**به نام ایزد منان**

**گزارش پروژه نرم افزار مدیریت شهربازی**

درس : روش های رسمی

دانشجویان : محمد سلیمانی اصل و مصطفی تقی پور

استاد : دکتر خانزادی

دانشگاه آزاد اسلامی واحد تهران – شمال

بهار 1404

**مقدمه**

در این پروژه، هدف طراحی یک نرم‌افزار مدیریتی برای یک شهربازی است که شامل سیستم فروش بلیت، مدیریت کاربران، بازی‌ها و بررسی ایمنی و درستی رفتار سیستم است. برای این منظور از مفاهیم روش‌های رسمی مانند مشخصه‌گذاری رسمی، منطق گزاره‌ای و ماشین حالت متناهی (FSM) استفاده شده است.

**5 اصل مهم در توسعه نرم افزار ( استفاده شده در این پروژه ):**

1. Problem (مسئله):

در این مرحله، شما مشکل یا نیاز اصلی پروژه رو شناسایی می‌کنید. این شامل درک کلی از اینکه چرا این پروژه وجود داره و هدف اصلی اون چیه میشه.

در پروژه شهربازی:

* شما باید مشکل مدیریت شهربازی رو شناسایی کنید.
* آیا سیستم فعلی مشکل دارد؟ (مثلاً مدیریت بلیت‌ها، مدیریت بازی‌ها، یا دسترسی به اطلاعات برای مدیران و کاربران)
* هدف این سیستم این است که فرآیندهای خرید بلیت، مدیریت بازی‌ها، و تعاملات کاربران با محیط شهربازی رو خودکار کنه.

سوالاتی که باید بپرسید:

* هدف سیستم جیست؟
* چه مشکلاتی برای کاربران و مدیران شهربازی وجود دارد؟
* کدام فرایندها نیاز به خودکارسازی دارند؟
* سیستم چه ویژگی‌هایی باید داشته باشد؟

1. Requirement (نیازمندی‌ها):

در این مرحله، شما به جمع‌آوری و تعریف نیازمندی‌ها می‌پردازید. این نیازمندی‌ها می‌توانند شامل نیازمندی‌های عملیاتی، غیرعملیاتی، فنی و تجاری باشند.

در پروژه شهربازی:

* نیازمندی‌های عملیاتی:
* خرید بلیت باید برای کاربران امکان‌پذیر باشد.
* مدیر باید بتواند بازی‌ها را فعال یا غیرفعال کند.
* کاربران باید پس از ورود بتوانند لیست بازی‌ها را مشاهده کنند.
* نیازمندی‌های غیرعملیاتی:
* سیستم باید سریع باشد (عملکرد بالا).
* باید امنیت خوبی برای ذخیره‌سازی داده‌های کاربران و بلیت‌ها وجود داشته باشد.
* نیازمندی‌های فنی:
* باید از یک پایگاه داده برای ذخیره‌سازی اطلاعات بلیت‌ها و بازی‌ها استفاده کنید.
* رابط کاربری باید ساده و کاربرپسند باشد.
* مثال: "سیستم باید در عرض 2 ثانیه بتواند بلیت‌ها را در صفحه نمایش دهد."

1. Specification (مشخصات):

در این مرحله، باید مشخص کنید که نرم‌افزار باید چگونه عمل کند. این شامل تعریف دقیق فرآیندها و قوانین است که سیستم باید از آن‌ها پیروی کند.

در پروژه شهربازی:

* تعریف قوانین و فرایندها:
* وقتی یک کاربر وارد می‌شود، باید اطلاعاتش ذخیره شده و به او بلیت بفروشد.
* بازی‌ها باید دارای وضعیت "فعال" یا "غیرفعال" باشند و تنها بازی‌های فعال قابل خرید بلیت هستند.
* مدیر باید بتواند وضعیت بازی‌ها را تغییر دهد.
* استفاده از منطق رسمی مانند predicates برای شبیه‌سازی فرآیندهای سیستم (مثلاً شرایط خرید بلیت یا تغییر وضعیت بازی‌ها).
* قوانین سیستم: تعریف شرایط و قواعدی که سیستم باید پیروی کند. برای مثال، "اگر بازی فعال باشد، کاربر باید بتواند بلیت خریداری کند."
* شبیه‌سازی فرآیندها: رسم نمودارهای مختلف مثل نمودارهای حالت (FSM) برای نمایش وضعیت‌های مختلف سیستم و انتقالات میان آن‌ها.

1. Realisation (تحقق):

این مرحله شامل پیاده‌سازی اولیه و ایجاد مدل‌های مختلف است. شما با استفاده از ابزارهای طراحی و مدل‌سازی (مثل FSM که قبلاً صحبت کردیم) و همچنین الگوهای طراحی (مثل MVC برای پروژه‌های وب) شروع به تحقق پروژه می‌کنید.

در پروژه شهربازی:

* مدل‌سازی اولیه: استفاده از نمودارهای FSM برای نمایش وضعیت‌های مختلف سیستم (مثلاً وضعیت "ورود" کاربر، "پنل مدیریت" برای مدیر و غیره).
* مدل‌سازی منطقی: استفاده از منطق گزاره‌ای برای تعریف قواعد و شرایط سیستم.
* شروع به پیاده‌سازی پایگاه داده، به‌عنوان مثال: جداول مربوط به کاربران، بازی‌ها، بلیت‌ها.

1. Implementation (پیاده‌سازی):

این مرحله شامل پیاده‌سازی واقعی سیستم در زبان برنامه‌نویسی و ابزارهای مختلف است. در این مرحله، نرم‌افزار آماده اجرا و استفاده خواهد بود.

در پروژه شهربازی:

* برنامه‌نویسی: پیاده‌سازی منطق خرید بلیت، نمایش بازی‌ها، ورود کاربران و مدیریت توسط مدیر.
* ساخت پایگاه داده: ایجاد جداول و روابط بین کاربران، بازی‌ها و بلیت‌ها.
* ساخت رابط کاربری: استفاده از ابزارهای طراحی مانند React یا Tailwind CSS برای طراحی صفحات وب شهربازی.
* تست سیستم: آزمایش عملکرد صحیح خرید بلیت، ورود به سیستم، و مدیریت بازی‌ها.
* کدنویسی: پیاده‌سازی منطق و الگوریتم‌ها با زبان برنامه‌نویسی انتخابی.
* مثلاً در پروژه شهربازی می‌توانید از Python برای پیاده‌سازی منطق خرید بلیت و ثبت بازی‌ها و از React.js برای ساخت رابط کاربری استفاده کنید.
* تست نرم‌افزار: در این مرحله باید نرم‌افزار تست شود تا اطمینان حاصل شود که به درستی کار می‌کند و مشکلات عملکردی یا امنیتی ندارد.
* تست واحد (Unit Testing)
* تست یکپارچگی (Integration Testing)
* تست سیستم (System Testing)
* راه‌اندازی نرم‌افزار: پس از تست، نرم‌افزار آماده برای اجرا در محیط تولید می‌شود.

**مشخصات غیررسمی سیستم:**

- بازدیدکنندگان می‌توانند ثبت‌نام و ورود کنند.

- پس از ورود، می‌توانند بلیت خریداری کنند.

- مدیر سیستم می‌تواند بازی‌ها را فعال یا غیرفعال کند.

- کاربران فقط در صورت ورود، می‌توانند بلیت خریداری کنند.

- بازی‌ها فقط در صورت فعال بودن، قابل رزرو هستند.

**مشخصات رسمی (Formal Specification):**

1. ∀u ∈ Users: ¬authenticated(u) → ¬canBuyTicket(u)
2. ∀g ∈ Games: ¬active(g) → ∀u ∈ Users: ¬canBuyTicketFor(u, g)
3. ∀t ∈ Tickets: valid(t) → ∃u ∈ Users: t ∈ u.tickets

**ماشین حالت متناهی (FSM):**

برای مدل‌سازی رفتار نرم‌افزار از یک ماشین حالت متناهی (FSM) استفاده شده است. حالت‌های اصلی شامل: Start، Authenticated، Visitor Panel، Admin Panel، Game Management و Ticket Purchase هستند.

![FSM](FSM.drawio)

فایل FSM.drawio در GitHubرا در سایت <https://app.diagrams.net/> آپلود کنید و نمودار کشیده شده را مشاهده فرمایید.

**بررسی ویژگی‌های ایمنی و زنده بودن:**

- Safety: سیستم اجازه نمی‌دهد کاربر بدون ورود، بلیت بخرد.

- Liveness: اگر کاربر وارد شود و بازی فعال باشد، می‌تواند بلیت بخرد.

**نتیجه‌گیری:**

در این تمرین با استفاده از روش‌های رسمی، یک سیستم مدیریتی برای شهربازی مدل‌سازی شد. با استفاده از ماشین حالت و منطق گزاره‌ای، رفتارهای صحیح و ناصحیح سیستم بررسی شده‌اند تا نرم‌افزار نهایی از لحاظ ایمنی و عملکرد قابل اطمینان باشد.

**فایل‌های ضمیمه:**

1. FSM.drawio: نمودار ماشین حالات
2. predicates.txt: منطق گزاره‌ای سیستم
3. README.md: توضیحات کلی پروژه