گزارش پروژه

نقشه عملیاتی همکارانه(Ops Map)

درس: برنامه‌سازی پیشرفته - زبان جاوا

استاد: **دکتر طارمی**

بخش اول: نحوه همکاری اعضای گروه

عضو ۱ - [هلیا باقری] - مسئولیت اصلی: طراحی و پیاده‌سازی سرور

اینجانب وظیفه‌ی پیاده‌سازی سمت سرور پروژه را بر عهده داشتم. کلاس‌های MainServer ، ClientHandler، AuthManager، UserStore، MapStateManager**و** ClientManager **توسط من نوشته شد. منطق احراز هویت کاربران، ذخیره‌سازی موقت اطلاعات در حافظه، مدیریت کلاینت‌های متصل و پخش پیام‌های همگانی** (Broadcast) **از جمله بخش‌هایی بود که پیاده‌سازی کردم. همچنین مکانیزم ذخیره و بارگذاری وضعیت نقشه با استفاده از سریال‌سازی در کلاس** MapStorage **انجام شد. در کنار این موارد، در طراحی معماری کلی سیستم و اشکال‌زدایی بخش‌های مشترک با سایر اعضا همکاری داشتم و در نهایت در تهیه گزارش نهایی در رابطه با پروژه با هر سه نفر مشترکا همکاری کردم.**

عضو ۲ - [ابوالفضل شیرافکن] - مسئولیت اصلی: طراحی و پیاده‌سازی کلاینت با JavaFX

مسئولیت من پیاده‌سازی رابط کاربری گرافیکی و منطق سمت کلاینت بود. صفحه‌ی ورود و ثبت‌نام Login.fxml**و** LoginController **و صفحه‌ی اصلی نقشه** Main.fxml **و** MainController **توسط من طراحی و کدنویسی شد. ابزارهای رسم مسیر، نشانگر و منطقه، تنظیم رنگ و ضخامت، انتخاب نوع نشانگر و شکل منطقه، و نیز قابلیت بارگذاری تصویر پس‌زمینه در کلاینت پیاده‌سازی گردید. همچنین ارتباط شبکه‌ای کلاینت با سرور از طریق کلاس** ClientConnection **و مدیریت پیام‌های**

دریافتی با پیاده‌سازی اینترفیس MessageListener انجام پذیرفت. علاوه بر این، قابلیت چت متنی همگانی به عنوان یک ویژگی خلاقانه به برنامه اضافه شد.همچنین در تهیه گزارش نهایی نیز شرکت داشتم.

عضو ۳ - [محمدحسین صمدی] - مسئولیت اصلی: طراحی مدل‌های اشتراکی و مدیریت خطا

نقش من در پروژه طراحی و پیاده‌سازی کلاس‌های اشتراکی (Shared) بین سرور و کلاینت بود. کلاس‌های User, Route, Marker, RegionShape و زیرکلاس‌های آن، MapState, MousePosition, ChatMessage و همچنین ساختار پیام‌ها (Message, MessageType, ErrorPayload) توسط من ایجاد شد. همچنین مدیریت استثناها و ارسال خطاهای معنادار از سرور به کلاینت را با هماهنگی اعضای گروه طراحی کردم. در رفع اشکالات مربوط به سریال‌سازی و اطمینان از یکپارچگی داده‌ها هنگام انتقال از طریق شبکه نیز همکاری داشتم. مستندسازی پروژه و آماده‌سازی گزارش نهایی نیز توسط هرسه نفرمان به صورت مشترک انجام گرفت.

بخش دوم: چالش‌های پیاده‌سازی برنامه کاربردی

چالش اول: همگام‌سازی بلادرنگ وضعیت نقشه بین کلاینت‌ها

یکی از مهم‌ترین چالش‌ها، حفظ یکپارچگی و همگام‌سازی لحظه‌ای وضعیت نقشه میان تمام کاربران آنلاین بود. هر کاربر می‌تواند هم‌زمان با دیگران اقدام به رسم مسیر، افزودن نشانگر یا منطقه کند و باید تغییرات به سرعت در نقشه‌ی سایرین منعکس شود. راه‌حل اولیه، ارسال مستقیم هر رویداد به سرور و پخش آن برای سایر کلاینت‌ها بود. اما این روش باعث ایجاد تداخل و ناهماهنگی در مواردی مانند حذف اشیاء و بارگذاری مجدد نقشه می‌شد. برای رفع این مشکل، سرور یک نمونه‌ی مرکزی از وضعیت نقشه را در MapStateManager نگهداری می‌کند و هر تغییر ابتدا در این نمونه اعمال، سپس پیام مربوطه پخش می‌شود. همچنین پس از بارگذاری یک نقشه‌ی ذخیره‌شده، سرور وضعیت جدید را برای همه‌ی کاربران ارسال می‌کند تا همگی به‌طور کامل همگام شوند.

چالش دوم: پیاده‌سازی امن و صحیح حذف اشیاء با در نظر گرفتن مالکیت و نقش کاربران

بر اساس نیازمندی پروژه، هر کاربر فقط می‌تواند اشیایی را که خود ایجاد کرده است حذف کند و کاربران با نقش فرمانده (Commander) **مجاز به حذف هر شیء هستند. چالش اصلی در این بخش، اعمال این محدودیت‌ها در سمت سرور بود، زیرا در پیاده‌سازی اولیه تمام بررسی‌های مجوز فقط در کلاینت انجام می‌شد و به راحتی قابل دور زدن بود. برای حل این مسئله، ابتدا کلاس** Route **به یک شناسه‌ی یکتا مجهز شد تا قابلیت حذف مسیرها نیز فراهم گردد. سپس در سرور، متد** handleMessage **برای پیام** REMOVE_OBJECT **توسعه یافت: ابتدا مالک شیء از طریق شناسه از وضعیت جاری نقشه استخراج می‌شود، سپس با نام کاربری فرستنده و نقش او مقایسه می‌گردد. تنها در صورت احراز مجوز، عملیات حذف انجام و پیام به سایر کلاینت‌ها پخش می‌شود. این اصلاحیه امنیت و پایبندی به نیازمندی را تضمین کرد.**

چالش سوم: مدیریت هم‌روندی (Concurrency) **در سرور**

سرور پروژه به صورت چندنخی طراحی شده و هر کلاینت در یک نخ مجزا توسط ClientHandler **مدیریت می‌شود. مشکل زمانی بروز کرد که چندین نخ به طور هم‌زمان سعی در تغییر وضعیت نقشه (افزودن/حذف اشیاء) یا لیست کاربران آنلاین داشتند. استفاده از** HashMap **و** ArrayList **بدون همگام‌سازی منجر به بروز** ConcurrentModificationException **و ناهماهنگی داده‌ها می‌شد. برای رفع این چالش، ساختار** UserStore **به** ConcurrentHashMap **تغییر یافت. همچنین متد** removeObject **در** MapStateManager **با کلیدواژه‌ی** synchronized **همگام‌سازی شد تا در یک لحظه فقط یک نخ بتواند وضعیت نقشه را تغییر دهد. در** ClientManager **نیز از** Collections.synchronizedSet **استفاده شده و هنگام پیمایش مجموعه از بلوک** synchronized **بهره گرفته شد. این اصلاحات پایداری سرور را در شرایط بار هم‌زمان افزایش داد.**

بخش سوم: عملکرد خلاقانه - توجیه ضرورت و فایده

۱. چت متنی همگانی (Public Chat)

در بسیاری از سامانه‌های عملیاتی و نقشه‌های تعاملی، ابزاری برای ارتباط سریع بین اپراتورها ضروری است. افزودن قابلیت چت متنی در کنار نقشه، امکان هماهنگی بهتر و انتقال پیام‌های فوری را فراهم می‌کند. این ویژگی با الهام از نیاز واقعی در اتاق‌های فرمان پیاده‌سازی شد. کاربران می‌توانند در حین کار روی نقشه، پیام‌های خود را برای همه‌ی افراد آنلاین ارسال کنند. پیاده‌سازی آن از طریق تعریف کلاس ChatMessage و استفاده از نوع پیام CHAT در پروتکل شبکه انجام شده است. کلاینت با دریافت این پیام، متن را در یک ListView نمایش می‌دهد. اعتبارسنجی و پالایش ورودی (Sanitization) نیز برای جلوگیری از حملات ساده در نظر گرفته شده است.

۲. بارگذاری تصویر دلخواه به عنوان پس‌زمینه نقشه

یکی از محدودیت‌های نسخه‌ی اولیه برنامه، استفاده از یک پس‌زمینه‌ی ساده و ثابت بود که قابلیت شخصی‌سازی را از کاربر می‌گرفت ،اکنون کاربر می‌تواند هر تصویری را از حافظه‌ی محلی انتخاب کرده و به عنوان نقشه‌ی زمینه قرار دهد. این قابلیت با استفاده از FileChooser و BackgroundImage در JavaFX پیاده‌سازی شده است. مزیت این ویژگی درتطبیق برنامه با نیازهای متنوع کاربران است.

هر دو قابلیت مذکور، بدون ایجاد پیچیدگی اضافی در معماری اصلی سیستم، ارزش کاربردی بالایی به پروژه افزوده‌اند .

تهیه‌کنندگان: هلیا باقری،محمدحسین صمدی،ابوالفضل شیرافکن