف سیستم.

پیاده سازی

در این مرحله، لازم است تمامی بخش‌های مورد نیاز افزونه به صورت کامل و دقیق پیاده‌سازی شوند. همچنین استفاده از ابزارهای مدیریت نسخه مانند گیت^۸ ضروری است.

انتظارات در مرحله پیاده سازی

(۱) پیاده سازی کد

- تمامی نیازمندی‌های کارکردی و غیرکارکردی مستند شده را به کد تبدیل کنید
- کد باید مطابق با استانداردهای برنامه‌نویسی باشد و قابل خواندن و نگهداری باشد

(۲) مدیریت نسخه^۹

- از ابزارهای مدیریت نسخه مانند گیت برای پیگیری تغییرات کد استفاده کنید
- مخزن کد^{۱۰} را ایجاد کرده و تغییرات را به صورت منظم و با پیام‌های مستند به آن اضافه کنید
- شاخه‌های مختلف^{۱۱} برای ویژگی‌های مختلف ایجاد کرده و در نهایت آن‌ها را با شاخه اصلی ادغام کنید

(۳) مستندسازی کد

- کد خود را به صورت کامل و واضح مستند کنید
- توضیحات لازم را در داخل کد اضافه کنید تا دیگران بتوانند به راحتی آن را درک کنند
- از ابزارهای مستندسازی خودکار مانند Javadoc یا Sphinx استفاده کنید تا مستندات دقیق و کاملی از کد تولید شود

Git^۸

Version Control^۹

Repository^{۱۰}

Branches^{۱۱}

آزمون

در این بخش، انتظار می‌رود که برای افزونه خود مجموعه‌ای از آزمون‌ها طراحی و اجرا کنید. لازم است نوع آزمون‌های مورد نیاز افزونه خود را با دقت بررسی کرده و انتخاب کنید. این آزمون‌ها می‌توانند شامل آزمون‌های واحد برای ارزیابی عملکرد بخش‌های مجزا، آزمون یکپارچگی برای اطمینان از هماهنگی صحیح میان اجزای مختلف، و آزمون‌های کارکردی برای بررسی رفتار افزونه در شرایط واقعی باشند. امروزه تست و پیاده‌سازی تقریباً با هم ترکیب شده‌اند، اما برای درک بهتر دانشجویان این بخش‌ها را از هم جدا کردیم.

مراحل و انتظارات در بخش آزمون

(۱) انتخاب نوع آزمون‌ها

- بررسی و انتخاب نوع آزمون‌های مناسب برای افزونه
- این آزمون‌ها می‌توانند شامل موارد زیر باشند
 - آزمون‌های واحد^{۱۲}: برای ارزیابی عملکرد بخش‌های مجزای افزونه.
 - آزمون‌های یکپارچگی^{۱۳}: برای اطمینان از هماهنگی صحیح میان اجزای مختلف
 - آزمون‌های کارکردی^{۱۴}: برای بررسی رفتار افزونه در شرایط واقعی

(۲) طراحی آزمون‌ها

- طراحی دقیق هر یک از آزمون‌ها بر اساس نیازمندی‌ها و مشخصات افزونه
- تعریف موارد آزمون^{۱۵} برای هر آزمون
- مشخص کردن ورودی‌ها، خروجی‌ها و شرایط اولیه برای هر آزمون

(۳) پیاده‌سازی آزمون‌ها

- پیاده‌سازی آزمون‌ها با استفاده از ابزارها و چارچوب‌های مناسب
- اجرای آزمون‌ها به صورت خودکار یا دستی و ثبت نتایج

(۴) ارزیابی و رفع مشکلات

- تحلیل نتایج آزمون‌ها و شناسایی مشکلات و نواقص
- رفع مشکلات شناسایی شده و بهبود افزونه بر اساس نتایج آزمون‌ها
- تکرار فرآیند آزمون تا اطمینان از عملکرد صحیح و پایداری افزونه در شرایط مختلف

Unit Tests^{۱۲}
 Integration Tests^{۱۳}
 Functional Tests^{۱۴}
 Test Cases^{۱۵}

استقرار

دانشجویان باید افزونه‌ی خود را در محیط vscode مستقر کنند به صورتی که برای افراد دیگر قابل دسترسی باشد. دقت کنید که این بخش بسیار مهم است حتی اگر فقط یک ویژگی پیاده‌سازی کرده باشید، باز هم باید آن را مستقر کنید. سپس در این بخش، لازم است مستندات تهیه شود که فرآیند استقرار افزونه در محیط توسعه یکپارچه (IDE) را به طور دقیق و جامع توضیح دهند. این مستندات به کاربران و توسعه‌دهندگان کمک می‌کنند تا افزونه را به درستی و بدون مشکل در IDE خود مستقر کنند. انجام همه‌ی این مستندات اجباری نیست و هرکدام را که برای پروژه‌تان نیاز است، انجام دهید. برخی از این مستندات می‌توانند با هم ترکیب شوند.

مستندات مورد نیاز برای استقرار افزونه

(۱) پیش‌نیازها و نصب

- پیش‌نیازها: لیست تمامی پیش‌نیازهای نرم‌افزاری و سخت‌افزاری
- نصب و پیکربندی: راهنمای گام به گام نصب افزونه در IDE به همراه تصاویر

(۲) راه‌اندازی اولیه و رفع مشکلات

- راه‌اندازی اولیه : نحوه راه‌اندازی افزونه پس از نصب
- رفع مشکلات متداول : فهرست مشکلات متداول و راهکارهای رفع آن‌ها

(۳) مستندات پشتیبانی و نگهداری

- نحوه به‌روزرسانی افزونه و اعمال تغییرات جدید
- توضیحات درباره نگهداری و پشتیبانی از افزونه در بلندمدت

مستند کاربر

برای استفاده کاربران از افزونه، تهیه و تدوین یک مستند کاربر جامع و واضح ضروری است. این مستند باید شامل توضیحات دقیق و کاربردی درباره نحوه استفاده از قابلیت‌ها و بخش‌های مختلف افزونه باشد تا کاربران با کمترین دشواری بتوانند از افزونه استفاده کنند

مستند کاربر می‌تواند شامل موارد زیر باشد

(۱) نحوه استفاده از قابلیت‌ها

- توضیحات گام به گام درباره نحوه استفاده از هر قابلیت
- ارائه مثال‌های عملی و کاربردی

(۲) راهنمای تصویری

• استفاده از تصاویر و نمودارها برای نمایش بخش‌های مختلف افزونه و نحوه استفاده از آن‌ها

(۳) نکات مهم و بهترین شیوه‌ها

• نکات کلیدی برای استفاده بهینه از افزونه

• راهکارهای جلوگیری از بروز مشکلات متداول

(۴) رفع مشکلات متداول

• فهرست مشکلات متداول و راهکارهای پیشنهادی برای رفع آن‌ها

این مستندات به کاربران کمک می‌کنند تا فرآیند استقرار و استفاده از افزونه را به صورت کامل و دقیق انجام دهند و از عملکرد صحیح و پایداری افزونه اطمینان حاصل کنند

برنامه ریزی

مدیریت پروژه به صورت ساختارمند ضروری است تا تمامی مراحل توسعه به شکل منظم و هماهنگ پیش برود. یکی از مهم‌ترین بخش‌های پروژه است و لازم است با دقت و توجه انجام شود. موارد گفته شده برای برنامه‌ریزی برای شفاف شدن موضوع بوده و اجباری به انجام همه‌ی موارد نیست، اما دانشجویان باید دقت کنند موارد گفته شده را متناسب با پروژه‌ی خود انجام دهند. برای این منظور، استفاده از ابزارهای مدیریت پروژه مانند Trello یا ابزارهای مشابه پیشنهاد می‌شود. در این ابزارها باید وظایف هر عضو تیم مشخص شده و پیشرفت کار در هر مرحله پیگیری شود.

روش‌های مختلف و ساده برای برنامه‌ریزی پروژه‌ی تیمی کوچک

(۱) تقسیم وظایف

- تعیین نقش‌ها: هر عضو تیم باید نقش و وظایف مشخصی داشته باشد
- تعیین وظایف: وظایف هر عضو به صورت دقیق و واضح مشخص شود
- استفاده از ابزارهای مدیریت وظایف: ابزارهایی مانند Asana، Trello یا Jira برای تقسیم وظایف و پیگیری پیشرفت کار استفاده شود

(۲) برنامه‌ریزی زمانی

- تعیین مهلت‌ها: برای هر وظیفه مهلت‌های زمانی مشخصی تعیین کنید
- استفاده از تقویم پروژه: از تقویم‌های پروژه مانند Calendar Google یا ابزارهای مشابه برای برنامه‌ریزی زمانی و پیگیری مهلت‌ها استفاده کنید

(۳) جلسات منظم

- برگزاری جلسات تیمی منظم: جلسات هفتگی یا روزانه برای بررسی پیشرفت کار و حل مشکلات برگزار کنید
- استفاده از ابزارهای ارتباطی: ابزارهایی مانند Slack یا Teams Microsoft برای ارتباط و هماهنگی بین اعضای تیم استفاده شود

(۴) پیگیری و بازبینی

- مانیتورینگ پیشرفت: پیشرفت کار به صورت منظم پیگیری و مانیتور شود
- بازبینی وظایف: وظایف انجام شده بازبینی و ارزیابی شوند تا از کیفیت و دقت آن‌ها اطمینان حاصل شود

(۵) مستندسازی

- ثبت پیشرفت کار: تمامی پیشرفت‌ها و تغییرات در پروژه مستند شوند
- مستندسازی جلسات: نتایج و نکات مهم جلسات تیمی مستند و به اشتراک گذاشته شوند

تحویل مستندات برنامه ریزی

دانشجویان باید مستندات برنامه ریزی خود را به صورت دقیق و شفاف تهیه و تحویل دهند. این مستندات شامل موارد زیر است

(۱) برنامه زمان بندی پروژه

- فهرست تمامی وظایف و فعالیت های پروژه با مهلت های تعیین شده
- استفاده از ابزارهای برنامه ریزی مانند گانت چارت^{۱۶} برای نمایش زمان بندی وظایف

(۲) تقسیم وظایف و نقش ها

- مستنداتی که نقش و وظایف هر عضو تیم را به وضوح مشخص می کند
- استفاده از جداول و نمودارها برای نمایش توزیع وظایف

(۳) گزارشات جلسات

- مستندات مربوط به جلسات تیمی، شامل تاریخ و زمان برگزاری، موضوعات مطرح شده، تصمیمات گرفته شده و وظایف اختصاص یافته

(۴) پیشرفت کار و بازبینی

- گزارشات دوره ای از پیشرفت کار، شامل وضعیت فعلی وظایف، مشکلات و چالش های موجود و راهکارهای پیشنهادی
- مستندات بازبینی وظایف انجام شده و ارزیابی کیفیت آن ها

(۵) نکات و دستاوردهای مهم

- مستنداتی که شامل نکات کلیدی و دستاوردهای مهم پروژه تا به این لحظه هستند

چالش ها

در پایان پروژه، از هر دانشجو انتظار می‌رود که یک مستند جامع درباره چالش‌های پیش آمده تهیه کند. این بخش یکی از مهم‌ترین بخش‌های ارزیابی پروژه است و باید با دقت و شفافیت کامل تهیه شود. در این مستند، دانشجو باید به طور کامل و دقیق تمامی مشکلات و چالش‌هایی را که در طول اجرای پروژه با آن‌ها روبرو شده است، شرح دهد.

موارد مورد انتظار در مستند چالش ها

(۱) چالش‌های فنی

- مشکلاتی که در کدنویسی، طراحی سیستم، استفاده از ابزارها و فناوری‌های مورد استفاده به وجود آمده‌اند
- توضیحاتی درباره اینکه چگونه این مشکلات شناسایی و رفع شده‌اند
- مثلاً، مسائل مربوط به عملکرد، سازگاری، باگ‌ها و تأثیری که این چالش‌ها بر روند پروژه داشته‌اند

(۲) چالش‌های مدیریتی

- مسائلی که در مدیریت زمان، منابع، تقسیم وظایف و هماهنگی بین اعضای تیم رخ داده‌اند
- راهکارهایی که برای بهبود مدیریت پروژه به کار گرفته شده‌اند
- مثلاً، مشکلات در تخصیص زمان، پیگیری پیشرفت، مدیریت استرس و تأثیری که این چالش‌ها بر روند پروژه داشته‌اند

(۳) چالش‌های ارتباطی

- مشکلاتی که در ارتباطات تیمی، هماهنگی و همکاری بین اعضای تیم به وجود آمده‌اند
- روش‌هایی که برای بهبود ارتباطات و همکاری‌ها استفاده شده‌اند
- مثلاً، مشکلات در انتقال اطلاعات، تفاهم نادرست، تأخیر در پاسخ‌دهی و تأثیری که این چالش‌ها بر روند پروژه داشته‌اند

(۴) محدودیت‌ها و دشواری‌ها

- هرگونه محدودیت یا دشواری دیگری که در مسیر پیشرفت پروژه تأثیرگذار بوده‌اند
- روش‌هایی که برای غلبه بر این محدودیت‌ها پیشنهاد و به کار گرفته شده‌اند
- به عنوان مثال محدودیت‌های زمانی، مالی، تکنولوژیکی و تأثیری که این چالش‌ها بر روند پروژه داشته‌اند

توجه ۱: از زیاده‌گویی و پر کردن این بخش به طور مصنوعی بپرهیزید. موارد باید مختصر و شفاف باشند و فقط مشکلاتی که از دید خودتان موثر بوده‌اند را بیان کنید. از نقد و بدگویی درباره هم‌تیمی‌ها

یا طرح مسائل شخصی خود اجتناب کنید؛ هدف این مستند این نیست. بلکه باید چالش‌هایی که در پیشرفت پروژه تأثیر داشته‌اند را به صورت واضح بیان نمایید.

توجه ۲: لازم نیست چالش‌هایی که با آن‌ها مواجه شده‌اید پیچیده یا بزرگ باشند. این بخش بر اساس تعداد یا سطح چالش‌ها ارزیابی نمی‌شود. مهم‌تر از همه، نگاه شما به این چالش‌ها، راه حل‌هایی که ارائه داده‌اید، و تأثیر این چالش‌ها بر روند پروژه از دیدگاه شما است. بنابراین، حتی اگر در موارد ساده و ابتدایی هم چالشی داشته‌اید، حتماً به آن اشاره کنید.

خلاصه

خلاصه کارهایی که باید انجام شود. مورد آخر به صورت فردی و بقیه موارد به صورت گروهی می باشد.

(۱) نیازمندی ها

- توضیح هدف پروژه و مشکلاتی که حل می کند
- شناسایی نیازمندی ها: تمامی نیازمندی های کارکردی و غیرکارکردی را شناسایی و مستند کنید
- نکته: استفاده از روش های مستند سازی مناسب مثل داستان کاربر، نمودارها یا جداول، هدف است، نه استفاده از یک روش خاص

(۲) تحلیل

- تحلیل دقیق نیازمندی ها: بررسی و تحلیل نیازمندی های شناسایی شده
- مدل های مفهومی: ارائه مدل های مفهومی، تحلیل جریان داده ها و نمودارهای ارتباطی
- نکته: از روش های مناسب برای تحلیل و مستند سازی استفاده کنید، نه فقط از یک روش خاص

(۳) طراحی

- معماری کلی افزونه: تشریح معماری کلی افزونه
- نمودارهای طراحی: شامل نمودارهای معماری، کلاس، ترتیبی، فعالیت، مؤلفه ها و توالی
- نکته: روش های مستند سازی مناسب را برای طراحی انتخاب کنید

(۴) پیاده سازی

- کدنویسی: پیاده سازی کد بر اساس نیازمندی ها و طراحی های انجام شده
- مدیریت نسخه: استفاده از ابزارهای مدیریت نسخه مانند Git
- مستندسازی کد: مستندسازی دقیق و واضح کد

(۵) آزمون

- طراحی و اجرای آزمون ها: مجموعه ای از آزمون ها برای ارزیابی عملکرد افزونه طراحی و اجرا کنید
- شامل آزمون های واحد، یکپارچگی و کارکردی
- ثبت نتایج آزمون ها و رفع مشکلات شناسایی شده

(۶) استقرار

- نصب و راه اندازی افزونه: افزونه را به صورت قابل دسترسی برای دیگران نصب کنید
- مستندات استقرار: شامل پیش نیازها، نصب، پیکربندی و راه اندازی اولیه
- نکته: توضیحات دقیق درباره فرآیند استقرار افزونه در IDE

(۷) مستند کاربر

- مستند کاربر: تهیه و تدوین یک مستند جامع و واضح برای استفاده کاربران از افزونه
- شامل توضیحات دقیق درباره نحوه استفاده از قابلیت‌ها، راهنمای تصویری و رفع مشکلات متداول

(۸) برنامه ریزی

- مدیریت پروژه: به صورت ساختارمند با استفاده از ابزارهای مدیریت پروژه مانند Trello
- تقسیم وظایف و نقش‌ها: تعیین نقش‌ها و وظایف هر عضو تیم
- مستندسازی پیشرفت کار: شامل جلسات، بازبینی‌ها و نتایج

(۹) چالش‌ها

- مستند چالش‌ها: تهیه مستندی درباره چالش‌های پیش آمده
- شرح مشکلات فنی، مدیریتی، ارتباطی و محدودیت‌ها
- تأکید بر تأثیر چالش‌ها بر روند پروژه از دید دانشجو

سخن پایانی

دانشجویان عزیز،

این پروژه یک فرصت فوق‌العاده برای یادگیری و رشد در زمینه مهندسی نرم‌افزار است. در صورت برخورد با هرگونه مشکل یا ابهام، حتماً با مدرسان یا کمک‌مدرس‌های درس در میان بگذارید. آن‌ها برای حمایت و راهنمایی شما در دسترس هستند. از اشتباه کردن و به چالش کشیده شدن نترسید. این بخشی از فرآیند یادگیری است و می‌تواند به شما در درک عمیق‌تر مفاهیم کمک کند. سعی کنید با مطالعه و کسب دانش بیشتر، هر کدام از بخش‌ها را با دقت و توجه انجام دهید. به جای انجام سرسری وظایف، وقت بگذارید و به سواد مهندسی نرم‌افزاری خود بیافزایید. یادگیری و رشد زمانی اتفاق می‌افتد که با چالش‌ها مواجه شوید و از آن‌ها درس بگیرید. از تمامی منابع و فرصت‌های موجود بهره ببرید و با انگیزه و علاقه به کار خود ادامه دهید.

موفق باشید و منتظر موفقیت‌های بزرگی از سوی شما هستیم!

با آرزوی بهترین‌ها، تیم مدرسان