

پیاده سازی سیستم رزرو بلیت سینما

پروژه درس مهندسی نرم افزار دو

استاد: آقای دکتر شجاع

اعضای تیم:

پویا امید

محمد حسین زاده

آریا رافع

سبحان صفرزاده

صبا عفتی

فهرست

۱.	مقدمه	۴
۲.	اهداف پروژه	۴
۳.	ضرورت اجرای پروژه	۵
۴.	محدوده پروژه	۵
۵.	دامنه نرم افزار	۶
۶.	متدولوژی انتخابی برای توسعه نرم افزار	۶
۱.۶.	نقش ها و وظایف اعضای تیم	۷
۲.۶.	ارتباطات	۸
۳.۶.	برنامهریزی	۸
۴.۶.	ابزارهای مدیریت پروژه و همکاری	۹
۷.	چرخه عمر سیستم	۱۰
۸.	مدل سازی (تجزیه و تحلیل نیازمندی ها)	۱۱
۹.	طراحی پایگاه داده:	۲۰
۱۰.	فریم ورک ها و کتابخانه های مورد استفاده	۲۴
۱.۱۰.	بک اند (Backend) و ابزارهای توسعه	۲۴
۲.۱۰.	فرانت اند (Frontend)	۲۴
۱۱.	ساخت و ساز (کدنویسی)	۲۵
۱۲.	طراحی واسط کاربری (UI/UX Design)	۲۵
۱.۱۲.	صفحات اصلی سیستم:	۲۵
۲.۱۲.	بررسی اصول UX در طراحی رابط کاربری (UI) سیستم رزرو بلیت	۲۸
۱۳.	معماری سیستم	۲۹
۱.۱۳.	ارتباط بین لایه ها	۳۰
۱۴.	جزئیات پیاده سازی	۳۰
۱.۱۴.	ماژول حساب کاربری	۳۰
۲.۱۴.	ماژول بلیت فروشی	۳۱
۱.۴.۳.	ویژگی های امنیتی	۳۱
۱.۴.۵.	مدیریت فایل های ایستا و رسانه ای	۳۱
۱۵.	مشکلات و چالش ها	۳۲

۱۵.۱. مشکلات فنی.....	32
۱۵.۲. چالش‌های طراحی و راه‌حل‌ها.....	33
۱۶. ساخت و ساز (آزمایش).....	34
۱.۱۶. سناریوهای آزمایشی و نتایج.....	34
۱۸. نتیجه‌گیری و دستاوردها.....	36
۱۹. دستاوردها نسبت به اهداف اولیه.....	37
۲۰. تأثیر سیستم بر تجربه کاربری و مدیریت سینما.....	37

۱. مقدمه

با گسترش فناوری اطلاعات و رشد سریع خدمات آنلاین، سیستم‌های اطلاعاتی نقش مهمی در تسهیل فرایندهای روزمره زندگی ایفا می‌کنند. یکی از حوزه‌هایی که به شکل چشمگیری تحت تأثیر این تحول قرار گرفته، صنعت سینما و مدیریت رویدادهای فرهنگی است. در گذشته، فرآیند خرید بلیت سینما به صورت حضوری انجام می‌شد که علاوه بر اتلاف وقت، باعث ایجاد صف‌های طولانی و محدودیت در انتخاب صندلی می‌گردید.

سیستم اطلاعاتی رزرو بلیت سینما، راهکاری نوین برای رفع این مشکلات ارائه می‌دهد. این سیستم به کاربران امکان می‌دهد که در هر زمان و از هر مکان، اطلاعات فیلم‌ها و سانس‌های نمایش را مشاهده کرده، صندلی دلخواه خود را انتخاب و بلیت را به صورت آنلاین رزرو یا خریداری کنند. از سوی دیگر، مدیران سینما با استفاده از این سیستم می‌توانند برنامه‌ریزی نمایش فیلم‌ها، مدیریت سالن‌ها، و تحلیل داده‌های فروش را به شکل دقیق‌تری انجام دهند.

در این سیستم تلاش شده تا با استفاده از اصول مهندسی نرم‌افزار، نیازمندی‌های کاربران شناسایی و بر اساس آن یک راه حل کارآمد، قابل اعتماد و کاربرپسند ارائه شود. این پروژه علاوه بر پوشش جنبه‌های فنی، بر تجربه کاربری و امنیت اطلاعات نیز تمرکز دارد تا در نهایت، رضایت کاربران و افزایش بهره‌وری مجموعه‌های سینمایی حاصل شود.

۲. اهداف پروژه

هدف اصلی این پروژه، ایجاد یک سامانه قابل اعتماد و کاربرپسند برای رزرو و خرید بلیت سینما به صورت غیرحضوری است. این سیستم باید هم برای کاربر عادی ساده و قابل فهم باشد، و هم برای مدیران سینما ابزارهای مدیریتی و تحلیلی لازم را فراهم کند. از جمله اهداف جزئی می‌توان به موارد زیر اشاره کرد:

- **سهولت دسترسی کاربران**

امکان مشاهده لیست فیلم‌ها، سانس‌ها و ظرفیت سالن‌ها بدون نیاز به مراجعه حضوری.

- **مدیریت هوشمند ظرفیت سالن‌ها**

کمک به مدیران سینما برای جلوگیری از فروش بیش از ظرفیت و برنامه‌ریزی بهینه.

- **کاهش هزینه‌ها و اتلاف زمان**

حذف نیاز به صف‌های طولانی و حضور فیزیکی برای خرید بلیت.

با پیاده‌سازی این اهداف، سیستم ما نه تنها تجربه کاربری را بهبود می‌دهد، بلکه فرآیندهای مدیریتی سینما را هم سریع‌تر و دقیق‌تر می‌کند.

۳. ضرورت اجرای پروژه

در سال‌های اخیر، الگوی مصرف فرهنگی مردم به سمت استفاده از خدمات آنلاین تغییر کرده است. تجربه نشان داده که بسیاری از مخاطبان سینما ترجیح می‌دهند قبل از حرکت به سمت سالن، از طریق اینترنت فیلم موردنظرشان را انتخاب کرده و صندلی خود را رزرو کنند. این تغییر رفتار، برای سینماها فرصتی ایجاد کرده تا با ارائه خدمات رزرو آنلاین، رضایت مشتریان را افزایش دهند و هم‌زمان هزینه‌های عملیاتی خود را کاهش دهند.

نبود یک سیستم یکپارچه برای مدیریت رزرو و فروش بلیت، باعث بروز مشکلاتی مانند فروش بیش از ظرفیت، ایجاد صف‌های طولانی، و نارضایتی مشتریان می‌شود. همچنین نبود داده‌های دقیق از میزان فروش و سلیقه مخاطبان، برنامه‌ریزی برای اکران فیلم‌ها را دشوار می‌کند.

پیاده‌سازی سیستم اطلاعاتی رزرو بلیت سینما می‌تواند این مشکلات را برطرف کرده و مزایای متعددی ایجاد کند، از جمله:

- افزایش رضایت مخاطبان با ساده‌سازی فرآیند خرید بلیت.
 - بهبود بهره‌وری مدیران سینما.
 - کاهش هزینه‌ها با حذف بخشی از عملیات حضوری و کاغذی.
 - ایجاد رقابت‌پذیری بیشتر در برابر سایر مراکز فرهنگی که خدمات آنلاین ارائه می‌دهند.
- به همین دلیل، طراحی و پیاده‌سازی چنین سامانه‌ای نه تنها یک نیاز تکنولوژیک، بلکه یک الزام برای حفظ و جذب مشتریان در صنعت سینما محسوب می‌شود.

۴. محدوده پروژه

این پروژه بر طراحی و پیاده‌سازی یک سامانه اطلاعاتی برای رزرو و خرید بلیت سینما تمرکز دارد که نسخه تحت وب را پشتیبانی کند. محدوده کار به نحوی تعریف شده که تمام فرآیندهای اصلی رزرو بلیت را پوشش دهد، اما وارد بخش‌های غیرضروری یا فرعی مثل مدیریت مالی کل مجموعه نمی‌شود.

در این سیستم، امکانات زیر به عنوان محدوده اصلی کار در نظر گرفته شده‌اند:

- نمایش اطلاعات فیلم‌ها و سانس‌ها شامل پوستر، خلاصه داستان، مدت زمان، و زمان‌های اکران.
 - نمایش وضعیت سالن‌ها و سانس‌ها.
 - امکان ثبت‌نام و ورود کاربران با احراز هویت ساده و امن.
- این محدوده به گونه‌ای تعریف شده که نیازهای واقعی کاربران و مدیران را برطرف کند و هم در بازه زمانی پروژه قابل پیاده‌سازی باشد.

۵. دامنه نرم افزار

نرم افزار «سیستم رزرو بلیت سینما» به دسته نرم افزارهای تحت وب نزدیک تر است؛ چرا که این سیستم از طریق یک مرورگر وب قابل دسترسی است و با معماری سه لایه (شامل لایه ارائه، منطق کسب و کار و داده) توسعه یافته است که ویژگی اصلی نرم افزارهای تحت وب است.

۶. متدولوژی انتخابی برای توسعه نرم افزار

با توجه به ماهیت پروژه و محدودیت زمانی آن، تیم ما تصمیم گرفت از رویکرد توسعه چابک (Agile) استفاده کند. با در نظر گرفتن ویژگی های این پروژه، چارچوب اسکرامبان (Scrumban) که ترکیبی از اصول اسکرام و کانبان است، به عنوان رویکرد اصلی انتخاب شد. این انتخاب به ما اجازه داد تا از ساختار تکرارشونده اسکرام برای مدیریت زمان و از انعطاف پذیری کانبان برای مدیریت جریان کار بهره ببریم.

در این روش، پروژه به چندین اسپرینت (Sprint) با طول مشخص (معمولاً ۲ تا ۳ هفته) تقسیم شد. در هر اسپرینت، بخشی از قابلیت های سیستم طراحی، پیاده سازی و تست شدند تا در پایان هر دوره، یک نسخه قابل اجرا از نرم افزار در اختیار اسکرام مستر قرار گیرد.

علاوه بر این، برای مدیریت وظایف در داخل هر اسپرینت، از یک تابلوی کانبان بصری استفاده شد که وظایف را به ستون های در حال انجام (In Progress)، در حال تست (Testing) و انجام شده (Done) تقسیم می کرد. این رویکرد به ما کمک کرد تا:

- انعطاف در برابر تغییرات: در صورت تغییر نیازمندی ها، تیم می تواند در میان اسپرینت، تغییرات را به صورت فوری به تابلوی کانبان اضافه و اعمال کند.
- تحویل سریع ارزش به کاربر فرضی: تحویل محصول به صورت افزایشی و در پایان هر اسپرینت، به صورت پیوسته انجام می شد.
- بهبود کیفیت: با استفاده از تابلوی کانبان، فرآیند تست و رفع خطا به صورت مداوم پیگیری می شد.
- همکاری تیمی مؤثر: تابلوی کانبان به عنوان یک ابزار بصری، شفافیت را افزایش می داد و به اعضای تیم اجازه می داد تا به صورت خودسازمان دهنده و با هماهنگی بیشتر کار کنند.

این رویکرد ترکیبی، تعادل خوبی بین نیاز به تحویل سریع، کیفیت مطلوب نرم افزار و سازگاری با محدودیت های زمانی پروژه دانشگاهی برقرار کرد.

نظر بسیار خوبی است. اضافه کردن بخشی در مورد نقش ها و وظایف اعضای تیم، نشان دهنده درک عمیق شما از نحوه مدیریت یک پروژه نرم افزاری در یک محیط چابک است. این بخش، گزارش شما را از یک مستند فنی صرف، به یک گزارش کامل تر در زمینه مهندسی نرم افزار تبدیل می کند.

۱.۶. نقش‌ها و وظایف اعضای تیم

با پیروی از چارچوب اسکرامبان، تیم توسعه پروژه بر اساس نقش‌های زیر سازماندهی شد و هر یک وظایف مشخصی را بر عهده داشتند:

مدیر محصول (Product Owner) :

- **وظایف:** مدیریت نیازمندی‌های پروژه، اولویت‌بندی قابلیت‌ها در بک‌لاگ، جمع‌آوری بازخورد از ذی‌نفعان (فرضی) و اطمینان از هم‌راستایی محصول با اهداف کسب‌وکار.
- **تخصص:** تمرکز بر روی کارکرد سیستم و تجربه کاربری (UI/UX)
- **نام:** فرضی

اسکرام مستر (Scrum Master) :

- **وظایف:** تسهیل فرآیند توسعه چابک، برگزاری جلسات روزانه، رفع موانع فنی و مدیریتی، و آموزش تیم در زمینه اصول اسکرامبان.
- **تخصص:** تمرکز بر روی فرآیند، هماهنگی و حل مشکلات تیمی.
- **نام:** صبا عفتی

توسعه‌دهنده بک‌اند (Backend Developer) :

- **وظایف:** طراحی و پیاده‌سازی منطق اصلی سیستم با استفاده از Django ، مدل‌سازی پایگاه داده، مدیریت ارتباطات با دیتابیس و پیاده‌سازی فرآیندهای حیاتی مانند رزرو بلیت.
- **تخصص:** برنامه‌نویسی سمت سرور و مدیریت پایگاه داده.
- **نام:** آریا رافع، سبحان صفرزاده، صبا عفتی

توسعه‌دهنده فرانت‌اند (Frontend Developer) :

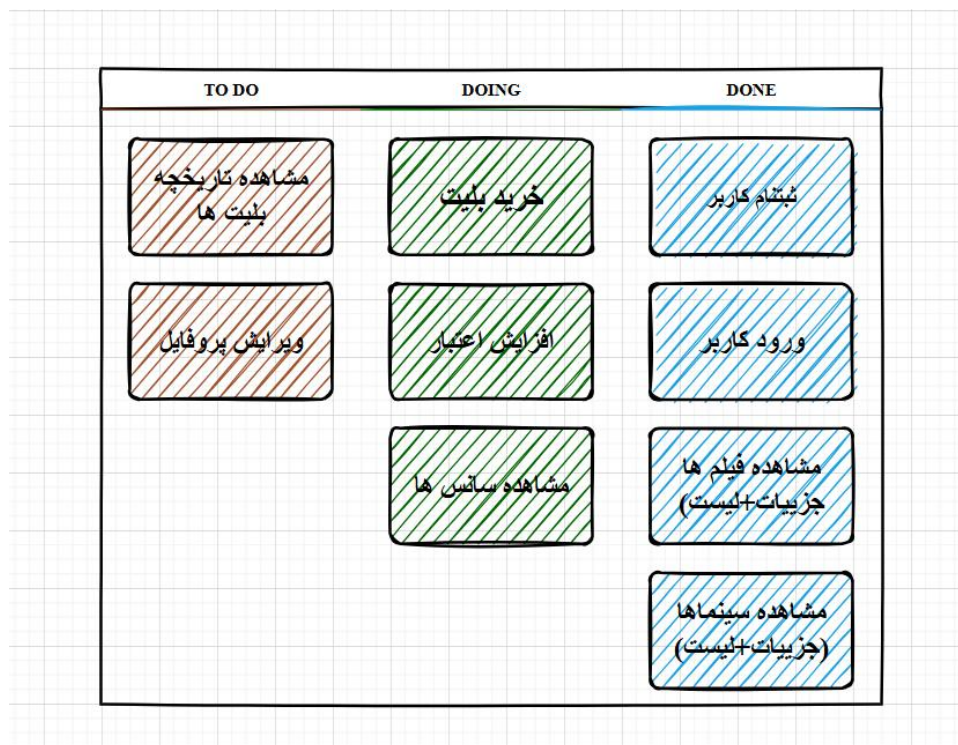
- **وظایف:** طراحی و پیاده‌سازی رابط کاربری با HTML ، CSS و JavaScript ، اطمینان از واکنش‌گرایی (Responsiveness) سیستم و بهبود تجربه کاربری بصری.
- **تخصص:** طراحی رابط کاربری و تعاملات سمت کاربر.
- **نام:** پویا امید، محمد حسین زاده

این ساختار نقش‌ها، به تیم کمک کرد تا با وجود محدودیت‌های پروژه، به صورت منظم و کارآمد به اهداف خود دست یابد.

۲.۶. ارتباطات

در چارچوب اسکرامبان، ارتباطات بر اساس شفافیت و هماهنگی مداوم است. برای این پروژه، فرآیند ارتباطات به شکل زیر بوده است:

- **جلسات روزانه:** تیم هر روز جلسه کوتاهی برگزار می‌کرد تا هر فرد گزارش دهد که چه کاری انجام داده، چه کاری را قرار است انجام دهد و با چه موانعی روبرو است. این کار به هماهنگی سریع و حل مشکلات کمک می‌کرد.
- **بازخورد بصری:** تابلوی کانبان به عنوان یک ابزار ارتباطی، وضعیت هر وظیفه را به صورت لحظه‌ای به تمام اعضای تیم نشان می‌داد و نیاز به جلسات طولانی و گزارش‌های کتبی را کاهش می‌داد.



۳.۶. برنامه‌ریزی

فعالیت برنامه‌ریزی به صورت چابک و در سه سطح انجام می‌گرفت:

- **برنامه‌ریزی محصول (Product Planning):** در این مرحله، تمامی نیازمندی‌ها (مانند ثبت‌نام، مشاهده فیلم‌ها و خرید بلیت) در یک بک‌لاگ (Backlog) کلی جمع‌آوری می‌شد.
- **برنامه‌ریزی اسپرینت (Sprint Planning):** در ابتدای هر اسپرینت، تیم از میان وظایف بک‌لاگ، مهم‌ترین موارد را انتخاب کرده و برای تکمیل آن‌ها در طول اسپرینت برنامه‌ریزی می‌کرد.

- **برنامه‌ریزی تکراری (Iterative Planning):** در پایان هر اسپرینت، با دریافت بازخورد و ارزیابی پیشرفت، برنامه‌ریزی برای اسپرینت بعدی بهبود می‌یافت.

۴.۶. ابزارهای مدیریت پروژه و همکاری

- برای مدیریت فرآیند توسعه چابک و تسهیل همکاری میان اعضای تیم، از ابزارها و پلتفرم‌های زیر استفاده شد:
- **گیت‌هاب (GitHub):** این پلتفرم به عنوان سیستم کنترل نسخه اصلی پروژه به کار رفت. تمام کدهای پروژه در مخزن (Repository) گیت‌هاب نگهداری می‌شدند و هر یک از اعضای تیم، تغییرات خود را به صورت منظم با استفاده از commit و push ثبت می‌کرد. این کار چندین مزیت داشت:
 - **کنترل نسخه:** امکان پیگیری و بازگشت به نسخه‌های قبلی کد را فراهم کرد.
 - **همکاری:** به چندین توسعه‌دهنده اجازه داد تا به صورت همزمان بر روی بخش‌های مختلف پروژه کار کنند و تغییرات خود را به صورت ایمن با هم ادغام کنند.
 - **مدیریت مسائل (Issue Tracking):** از قابلیت Issues گیت‌هاب برای ثبت و پیگیری باگ‌ها و وظایف کوچک استفاده شد.

Profile not properly created during user registration #1

Edit New issue

Open

SabaEdr opened now

Owner

وارد صفحه ثبت‌نام بشید.
تمامی فیلدهای الزامی (نام، ایمیل، رمز عبور و ...) را پر کرده و فرم را ارسال کنید.
پروفایل کاربر ایجاد شده را در پایگاه داده یا صفحه پروفایل کاربر بررسی کنید.
نتیجه مورد انتظار:
یک پروفایل کامل برای کاربر ایجاد شود و تمام فیلدها به درستی پر شده باشند (نام، ایمیل، تصویر پروفایل، موجودی و ...).

نتیجه واقعی:
پروفایل کاربر به صورت ناقص ایجاد می‌شود و برخی فیلدها به درستی مقداردهی نشده‌اند.
این مشکل برای همه کاربران تازه ثبت‌نام شده رخ می‌دهد.

Create sub-issue

SabaEdr assigned BERRORCODE and SabaEdr now

Assignees

BERRORCODE

SabaEdr

Labels

No labels

Projects

No projects

Milestone

No milestone

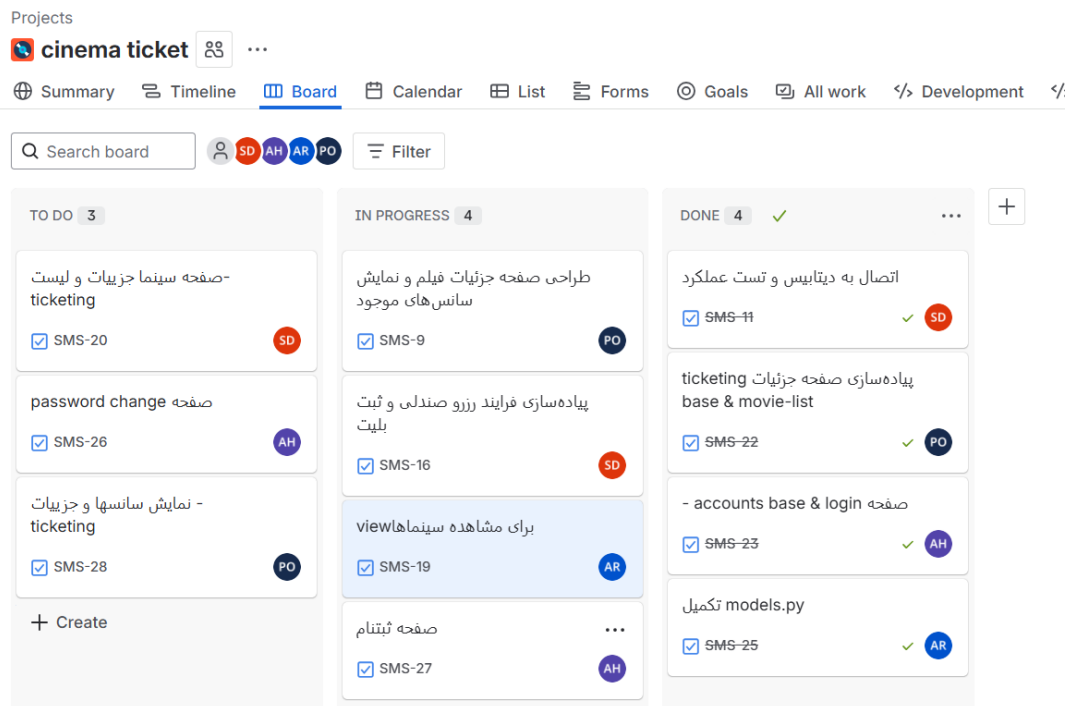
Relationships

None yet

- **جیرا (Jira):** از جیرا به عنوان ابزار مدیریت پروژه چابک استفاده شد تا فرآیند اسکرام‌بان به صورت بصری و سازمان‌یافته مدیریت شود.

- **تابلوی کانبان:** تابلوی کانبان در جیرا برای تجسم جریان کار مورد استفاده قرار گرفت که شامل ستون‌های To Do, In Progress, Testing, و Done بود.
- **مدیریت وظایف:** هر وظیفه و نیازمندی پروژه به صورت یک Task یا Story در جیرا تعریف می‌شد و به اعضای تیم اختصاص داده می‌شد.

- شفافیت: وضعیت هر وظیفه به صورت لحظه‌ای برای تمام اعضای تیم و مدیران پروژه قابل مشاهده بود، که این امر به افزایش شفافیت و هماهنگی کمک می‌کرد.



استفاده از این ابزارها، به تیم این امکان را داد که با وجود اینکه اعضای تیم به صورت فیزیکی در یک مکان نبودند، فرآیند توسعه را به صورت یکپارچه و کارآمد مدیریت کنند.

۷. چرخه عمر سیستم (SDLC)

چرخه عمر توسعه سیستم (System Development Life Cycle) مجموعه‌ای از مراحل متوالی است که از شناسایی نیازمندی‌ها آغاز می‌شود و تا تحویل، پشتیبانی و بهبود سیستم ادامه پیدا می‌کند. برای پروژه «سیستم اطلاعاتی رزرو بلیت سینما»، ما مراحل SDLC را با رویکرد چابک پیاده‌سازی کردیم، اما همچنان ساختار کلی چرخه عمر را رعایت کردیم.

مراحل اصلی چرخه عمر در این پروژه به صورت زیر بوده است:

۱. بررسی و تحلیل نیازمندی‌ها

در این مرحله با استفاده از بررسی سیستم‌های مشابه، نیازهای کاربران و مدیران سینما شناسایی شد. خروجی این مرحله یک سند نیازمندی‌ها (SRS) بود.

2. طراحی سیستم

بر اساس نیازمندی‌ها، مدل‌های مورد نیاز Use Case، دیاگرام‌های UML، طراحی پایگاه داده و واسط کاربری (تهیه شد. طراحی در دو سطح کلی (High-level design) و جزئی (Detailed design) انجام شد.

3. پیاده‌سازی (Implementation)

کدنویسی سیستم در چندین اسپرینت انجام شد. هر اسپرینت شامل توسعه بخشی از قابلیت‌ها و یک نسخه قابل تست بود.

4. تست و ارزیابی

پس از پایان هر اسپرینت، تست سیستمی (System Test) انجام شد تا عملکرد صحیح قابلیت‌ها و عدم وجود باگ‌های اساسی تضمین شود.

مزیت استفاده از SDLC در این پروژه، ایجاد یک مسیر روشن برای توسعه و اطمینان از این بود که هیچ مرحله‌ای از نظر تحلیلی یا فنی نادیده گرفته نشود، حتی اگر اجرای آن با متدولوژی چابک صورت گیرد.

۸. مدل‌سازی (تجزیه و تحلیل نیازمندی‌ها)

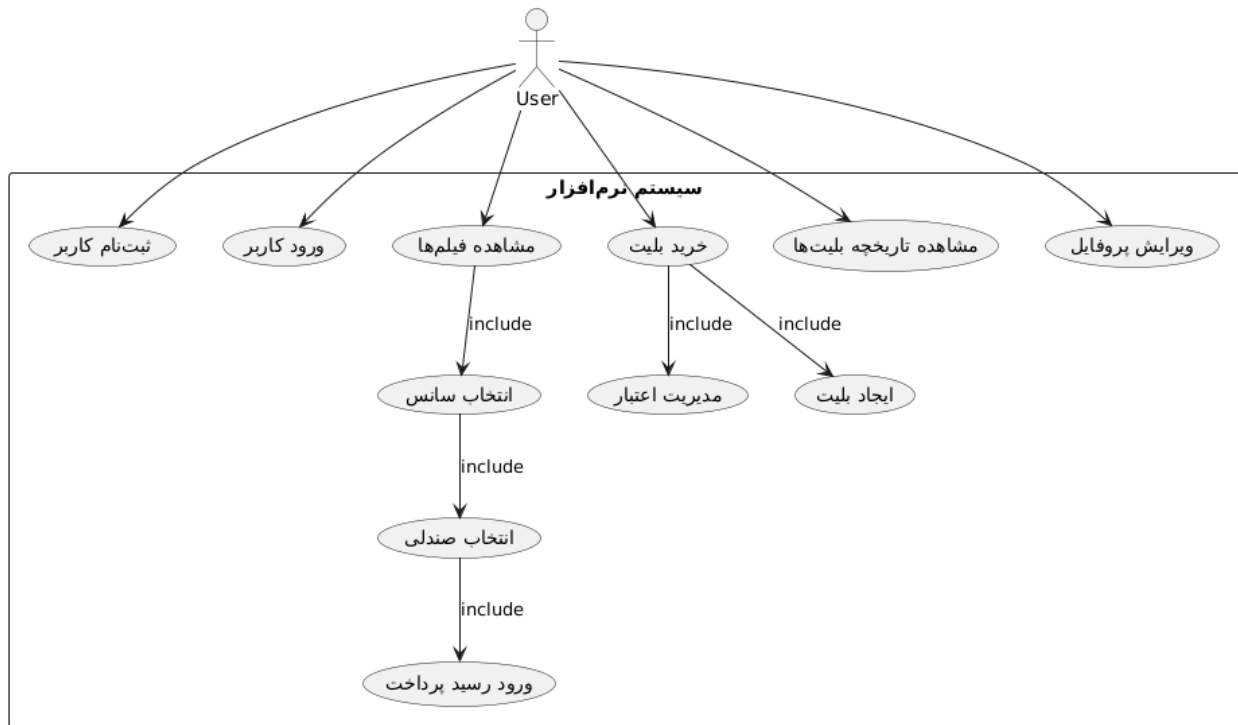
بر اساس تحلیل نیازمندی‌ها و پیاده‌سازی انجام‌شده در پروژه «سیستم اطلاعاتی رزرو بلیت سینما»، مدل‌های پایگاه داده و ساختار کلی سیستم به کمک زبان UML مستندسازی شده‌اند. این مستندات به توسعه‌دهندگان و مدیران پروژه کمک می‌کند تا ارتباط بین اجزای سیستم را بهتر درک کنند و در توسعه‌های پیش رو، مسیر مشخصی داشته باشند.

۸.۱ Use Case Diagram

نقش‌ها (Actors) و عملکردهای اصلی سیستم شامل:

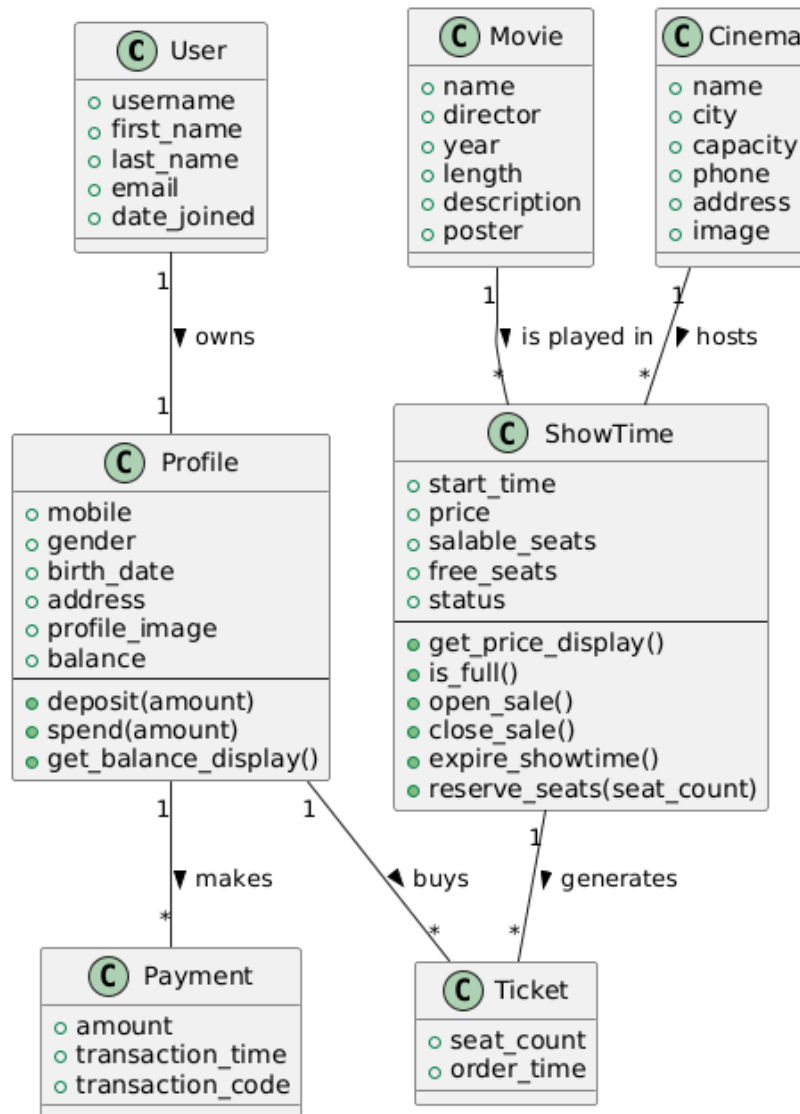
- کاربر مهمان: مشاهده لیست فیلم‌ها، مشاهده سینماها.
- کاربر ثبت‌نام‌شده: رزرو صندلی، خرید بلیت، مشاهده تاریخچه خرید.

- سیستم پرداخت: افزایش اعتبار کاربر.



۸.۲. Class Diagram

- Profile: اطلاعات کاربری شامل نام، جنسیت، تاریخ تولد، آدرس، تصویر پروفایل، و موجودی اعتبار.
- Payment: تراکنش‌های مالی انجام شده توسط کاربر، شامل مبلغ، زمان، و کد تراکنش.
- Movie: اطلاعات فیلم شامل عنوان، کارگردان، سال تولید، مدت زمان، توضیح و پوستر.
- Cinema: اطلاعات سینما شامل نام، شهر، ظرفیت، آدرس و تصویر.
- Showtime: اطلاعات هر سانس شامل فیلم، سینما، زمان شروع، قیمت، تعداد صندلی‌های قابل فروش و وضعیت فروش.
- Ticket: بلیت‌های خریداری شده توسط کاربر برای یک سانس خاص، شامل تعداد صندلی و زمان خرید.



روابط اصلی بین کلاس‌ها:

1. ارتباط بین Profile و Payment

هر کاربر (Profile) می‌تواند چندین تراکنش مالی (Payment) انجام دهد. این تراکنش‌ها می‌توانند شامل افزایش اعتبار یا سایر پرداخت‌های مرتبط با خرید بلیت باشند. این رابطه از نوع یک به چند (One-to-Many) است؛ یعنی یک پروفایل می‌تواند به چند رکورد پرداخت متصل باشد، اما هر پرداخت فقط به یک پروفایل تعلق دارد.

2. ارتباط بین Profile و Ticket

هر کاربر می‌تواند چندین بلیت خریداری کند. این بلیت‌ها ممکن است مربوط به فیلم‌ها، سانس‌ها و تاریخ‌های متفاوت باشند. این رابطه هم یک به چند است؛ یک پروفایل ممکن است به صدها بلیت مرتبط باشد، اما هر بلیت فقط به یک کاربر تعلق می‌گیرد.

3. ارتباط بین Ticket و ShowTime

هر بلیت مربوط به یک سانس مشخص است. این سانس اطلاعاتی مانند فیلم، سالن نمایش، ساعت شروع و وضعیت فروش را در خود دارد. رابطه بین بلیت و سانس از نوع چند به یک است؛ چندین بلیت می‌توانند برای یک سانس صادر شوند، اما هر بلیت فقط به یک سانس تعلق دارد.

4. ارتباط بین ShowTime و Movie

هر سانس فقط یک فیلم را نمایش می‌دهد. این رابطه چند به یک است؛ ممکن است چند سانس برای یک فیلم وجود داشته باشد، اما هر سانس به یک فیلم خاص متصل است.

5. ارتباط بین ShowTime و Cinema

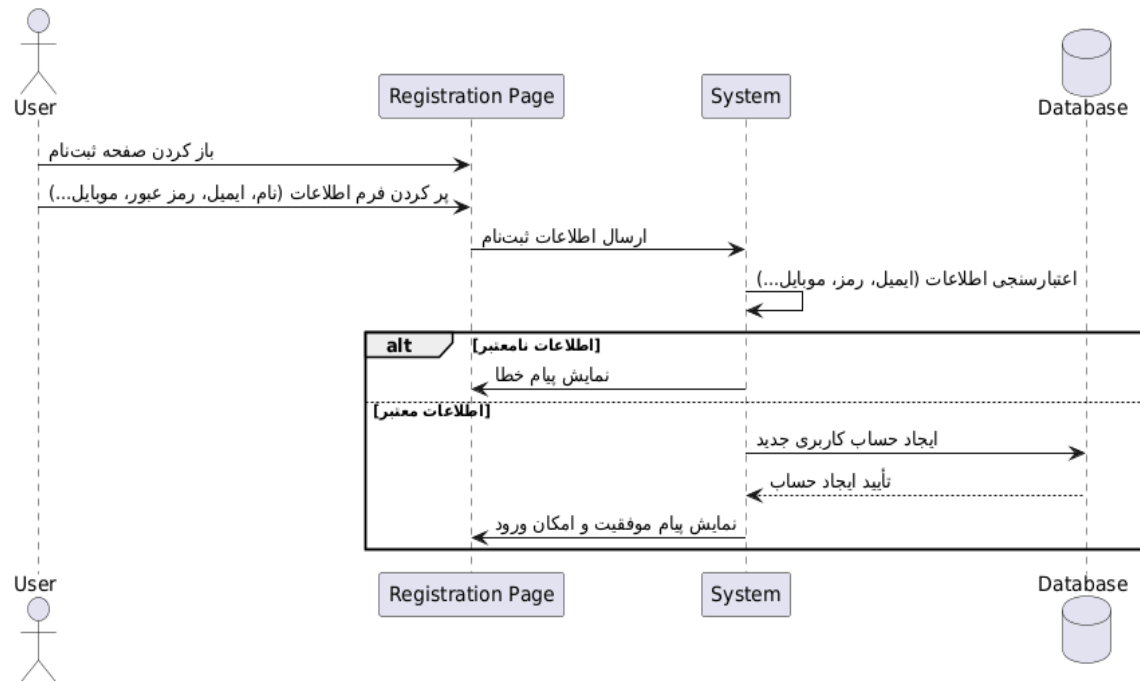
هر سانس در یک سینمای مشخص برگزار می‌شود. این رابطه هم چند به یک است؛ ممکن است چند سانس مختلف در یک سینما برگزار شود، اما هر سانس فقط در یک سینما انجام می‌شود.

۸.۳. Sequence Diagram

۱. ثبت‌نام کاربر (User Registration)

فرآیند ثبت‌نام کاربر جدید در سیستم:

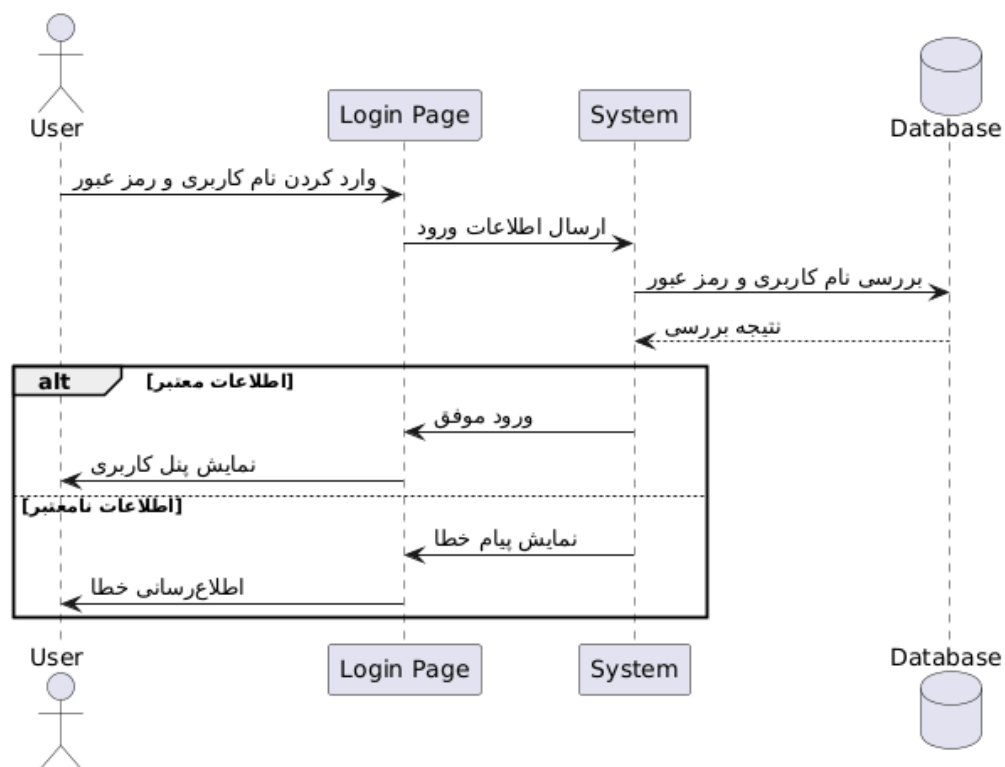
1. کاربر وارد صفحه ثبت‌نام می‌شود.
2. فرم اطلاعات شخصی (نام، ایمیل، رمز عبور و...) را پر می‌کند.
3. سیستم صحت اطلاعات را بررسی می‌کند (فرمت ایمیل، رمز عبور، شماره موبایل و...).
4. در صورت درست بودن اطلاعات، حساب کاربری ایجاد و ذخیره می‌شود.
5. سیستم پیام موفقیت و امکان ورود به کاربر نمایش می‌دهد.



۲. ورود به سیستم

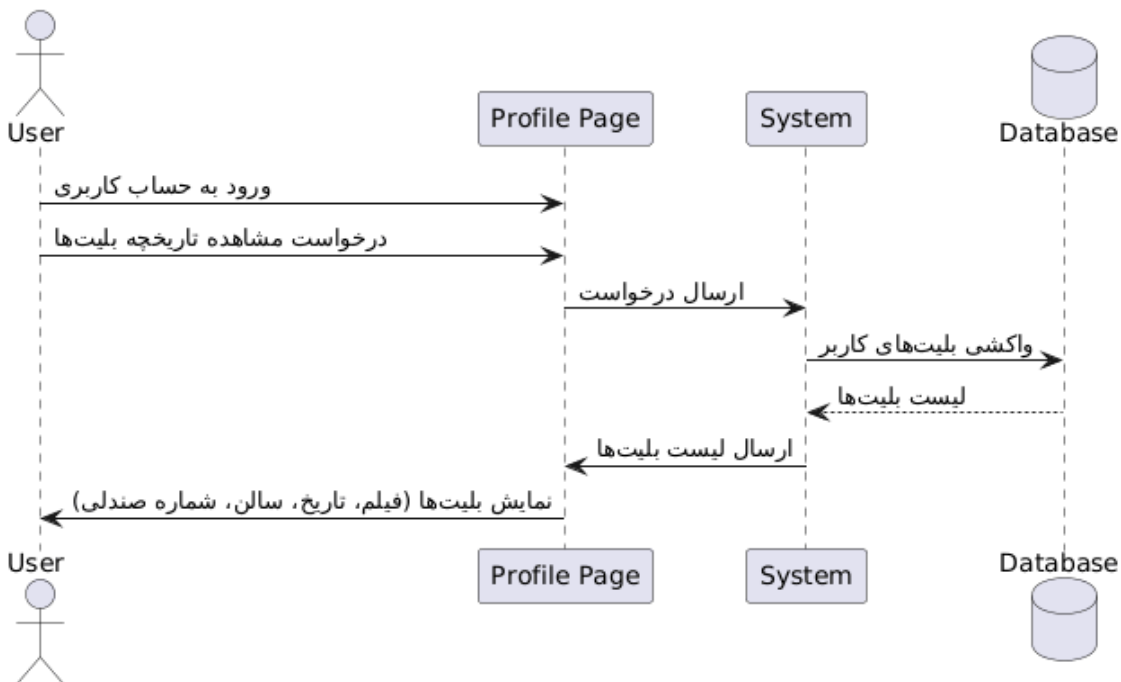
فرآیند ورود کاربر ثبت نام شده:

1. کاربر نام کاربری و رمز عبور وارد می کند.
2. سیستم اطلاعات ورود را بررسی می کند.
3. اگر معتبر باشد، کاربر وارد سیستم می شود و پنل خود را می بیند.
4. در صورت خطا، پیام مناسب نمایش داده می شود.



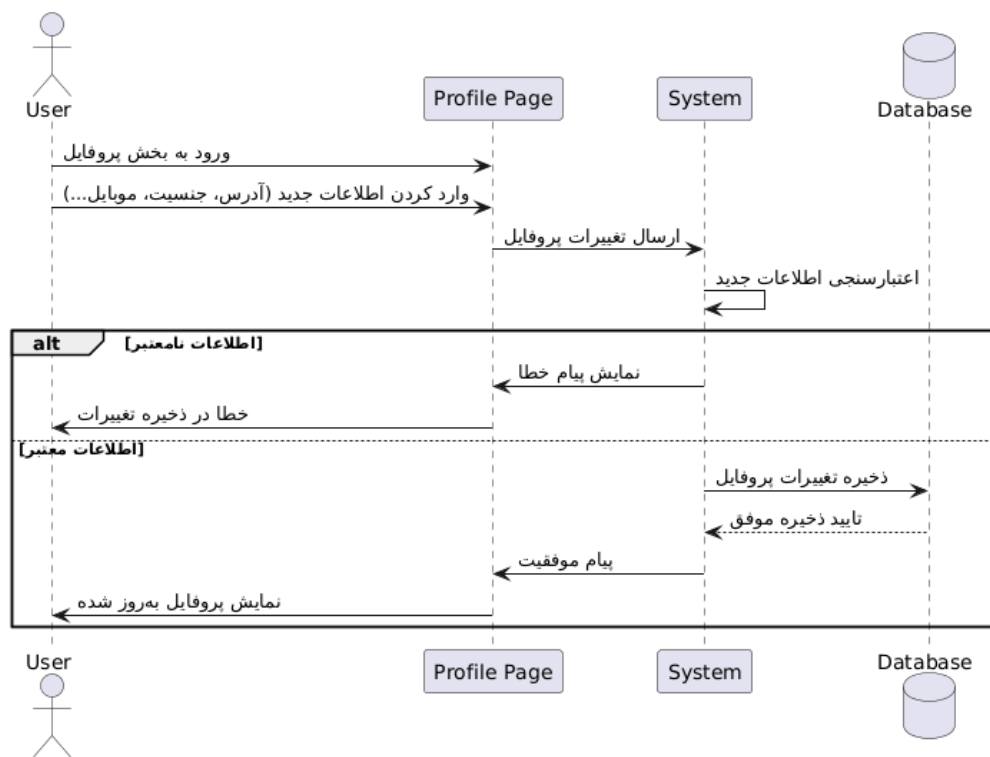
۳. مشاهده تاریخچه بلیت ها

1. کاربر وارد حساب کاربری خود می شود.
2. درخواست مشاهده لیست خریدها را ارسال می کند.
3. سیستم لیست بلیت ها را از پایگاه داده استخراج می کند.
4. لیست بلیت های قبلی به همراه جزئیات (فیلم، تاریخ، سالن، شماره صندلی) نمایش داده می شود.



۴. ویرایش پروفایل کاربر (Edit Profile)

1. کاربر وارد بخش پروفایل خود می‌شود.
2. اطلاعات جدید (مثل آدرس، جنسیت، شماره موبایل) را وارد می‌کند.
3. سیستم تغییرات را بررسی و ذخیره می‌کند.
4. پیام موفقیت نمایش داده می‌شود.



Swimlane Diagram .۴.۸

فرآیند خرید بلیت می‌تواند به صورت یک Swimlane Diagram مدل‌سازی شود که بازیگران اصلی و تعاملات آن‌ها را نشان می‌دهد.

- Swimlane (کاربر): مشاهده فیلم‌ها - انتخاب سانس - انتخاب صندلی - ورود اطلاعات پرداخت
- Swimlane (سیستم نرم‌افزار): نمایش اطلاعات - بررسی موجودی صندلی‌ها - ایجاد تراکنش - ذخیره بلیت



۸.۵. CRC Cards (ShowTime)

یک کارت CRC (Class, Responsibilities, Collaborators) برای کلاس ShowTime به شکل زیر است:

- کلاس ShowTime :
- مسئولیت ها :
 - نگهداری اطلاعات یک سانس نمایش (فیلم، سینما، زمان)
 - مدیریت تعداد صندلی های آزاد و قابل فروش
 - تغییر وضعیت سانس (مثلاً open_sale() یا close_sale())
- همکاران:

Movie, Cinema, Ticket

C ShowTime
Responsibilities
<ul style="list-style-type: none"> ■ نگهداری اطلاعات سانس (فیلم، سینما، زمان) □ مدیریت تعداد صندلی‌های آزاد و قابل فروش ■ تغییر وضعیت سانس (open_sale(), close_sale(), expire_showtime())
Collaborators
<ul style="list-style-type: none"> ○ Movie ○ Cinema ○ Ticket

۹. طراحی پایگاه داده:

پایگاه داده سیستم «رزرو بلیت سینما» بر اساس مدل سازی شیء‌گرا (OOP) در Django طراحی شده و از ساختار رابطه‌ای (Relational Database) استفاده می‌کند. هر جدول در پایگاه داده متناظر با یک Model در کد است. این طراحی به گونه‌ای انجام شده که تمام نیازمندی‌های ثبت اطلاعات فیلم‌ها، سینماها، سانس‌ها، کاربران و تراکنش‌ها را پوشش دهد.

۱. Profile

- کاربرد: نگهداری اطلاعات شخصی و مالی کاربران سیستم.

- فیلدها:

- User: ارتباط یک‌به‌یک با مدل کاربر پیش فرض Django نام، ایمیل، تاریخ عضویت و...
- Mobile: شماره تلفن همراه.
- Gender: جنسیت (مرد یا زن).
- birth_date: تاریخ تولد.
- address: آدرس محل سکونت.
- profile_image: تصویر پروفایل کاربر.
- balance: موجودی اعتبار کاربر.

- رفتارها:

- get_balance_display(): نمایش موجودی به صورت متنی با واحد تومان.

- deposit(amount) افزایش موجودی اعتبار.
- spend(amount) کاهش موجودی در صورت کافی بودن اعتبار.

۲. Payment

- کاربرد: ثبت تراکنش‌های مالی کاربران.
- فیلدها:
 - profile: ارتباط با پروفایل کاربر.
 - amount: مبلغ تراکنش.
 - transaction_time: زمان انجام تراکنش (به صورت خودکار ذخیره می‌شود).
 - transaction_code: کد پیگیری تراکنش.

۳. Movie

- کاربرد: نگهداری اطلاعات فیلم‌های موجود در سیستم.
- فیلدها:
 - name: عنوان فیلم.
 - director: نام کارگردان.
 - year: سال تولید.
 - length: مدت زمان نمایش (دقیقه).
 - description: خلاصه توضیح فیلم.
 - poster: تصویر پوستر فیلم.

۴. Cinema

- کاربرد: ذخیره اطلاعات سینماها و سالن‌ها.

- **فیلدها:**

- name : نام سینما.
- city: شهر محل سینما.
- capacity: ظرفیت سالن.
- phone: شماره تماس.
- address: آدرس سینما.
- Image: تصویر سینما.

۵. ShowTime

- **کاربرد:** مدیریت اطلاعات سانس‌های نمایش فیلم.

- **فیلدها:**

- Movie : ارتباط با مدل Movie
- Cinema : ارتباط با مدل Cinema
- start_time : زمان شروع نمایش.
- Price : قیمت بلیت.
- salable_seats : تعداد صندلی‌های قابل فروش.
- free_seats : تعداد صندلی‌های خالی.
- Status : وضعیت فروش (شروع نشده، در حال فروش، فروش کامل، بسته‌شده، پخش‌شده، لغوشده).

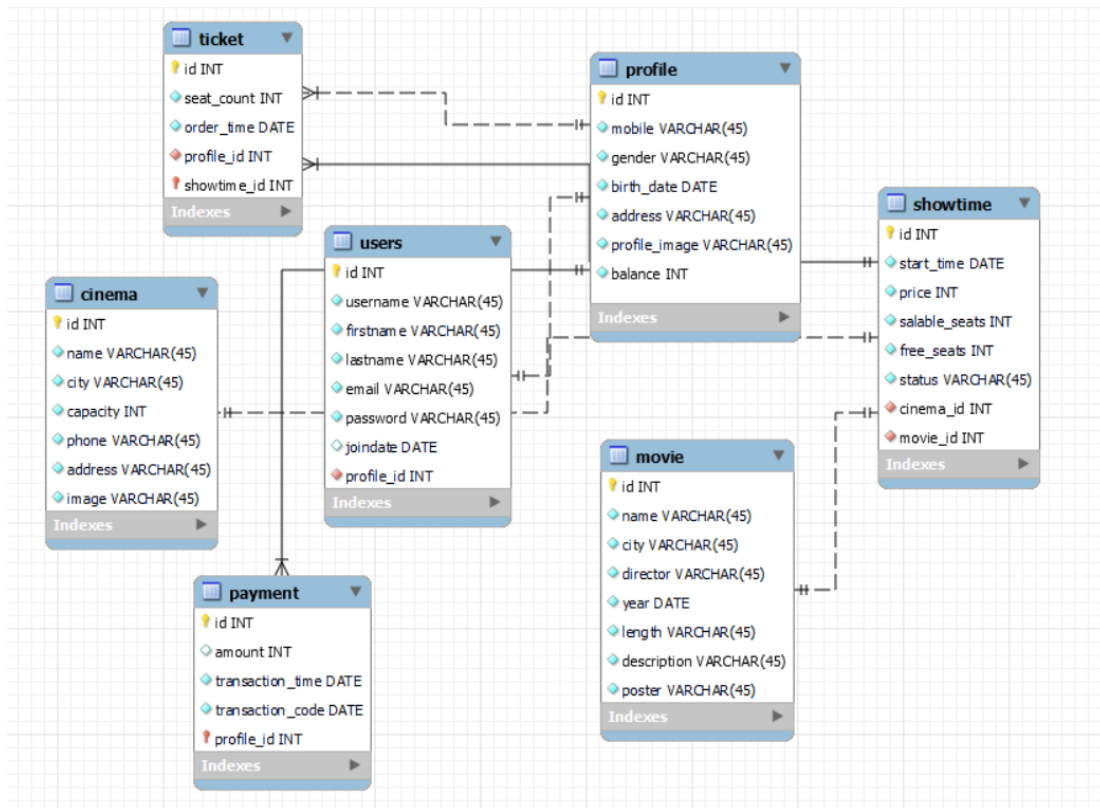
- **رفتارها (متدها):**

- get_price_display() نمایش قیمت بلیت با واحد تومان.
- is_full() بررسی پر بودن سالن.
- open_sale() باز کردن فروش بلیت.
- close_sale() بستن فروش بلیت.

- expire_showtime(is_canceled) تغییر وضعیت سانس به «پخش شده» یا «لغو شده».
- reserve_seats(seat_count) رزرو یک یا چند صندلی برای مشتری.

6. Ticket

- کاربرد: ثبت بلیت‌های خریداری شده توسط کاربران.
- فیلدها:
 - Showtime: ارتباط با مدل ShowTime.
 - Customer: ارتباط با مدل Profile خریدار بلیت
 - seat_count: تعداد صندلی‌های رزرو شده.
 - order_time: زمان خرید بلیت.



۱۰. فریم‌ورک‌ها و کتابخانه‌های مورد استفاده

این پروژه بر اساس یک معماری چندلایه و با استفاده از فناوری‌های متن‌باز برای هر دو بخش بک‌اند و فرانت‌اند توسعه یافته است. در طراحی و پیاده‌سازی این سیستم از فریم‌ورک‌ها و کتابخانه‌های زیر استفاده شده است:

۱۰.۱. بک‌اند (Backend) و ابزارهای توسعه

فهرست زیر، کتابخانه‌های نصب‌شده در محیط مجازی پروژه را نشان می‌دهد:

Django 5.2.4 : فریم‌ورک اصلی پروژه است که برای توسعه سریع و امن وب استفاده شده است.

asgiref 3.9.1 : یک کتابخانه برای پشتیبانی از ASGI (Asynchronous Server Gateway Interface) است که در نسخه‌های جدید Django به صورت پیش‌فرض استفاده می‌شود.

sqlparse 0.5.3 : این کتابخانه برای تجزیه و قالب‌بندی دستورات SQL استفاده می‌شود و Django نیز از آن بهره می‌برد.

Pillow 11.3.0 : کتابخانه‌ای قدرتمند برای پردازش تصویر است که برای مدیریت آپلود عکس‌ها در مدل‌های Movie و Profile ضروری است.

django-widget-tweaks 1.5.0 : این کتابخانه به توسعه‌دهنده اجازه می‌دهد تا کلاس‌های CSS و ویژگی‌های HTML را به ویجت‌های فرم‌های جنگو اضافه کند، که باعث انعطاف‌پذیری بیشتر در طراحی فرانت‌اند می‌شود.

tzdata 2025.2 : این کتابخانه اطلاعات مربوط به مناطق زمانی را فراهم می‌کند که برای مدیریت دقیق زمان و تاریخ در Django کاربرد دارد.

۱۰.۲. فرانت‌اند (Frontend)

فناوری‌های فرانت‌اند که در فایل‌های پروژه استفاده شده‌اند:

- HTML، CSS و JavaScript: این پروژه از زبان‌های اصلی وب برای ساختار، ظاهر و عملکرد تعاملی صفحات استفاده می‌کند.
- Bootstrap: برای طراحی واکنش‌گرا (Responsive) و استفاده از کامپوننت‌های آماده، از فریم‌ورک CSS Bootstrap استفاده شده است.
- jQuery: این کتابخانه جاوااسکریپت برای ساده‌سازی دستکاری DOM و مدیریت رویدادها در فرانت‌اند به کار رفته است.

- Font Awesome: برای نمایش آیکون‌ها در صفحات وب، از کتابخانه Font Awesome استفاده شده است.
- IRANSansWebFaNum: از این فونت فارسی برای بهبود تجربه خوانایی و نمایش صحیح متون فارسی در سیستم استفاده شده است.

۱۱. ساخت و ساز (کدنویسی)

فرآیند کدنویسی با رعایت اصول مهندسی نرم‌افزار به صورت زیر انجام شد:

- یکی از اصول بنیادین مهندسی نرم‌افزار، جداسازی وظایف مختلف در ماژول‌های مستقل است. این پروژه با تقسیم‌بندی به دو اپلیکیشن مجزا، این اصل را به خوبی پیاده‌سازی کرده است.
- اپلیکیشن **accounts**: مسئول تمام فرآیندهای مرتبط با مدیریت کاربران شامل ثبت‌نام، ورود، مدیریت پروفایل و تراکنش‌های مالی است.
- اپلیکیشن **ticketing**: مسئول فرآیندهای مربوط به نمایش فیلم‌ها، سانس‌ها و فروش بلیت است.
- رعایت اصل **DRY (Don't Repeat Yourself)**: کدهای تکراری در قالب توابع و کلاس‌های مشترک بازنویسی شدند تا از تکرار کد جلوگیری شود.
- کد خوانا و مستند**: کدها با استفاده از نام‌گذاری معنی‌دار برای متغیرها و توابع و همچنین با افزودن توضیحات، خوانا و قابل درک برای سایر توسعه‌دهندگان نوشته شده‌اند.


۱۲. طراحی واسط کاربری (UI/UX Design)

واسط کاربری سیستم «رزرو بلیت سینما» با هدف ایجاد تجربه‌ای سریع و کاربرپسند طراحی شده است. در این طراحی تلاش شده تا کاربران بتوانند بدون پیچیدگی، مسیر خرید بلیت را از مشاهده فیلم‌ها تا دریافت تأیید خرید بلیت طی کنند. اصول اصلی طراحی شامل سادگی، وضوح، و دسترسی سریع به اطلاعات بوده است.

۱.۱۲. صفحات اصلی سیستم:

۱. صفحه اصلی (Home Page)

- نمایش لیست فیلم‌های در حال اکران به همراه پوستر، عنوان و مدت زمان.


[صبا عفتی](#)
[خرید بلیت](#)
[سینماها](#)
[فیلمها](#)
[فهرست فیلمها](#)

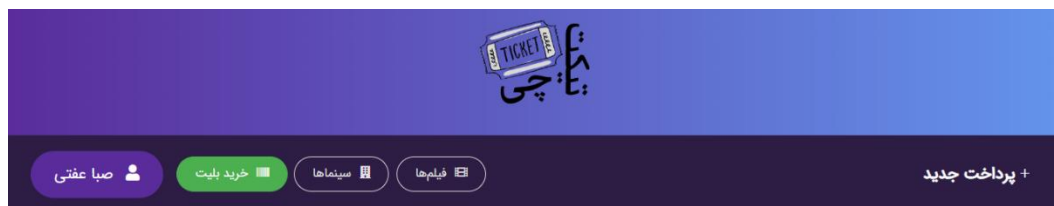
تمساح خونی
ﷺ جواد عزتی

رها
حسام فرهمند

صبحانه با زرافه‌ها
🌞 فروش صحت

📅 محصول سال: ۱۳۸۹

3. افزایش اعتبار



اطلاعات پرداخت را وارد کنید

مبلغ
۱۰۰۰۰۰

رسید تراکنش
#bank-۱۰۰۰۰۰-UHB۴۵۴GRHY۳BDYU

✓ ثبت پرداخت

→ بازگشت

4. پروفایل کاربری

- نمایش تاریخچه خرید بلیت‌ها به همراه جزئیات فیلم، تاریخ و شماره صندلی.
- امکان ویرایش اطلاعات پروفایل و مشاهده موجودی اعتبار.



اطلاعات شخصی

ایمیل: sabadaryuni۴۸@gmail.com
تاریخ تولد: ۲۰۰۴/۱۷/۲۳
تلفن همراه: ۰۹۹۰۷۵۴۰۳۰۱
اعتبار: ۱۸۴۰۰۰۰ تومان

+ افزایش

آدرس

ارومیه

صبا عفتی
admin@

ویرایش نمایه

تغییر گذرواژه

5. مشاهده سانس های موجود

The screenshot shows a movie booking website. At the top, there's a navigation bar with a logo and buttons for 'صفحه اصلی' (Home), 'خرید بلیت' (Buy Ticket), 'سینماها' (Cinemas), and 'فیلمها' (Movies). Below this is a search bar with filters for 'فقط سانس های قابل خرید' (Only purchasable showtimes), 'دقیقه' (Minutes), and 'حداقل زمان' (Minimum time). A search button labeled 'جستجو' (Search) is present. Below the search bar, a message states '۲ سانس نمایش یافت شد.' (2 showtimes found). Two movie listings are shown: 'در آغوش درخت' (Embrace the Tree) and 'صبحانه با زرافه ها' (Breakfast with Giraffes). Each listing includes the director's name, showtime, and a 'فروش آغاز نشده' (Sale not started) button.

6. خرید بلیت

The screenshot shows the movie details page for 'صبحانه با زرافه ها' (Breakfast with Giraffes). The top navigation bar includes 'صفحه اصلی' (Home), 'فهرست سانس ها' (Showtimes), and 'صبحانه با زرافه ها (یکشنبه ساعت ۱۸:۰۰)' (Breakfast with Giraffes (Monday 18:00)). The movie details section includes the title, director (Soroush Shah), showtime (18:00), and location (Arumiyeh, Khayaban Khyam Jonubi). A '۶ صندلی مانده' (6 seats left) badge is visible. The ticket price is listed as 'قیمت بلیت: ۶۰۰۰۰ تومان' (Ticket price: 60,000 Tomans). A 'خرید' (Buy) button is at the bottom.

۲.۱۲. بررسی اصول UX در طراحی رابط کاربری (UI) سیستم رزرو بلیت

- رنگ بندی مناسب و کنتراست بالا : استفاده از پس زمینه تیره و رنگ های شاد و متضاد مانند سبز برای دکمه "خرید بلیت" و رنگ سفید برای متن ها، باعث خوانایی بالا و جلب توجه کاربر به عناصر مهم و کلیدی شده است.

- ناوبری ساده و در دسترس : منوی اصلی (شامل سینماها و فیلم‌ها) به صورت کاملاً واضح در بالای صفحه قرار گرفته و دسترسی سریع به بخش‌های اصلی را فراهم می‌کند. دکمه‌های مهم مانند "خرید بلیت" و پروفایل کاربری نیز به طور ثابت و در جایگاه‌های مشخصی قرار گرفته اند.
- بازخورد (Feedback) بصری و فوری : پس از هر عملیات (مانند انتخاب صندلی یا تکمیل پرداخت)، پیام‌های واضحی برای موفقیت یا خطا به کاربر نمایش داده خواهد شد تا از وضعیت کار خود آگاه شود.
- واکنش‌گرایی (Responsive) : با توجه به طراحی مدرن و مرتب این رابط کاربری، انتظار می‌رود که این سیستم به طور کامل با انواع دستگاه‌ها، از موبایل و تبلت گرفته تا دسکتاپ، سازگار باشد تا تجربه کاربری در هر پلتفرمی یکسان و مطلوب باشد.

۱۳. معماری سیستم

سیستم «رزرو بلیت سینما» بر اساس یک معماری سه‌لایه (Three-Tier Architecture) طراحی و پیاده‌سازی شده است. این معماری شامل سه لایه اصلی زیر است:

۱۰. لایه ارائه (Presentation Layer)

این لایه مسئول تعامل مستقیم با کاربر است و از طریق رابط کاربری وب ارائه می‌شود. وظایف این لایه شامل:

- نمایش اطلاعات فیلم‌ها، سانس‌ها و صندلی‌ها به کاربر
- دریافت ورودی‌ها مانند اطلاعات ثبت‌نام، انتخاب صندلی، و پرداخت
- نمایش پیام‌ها و نتایج عملیات (موفقیت یا خطا)

فناوری‌های استفاده شده در این لایه:

HTML, CSS, JavaScript, Bootstrap برای استایل‌دهی

۲۰. لایه منطق کسب‌وکار (Application Layer)

این لایه مسئول پردازش درخواست‌های کاربر و اجرای منطق اصلی سیستم است. در این پروژه از Django Framework به عنوان بک‌اند استفاده شده که ویژگی‌های زیر را فراهم می‌کند:

- مدیریت مسیرها (Routing) و کنترل درخواست‌ها
- پیاده‌سازی منطق رزرو بلیت، بررسی موجودی صندلی‌ها، مدیریت تراکنش‌ها
- مدیریت احراز هویت و مجوزهای کاربری

۳. لایه داده (Data Layer)

این لایه شامل پایگاه داده رابطه‌ای (Relational Database) است که تمام داده‌های سیستم مانند اطلاعات کاربران، فیلم‌ها، سانس‌ها، تراکنش‌ها و بلیت‌ها در آن ذخیره می‌شود.

- مدل‌سازی داده‌ها: با استفاده از ORM در Django انجام شده است.
- مدیریت ارتباطات: از کلیدهای خارجی (Foreign Keys) برای ارتباط بین جداول استفاده شده است.

۱.۱۳. ارتباط بین لایه‌ها

1. کاربر از طریق مرورگر به رابط کاربری دسترسی دارد.
2. درخواست‌ها (HTTP Requests) از لایه ارائه به لایه منطق کسب‌وکار ارسال می‌شود.
3. لایه منطق کسب‌وکار با پایگاه داده ارتباط گرفته، داده‌ها را پردازش می‌کند و نتیجه را به لایه ارائه بازمی‌گرداند.

۱.۴. جزئیات پیاده‌سازی

در این بخش، فرآیندهای اصلی سیستم توضیح داده می‌شوند. این توضیحات بر دو ماژول اصلی پروژه، "حساب کاربری" و "بلیت‌فروشی"، تمرکز دارد.

۱.۴.۱. ماژول حساب کاربری (accounts)

این ماژول مسئول مدیریت تمام فرآیندهای مرتبط با کاربران از جمله ثبت‌نام، ورود، و مدیریت پروفایل است.

- **ثبت‌نام و ورود:** کاربران جدید با پر کردن فرم ثبت‌نام که در فایل `forms.py` تعریف شده، می‌توانند در سیستم ثبت‌نام کنند. پس از تأیید اطلاعات، یک حساب کاربری جدید ایجاد می‌شود و کاربر می‌تواند با استفاده از نام کاربری و رمز عبور وارد سیستم شود.
- **مدیریت پروفایل:** هر کاربر یک پروفایل اختصاصی دارد که شامل اطلاعاتی مانند نام، جنسیت و آدرس است. کاربران می‌توانند اطلاعات پروفایل خود را ویرایش کنند.
- **مدیریت مالی:** مدل `Payment` در این ماژول، تراکنش‌های مالی کاربران را مدیریت می‌کند. این مدل، امکان افزایش موجودی (deposit) و ثبت پرداخت‌ها را فراهم می‌سازد، که پایه اصلی خرید بلیت است.

۲.۱۴. ماژول بلیت‌فروشی (ticketing)

این ماژول قلب سیستم رزرو بلیت است و تمام فرآیندهای مربوط به نمایش فیلم‌ها و فروش بلیت را در بر می‌گیرد.

- **مدیریت فیلم‌ها و سینماها:** اطلاعات فیلم‌ها (مانند پوستر، خلاصه داستان و کارگردان) و سینماها (مانند نام و آدرس) از طریق مدل‌های Movie و Cinema مدیریت می‌شوند.
- **مدیریت سانس‌ها:** مهم‌ترین بخش این ماژول، مدل ShowTime است که اطلاعات هر سانس نمایش را ذخیره می‌کند. این مدل، با مدیریت وضعیت فروش (Status) و تعداد صندلی‌های آزاد free_seats و salable_seats، تضمین می‌کند که فروش بلیت به شکل صحیح انجام شود.
- **فرآیند خرید بلیت:** فرآیند خرید بلیت در فایل views.py تعریف شده است. کاربر ابتدا فیلم و سانس مورد نظر را انتخاب می‌کند، سپس تعداد صندلی‌های دلخواه خود را رزرو کرده و با انجام پرداخت، بلیت برای او ثبت می‌شود.

۱۴.۳. ویژگی‌های امنیتی

این پروژه، از چندین ویژگی امنیتی داخلی فریم‌ورک استفاده می‌کند تا از کاربران و داده‌ها محافظت کند. این پروژه از SECRET_KEY برای توابع رمزنگاری استفاده می‌کند و با فعال بودن CsrfViewMiddleware و AuthenticationMiddleware، در برابر حملات متداول مانند جعل درخواست بین‌سایتی (CSRF) محافظت می‌شود. برای محدود کردن دسترسی، بسیاری از views با دکوراتور @login_required محافظت شده‌اند که تضمین می‌کند فقط کاربران احراز هویت شده می‌توانند به صفحاتی مانند جزئیات پروفایل یا لیست بلیت‌ها دسترسی داشته باشند. همچنین، پروژه از سیستم مدیریت کاربر داخلی جنگو برای احراز هویت و ذخیره‌سازی امن رمز عبور استفاده می‌کند و با تابع update_session_auth_hash، امنیت نشست کاربری پس از تغییر رمز عبور حفظ می‌شود. اعتبارسنجی‌های رمز عبور نیز بطور موقت در فایل settings.py غیرفعال شده‌اند که در صورت نیاز به سیاست‌های سخت‌گیرانه‌تر برای رمز عبور، باید فعال شوند.

۱۴.۴. ساختار پروژه و مسیریابی (Routing)

این پروژه از الگوی طراحی MVC (Model-View-Controller) جنگو پیروی می‌کند که به طور مجزا کد مربوط به مدل‌ها، نماها و کنترل‌کننده‌ها را سازماندهی می‌کند. ساختار آن شامل یک پروژه اصلی به نام cinema و دو اپلیکیشن مجزا ticketing و accounts است. مسیریابی URL در دو سطح مدیریت می‌شود: فایل cinema/urls.py به عنوان فایل اصلی، درخواست‌ها را به اپلیکیشن‌های مربوطه ارسال می‌کند. سپس هر اپلیکیشن (مانند ticketing/urls.py) مسئول مدیریت مسیرهای داخلی خود است.

۱۴.۵. مدیریت فایل‌های ایستا و رسانه‌ای (Static & Media Files)

برای نمایش رابط کاربری بهینه، پروژه از فایل‌های ایستا مانند CSS، JavaScript و تصاویر استفاده می‌کند که در پوشه‌های static هر اپلیکیشن نگهداری می‌شوند. همچنین، فایل‌های رسانه‌ای که توسط کاربران آپلود می‌شوند (مانند پوستر فیلم‌ها و تصاویر پروفایل)، در مسیر MEDIA_ROOT ذخیره شده و از طریق MEDIA_URL در دسترس قرار می‌گیرند. این فرآیند با استفاده از django.conf.urls.static در cinema/urls.py مدیریت می‌شود.

۱۴.۶. سیستم مدیریت محتوا (Django Admin)

این پروژه از پنل مدیریت داخلی جنگو (Django Admin) برای مدیریت آسان داده‌ها استفاده می‌کند. فایل‌های admin.py در هر اپلیکیشن، مدل‌هایی مانند Movie, Cinema, ShowTime و Profile را به پنل مدیریت اضافه می‌کنند تا مدیران سیستم بتوانند به راحتی فیلم‌های جدید، سینماها، سانس‌ها و پروفایل کاربران را مدیریت کنند. این ویژگی، فرآیند افزودن، ویرایش و حذف اطلاعات را بدون نیاز به کدنویسی اضافی برای مدیران فراهم می‌سازد.

۷.۱۴. سیستم قالب‌ها (Templating System)

رابط کاربری پروژه با استفاده از سیستم قالب‌های (Templates) جنگو پیاده‌سازی شده است. این سیستم به جداسازی منطق از نمایش کمک می‌کند. تمامی فایل‌های HTML در پوشه‌های templates هر اپلیکیشن سازماندهی شده‌اند. از قالب‌های پایه مانند base_page.html برای ایجاد یکپارچگی در ظاهر صفحات و جلوگیری از تکرار کد استفاده شده است. این قالب‌ها به صورت پویا و با استفاده از زبان قالب جنگو (Django Template Language) مقادیر مورد نیاز را از Views دریافت و نمایش می‌دهند.

۱۵. مشکلات و چالش‌ها

در هر پروژه نرم‌افزاری، مواجهه با چالش‌های فنی و طراحی امری اجتناب‌ناپذیر است. در طول توسعه پروژه «سیستم رزرو بلیت سینما»، تیم با مشکلات مختلفی روبرو شد که با تحلیل و پیاده‌سازی راه‌حل‌های مناسب، برطرف گردید.

۱۵.۱. مشکلات فنی

- **همزمانی (Concurrency) در خرید بلیت:** این یکی از مهم‌ترین چالش‌های فنی بود. فرض کنید دو کاربر به طور همزمان برای خرید آخرین صندلی یک سانس اقدام کنند. اگر سیستم به درستی مدیریت نشود، هر دو کاربر موفق به خرید خواهند شد و در نتیجه، فروش بیش از ظرفیت رخ می‌دهد.

- **راه‌حل:** برای حل این مشکل، از مدیریت تراکنش‌ها در پایگاه داده استفاده شد. با اجرای عملیات خرید بلیت در یک تراکنش Atomic، تضمین شد که کل فرآیند (بررسی صندلی‌های آزاد، کسر مبلغ و ثبت بلیت) به صورت یک واحد انجام شود. اگر در هر مرحله خطایی رخ دهد، کل عملیات به حالت اولیه بازمی‌گردد. همچنین، متد reserve_seats() در مدل ShowTime به دقت پیاده‌سازی شد تا از بروز این خطا جلوگیری کند.

- **مدیریت ظرفیت سالن:** در ابتدای پروژه، مدیریت ظرفیت سالن و تعداد صندلی‌های آزاد به صورت دستی انجام می‌شد که احتمال خطا را افزایش می‌داد.

○ **راه‌حل:** برای خودکارسازی این فرآیند، فیلدهای `free_seats` و `salable_seats` به مدل `ShowTime` اضافه شدند. با هر بار خرید بلیت، تعداد صندلی‌های آزاد به صورت خودکار کاهش می‌یابد و سیستم به طور پویا وضعیت سالن را نمایش می‌دهد. متد `is_full()` نیز برای بررسی پر بودن سالن مورد استفاده قرار گرفت.

۱۵.۲. چالش‌های طراحی و راه‌حل‌ها

- **پیچیدگی واسط کاربری (UI/UX):** در طراحی اولیه، ممکن بود فرآیند خرید بلیت برای کاربرانی که آشنایی کمتری با سیستم‌های آنلاین دارند، گیج‌کننده باشد.

○ **راه‌حل:** با تمرکز بر اصول **سادگی و وضوح**، واسط کاربری مورد بازنگری قرار گرفت. از فریم‌ورک `Bootstrap` و کلاس‌های `CSS` برای ایجاد یک طراحی ریسپانسیو و کاربرپسند استفاده شد. دکمه‌های اصلی مانند «خرید بلیت» به صورت برجسته و مسیرهای ناوبری به صورت ساده و قابل دسترس طراحی شدند.

- **یکپارچه‌سازی با سیستم پرداخت:** اتصال به یک درگاه پرداخت آنلاین (چشم انداز آینده و نه پروژه فعلی) چالش‌هایی فنی در پی خواهد داشت، از جمله مدیریت تراکنش‌های موفق و ناموفق.

○ **راه‌حل:** برای این منظور، مدل `Payment` در اپلیکیشن `accounts` ایجاد شد تا تمام تراکنش‌های مالی به دقت ثبت شود. سیستم به گونه‌ای طراحی شد که در صورت ناموفق بودن تراکنش، هیچ بلیتی صادر نشود و هیچ مبلغی از حساب کاربر کسر نگردد.

- **بهبود مستمر و نگهداری آسان:** با توجه به ماهیت پروژه دانشگاهی، نیاز بود تا سیستم به گونه‌ای طراحی شود که در آینده قابل توسعه و نگهداری باشد.

○ **راه‌حل:** از معماری **سه‌لایه** و اصل **جداسازی نگرانی‌ها** با تقسیم پروژه به دو اپلیکیشن `accounts` و `ticketing` استفاده شد. این ساختار ماژولار باعث می‌شود که تغییر در یک بخش، تأثیری بر بخش‌های دیگر نداشته باشد و نگهداری آن آسان‌تر باشد.

۱۶. ساخت و ساز (آزمایش)

همانطور که اشاره کردید، آزمایش نرم افزار به صورت دستی و در سه سطح اصلی برای اطمینان از عملکرد صحیح پروژه انجام شده است. در ادامه، سناریوهای آزمایشی که شما مشخص کرده اید به همراه نتایج و مشکلات شناسایی شده آورده شده است.

انواع تست انجام شده

- **تست واحد (Unit Test):** در این سطح، کوچک ترین واحدهای منطقی کد (مانند توابع و متدها) به صورت جداگانه آزمایش شدند. برای مثال، متد `reserve_seats()` در مدل `ShowTime` به صورت مجزا تست شد تا از کاهش صحیح تعداد صندلی های آزاد اطمینان حاصل شود.
- **تست یکپارچه سازی (Integration Test):** در این تست، ماژول های مختلف سیستم (مانند ماژول حساب کاربری و بلیت فروشی) در کنار هم آزمایش شدند تا تعامل صحیح آن ها بررسی شود. برای مثال، فرآیند کامل خرید بلیت که شامل کسر مبلغ از موجودی کاربر و ثبت بلیت است، مورد آزمایش قرار گرفت.
- **تست سیستمی (System Test):** در این مرحله، کل سیستم به عنوان یک واحد کامل و تحت سناریوهای واقعی مورد ارزیابی قرار گرفت تا اطمینان حاصل شود که تمام قابلیت ها به درستی و مطابق با نیازمندی ها کار می کنند.

۱.۱۶. سناریوهای آزمایشی و نتایج

در طول فرآیند تست دستی، چندین سناریو برای شبیه سازی رفتار کاربران و شناسایی خطاها اجرا شد:

۱. سناریوی خرید بلیت با صندلی پر

- **هدف:** بررسی رفتار سیستم هنگام تلاش کاربر برای خرید بلیت برای یک سانس تکمیل ظرفیت.

○ گام های آزمایشی:

۱. یک سانس با ظرفیت محدود (مثلاً یک صندلی) ایجاد شد.

۲. کاربر اول بلیت آن یک صندلی را خریداری کرد.

۳. کاربر دوم تلاش کرد تا برای همان سانس بلیت خریداری کند.

- **نتیجه و مشکل شناسایی شده:** سیستم باید به کاربر دوم پیامی مبنی بر پر بودن سالن نمایش دهد. در نسخه های اولیه پروژه، به دلیل عدم مدیریت صحیح همزمانی (**Concurrency**)، ممکن بود سیستم به هر دو کاربر اجازه خرید دهد که منجر به فروش بیش از ظرفیت می شد. این مشکل با پیاده سازی متد `is_full()` در مدل `ShowTime` و مدیریت تراکنش ها برطرف شد.

2. سناریوی ورود با رمز عبور اشتباه

- هدف: بررسی فرآیند ورود به سیستم و نمایش پیام‌های خطا.
- گام‌های آزمایشی:
 1. کاربر با نام کاربری صحیح و رمز عبور اشتباه تلاش کرد تا وارد سیستم شود.
 2. کاربر با هر دو فیلد خالی فرم ورود را ارسال کرد.
- نتیجه و مشکل شناسایی شده: سیستم به درستی پیام خطای مناسب را نمایش داد. در نسخه‌های اولیه، پیام‌های خطا ممکن بود به درستی به کاربر منتقل نشوند، اما این مسئله با استفاده از فرم‌های جنگو و اعتبارسنجی سمت سرور اصلاح شد.

۱۷. کارهای آینده

پروژه «سیستم رزرو بلیت سینما» با موفقیت پیاده‌سازی شده و عملکرد اصلی خود را به خوبی انجام می‌دهد. با این حال، همیشه امکان گسترش و بهبود سیستم وجود دارد تا تجربه کاربری افزایش یافته و قابلیت‌های مدیریتی بیشتری فراهم شود. در ادامه، برخی از کارهای آینده برای توسعه پروژه ارائه شده است:

۱. اپلیکیشن موبایل

با توجه به افزایش استفاده از تلفن‌های همراه، توسعه یک اپلیکیشن موبایل برای سیستم رزرو بلیت می‌تواند گام بزرگی در جهت افزایش دسترسی کاربران باشد. این اپلیکیشن با استفاده از فریم‌ورک‌های توسعه موبایل (مانند React Native یا Flutter)، قابلیت‌های زیر را ارائه خواهد داد:

- رزرو و خرید بلیت: دسترسی سریع و آسان به فرآیندهای اصلی سیستم.
- نوتیفیکیشن: ارسال نوتیفیکیشن برای اطلاع‌رسانی در مورد سانس‌های جدید، تخفیف‌ها یا تأیید خرید بلیت.
- کیف پول دیجیتال: امکان شارژ حساب و پرداخت سریع از طریق اپلیکیشن.

۲. سیستم پیشنهاد فیلم (Recommendation System)

برای بهبود تجربه کاربری و افزایش فروش، می‌توان از یک سیستم پیشنهاد فیلم مبتنی بر هوش مصنوعی استفاده کرد. این سیستم با تحلیل داده‌های زیر، به کاربران فیلم‌هایی را پیشنهاد می‌دهد که احتمالاً مورد علاقه آن‌ها خواهد بود:

- تاریخچه خرید بلیت: فیلم‌هایی که کاربر قبلاً تماشا کرده است.
- امتیازدهی و نظرات: فیلم‌هایی که کاربران مشابه، به آن‌ها امتیاز بالا داده‌اند.
- الگوهای تماشا: ژانر، کارگردان، یا بازیگرانی که کاربر بیشتر به آن‌ها علاقه‌مند است.

۳. تخفیف‌ها و کدهای پروموشن (Promotional Codes)

ایجاد یک سیستم مدیریت تخفیف و کدهای پروموشن می‌تواند به جذب مشتریان جدید و افزایش وفاداری مشتریان فعلی کمک کند. این سیستم امکانات زیر را فراهم می‌کند:

- ایجاد کدهای تخفیف: مدیران می‌توانند کدهای تخفیف یک‌بار مصرف یا چندبار مصرف ایجاد کنند.
- تخفیف‌های ویژه: امکان ارائه تخفیف بر اساس معیارهای مختلف (مانند خرید بلیت در روزهای خاص یا برای تعداد مشخصی از صندلی‌ها).
- سیستم وفاداری: ارائه امتیاز به کاربران برای هر خرید و امکان استفاده از آن امتیازها برای خریدهای بعدی.

۴. تحلیل داده‌ها و گزارش‌های پیشرفته

با توجه به داده‌های زیادی که در پایگاه داده سیستم ذخیره می‌شود، می‌توان از ابزارهای تحلیلی برای ارائه گزارش‌های پیشرفته به مدیران سینما استفاده کرد. این گزارش‌ها می‌تواند شامل موارد زیر باشد:

- تحلیل فروش: گزارش‌های دقیق از میزان فروش فیلم‌ها و سانس‌ها.
 - تحلیل رفتار مخاطب: درک سلیقه مخاطبان بر اساس ژانر و نوع فیلم‌های مورد علاقه.
 - پیش‌بینی فروش: پیش‌بینی میزان تقاضا برای سانس‌های آینده بر اساس داده‌های تاریخی.
- این گزارش‌ها به مدیران کمک می‌کند تا تصمیمات بهتری برای برنامه‌ریزی اکران فیلم‌ها و بهینه‌سازی فروش بگیرند.

۱۸. نتیجه‌گیری و دستاوردها

پروژه «سیستم رزرو بلیت سینما» با موفقیت طراحی و پیاده‌سازی شد و توانست به اهداف اولیه خود که در مستندات پروژه مشخص شده بود، دست یابد. این سیستم نه تنها مشکلات فرآیندهای سنتی را برطرف کرد، بلکه دستاوردهای قابل توجهی در زمینه تجربه کاربری و مدیریت به همراه داشت.

۱۹. دستاوردها نسبت به اهداف اولیه

- **سهولت دسترسی کاربران:** سیستم به کاربران امکان می‌دهد تا به راحتی و از هر مکانی به لیست فیلم‌ها، سانس‌ها و ظرفیت سالن‌ها دسترسی پیدا کنند.
- **مدیریت هوشمند ظرفیت سالن‌ها:** با استفاده از مدل‌های داده‌ای مانند ShowTime و فیلد salable_seats، سیستم به مدیران سینما کمک کرد تا به صورت خودکار از فروش بیش از ظرفیت جلوگیری کرده و مدیریت بهینه‌ای بر روی صندلی‌های آزاد داشته باشند.
- **کاهش هزینه‌ها و اتلاف زمان:** با حذف نیاز به صف‌های طولانی و خرید حضوری بلیت، فرآیند خرید برای کاربران ساده و سریع شد.

۲۰. تأثیر سیستم بر تجربه کاربری و مدیریت سینما

- **بهبود تجربه کاربری:** این سیستم، تجربه خرید بلیت را از یک فرآیند زمان‌بر و خسته‌کننده به یک فرآیند سریع، لذت‌بخش و کاربرپسند تبدیل کرد. کاربران می‌توانند به راحتی از طریق یک واسط کاربری جذاب، فیلم‌ها را مرور کرده، صندلی دلخواه خود را انتخاب و بلیت را خریداری کنند.
- **افزایش بهره‌وری مدیران:** سیستم اطلاعاتی یکپارچه، ابزارهای لازم برای مدیریت برنامه‌های اکران، ردیابی فروش و مشاهده وضعیت سالن‌ها را به صورت لحظه‌ای در اختیار مدیران قرار داد. پنل مدیریت Django در این زمینه نقش مهمی ایفا کرد و امکان مدیریت آسان داده‌ها را فراهم ساخت.
- **تصمیم‌گیری مبتنی بر داده:** با ثبت تمام تراکنش‌ها و فعالیت‌ها، این سیستم به مدیران سینما این امکان را می‌دهد که با استفاده از داده‌های دقیق، تصمیم‌گیری‌های هوشمندانه‌تری در مورد برنامه‌ریزی فیلم‌ها و استراتژی‌های بازاریابی انجام دهند.

این پروژه نمونه‌ای از کاربرد اصول مهندسی نرم‌افزار برای حل یک مشکل واقعی در صنعت سینما و ارائه یک راه‌حل کارآمد و موثر است.