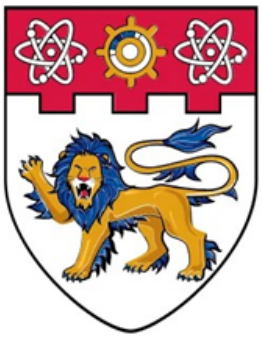


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

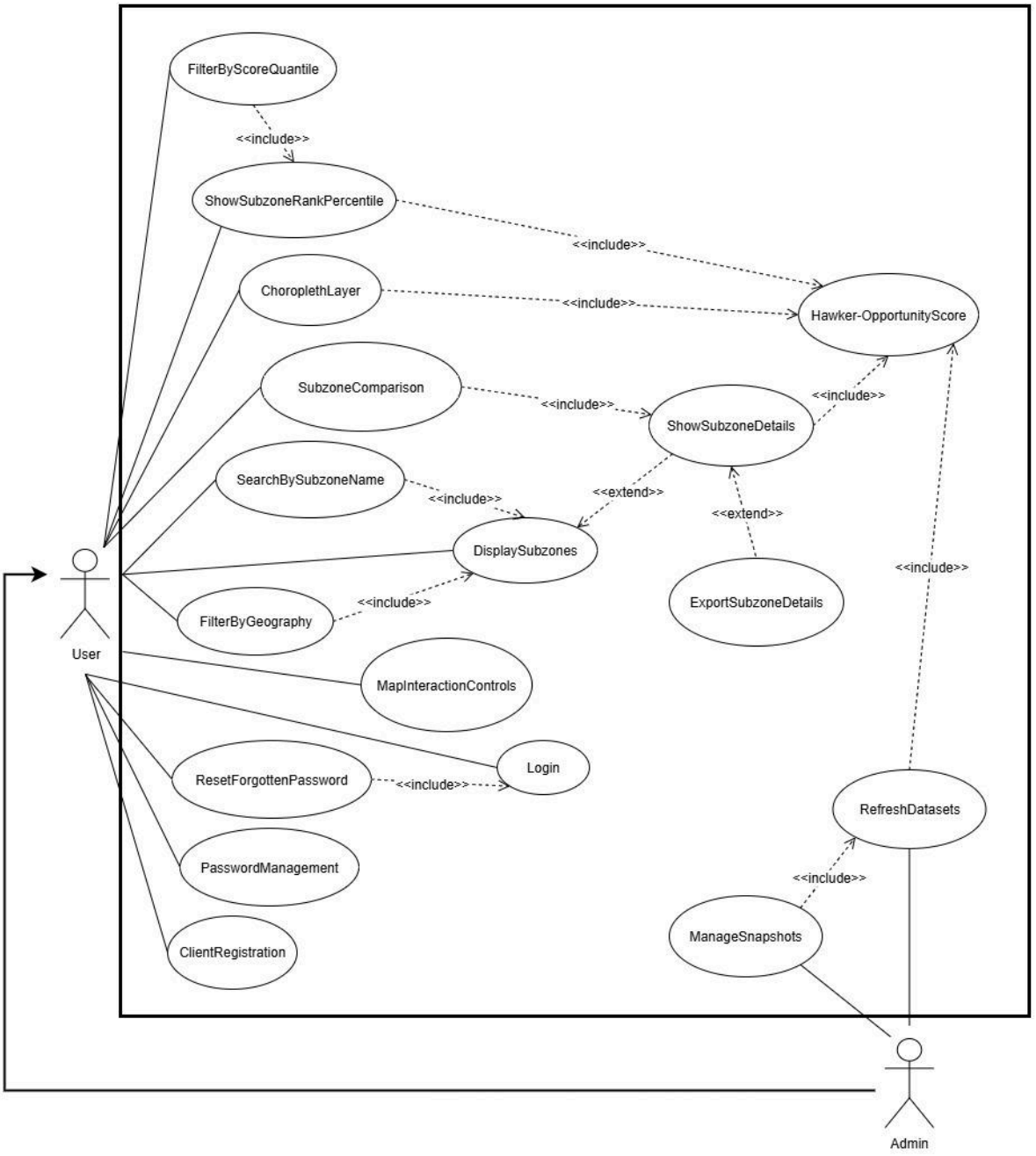
SC2006 – Software Engineering

Lab 3 Deliverables

Lab Group	SCS6
Team	5
Members	Nguyen Le Tam (U2420673E)
	Mehul Modi (U2423434H)
	Nguyen Tran Chien (U2420243A)
	Anthea Toh (U2423691F)
	Chan Tzen Loong Max (U2422224B)

Table of Contents

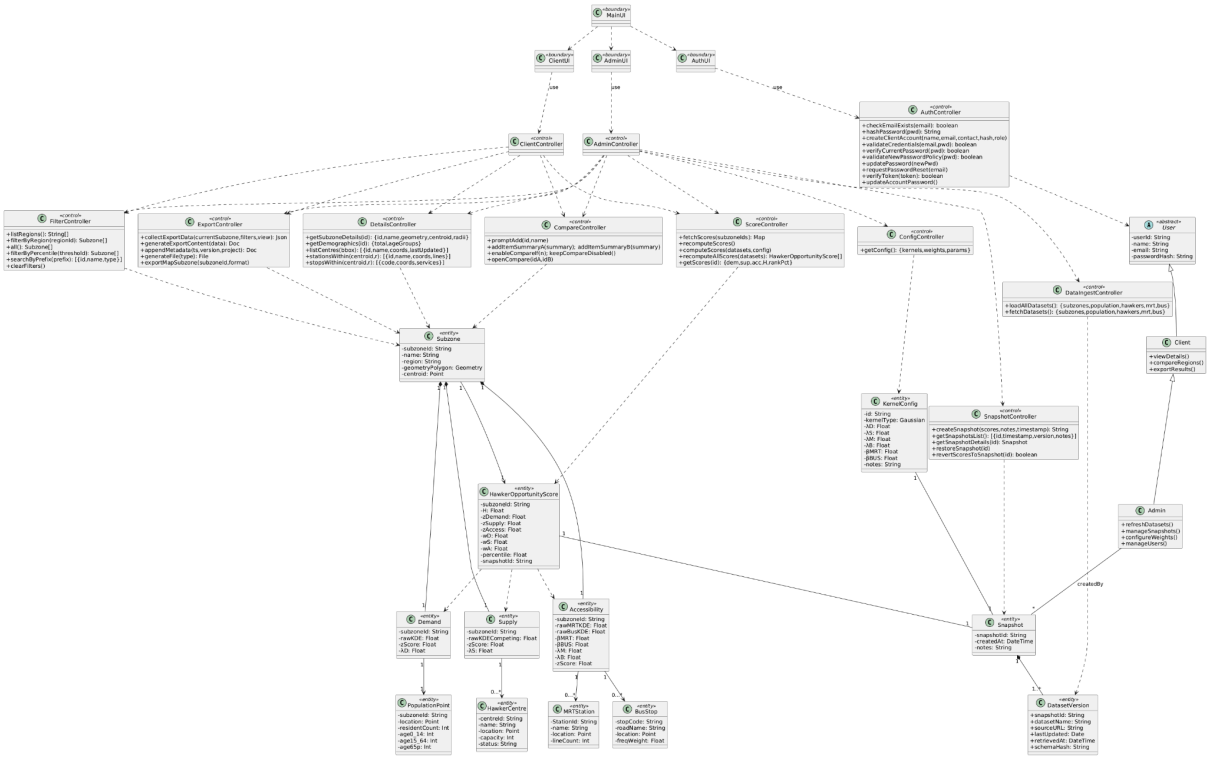
1. Completed Use Case Model



2. Design Model

A. Class Diagram

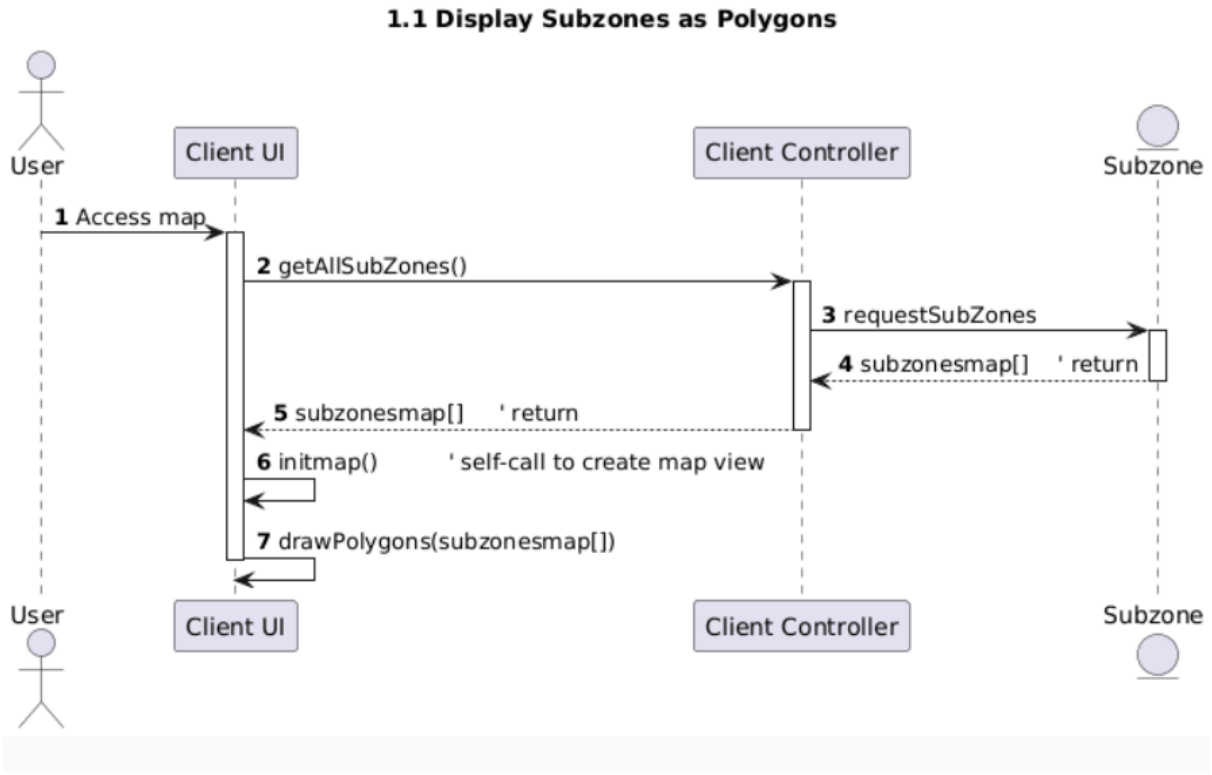
If the image is unclear, please refer to the img file that is uploaded together with this document.



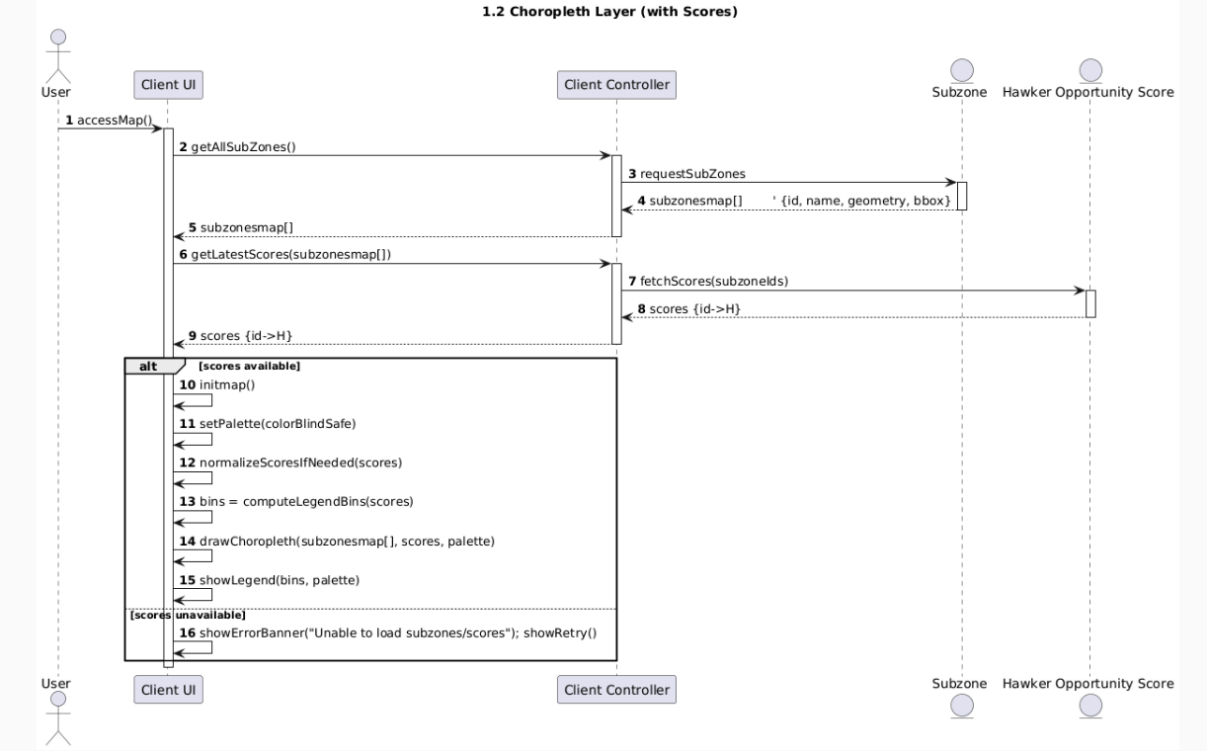
B. Sequence Diagrams

I. For Use Cases Under 1. (Display map)

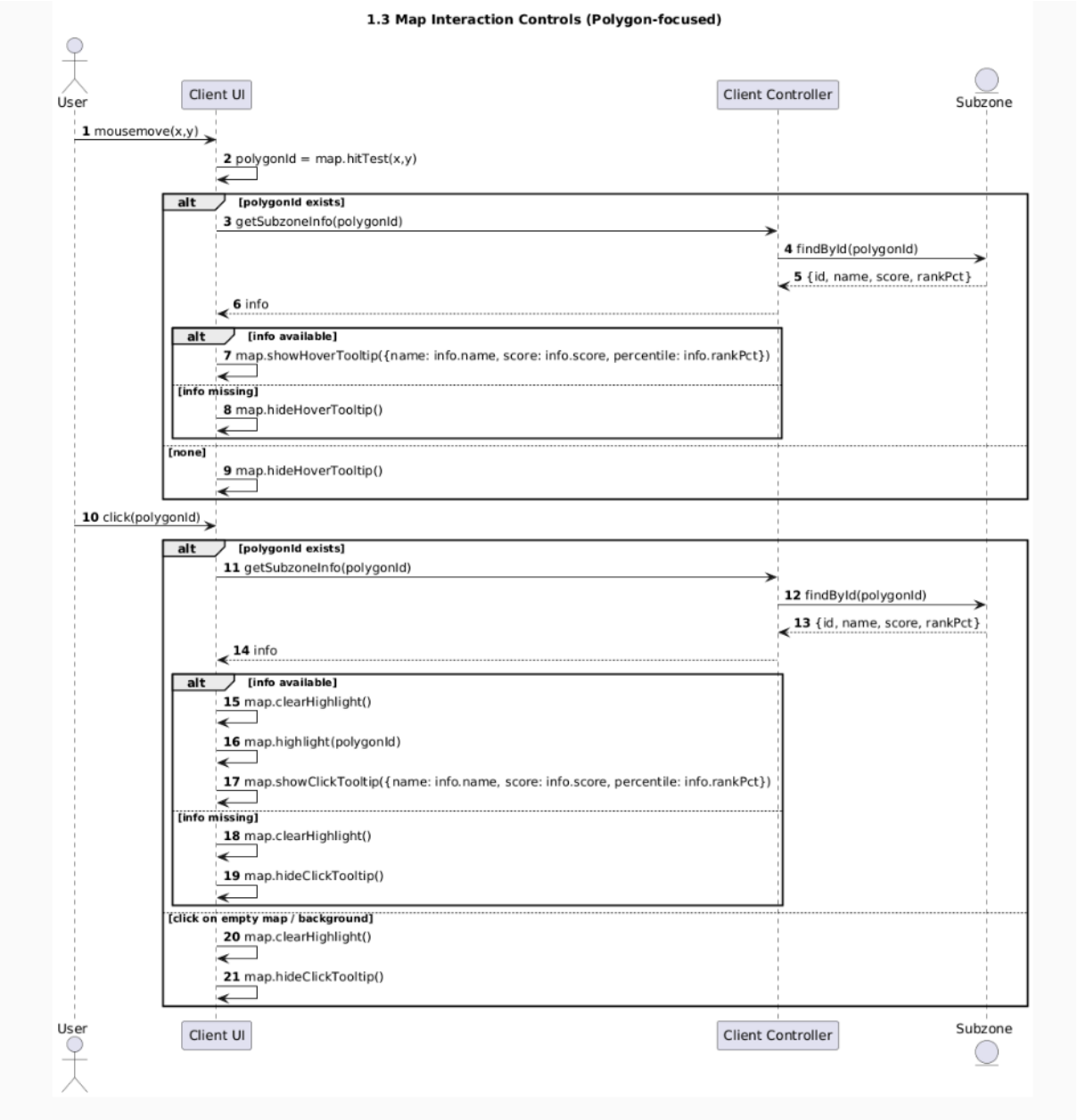
1.1 DisplaySubzones



1.2 Choropleth layer

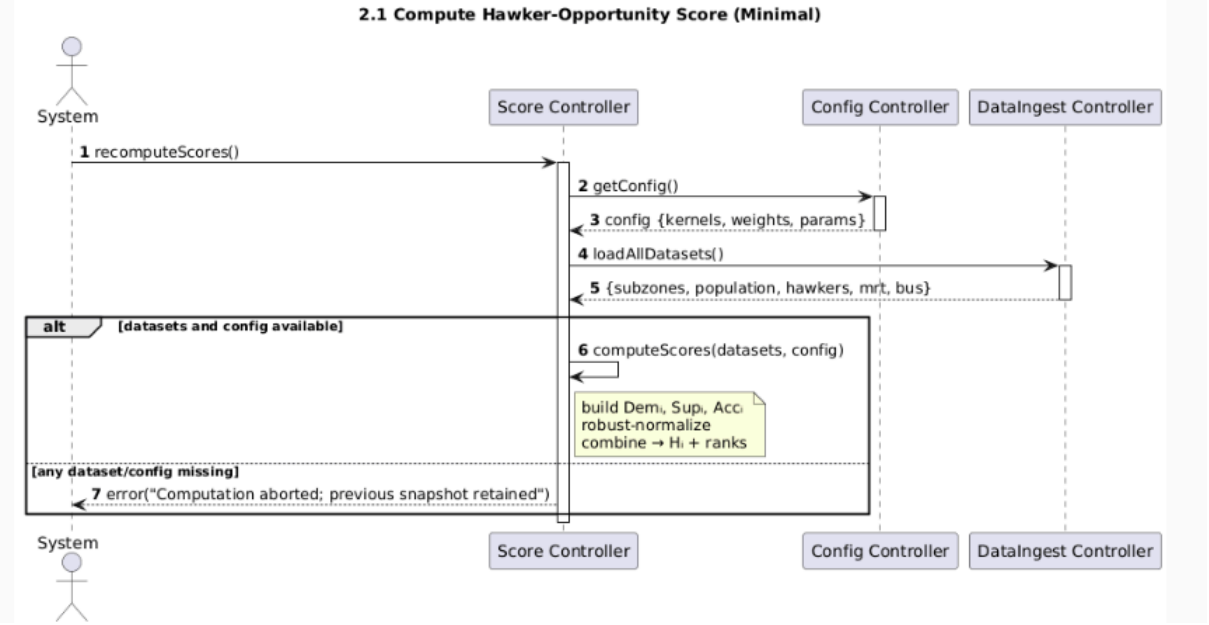


1.3 MapInteractionControls

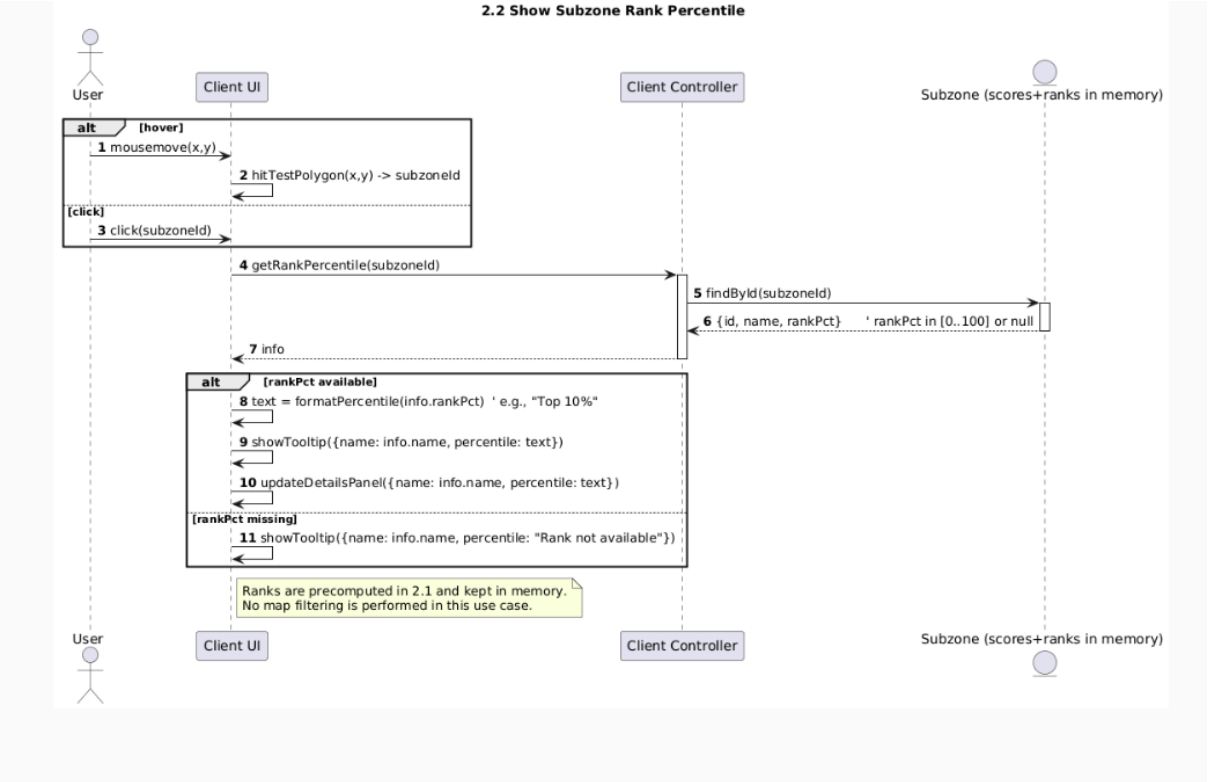


II. For Use Cases Under 2. (Display score and percentile)

2.1 Hawker-Opportunity Score

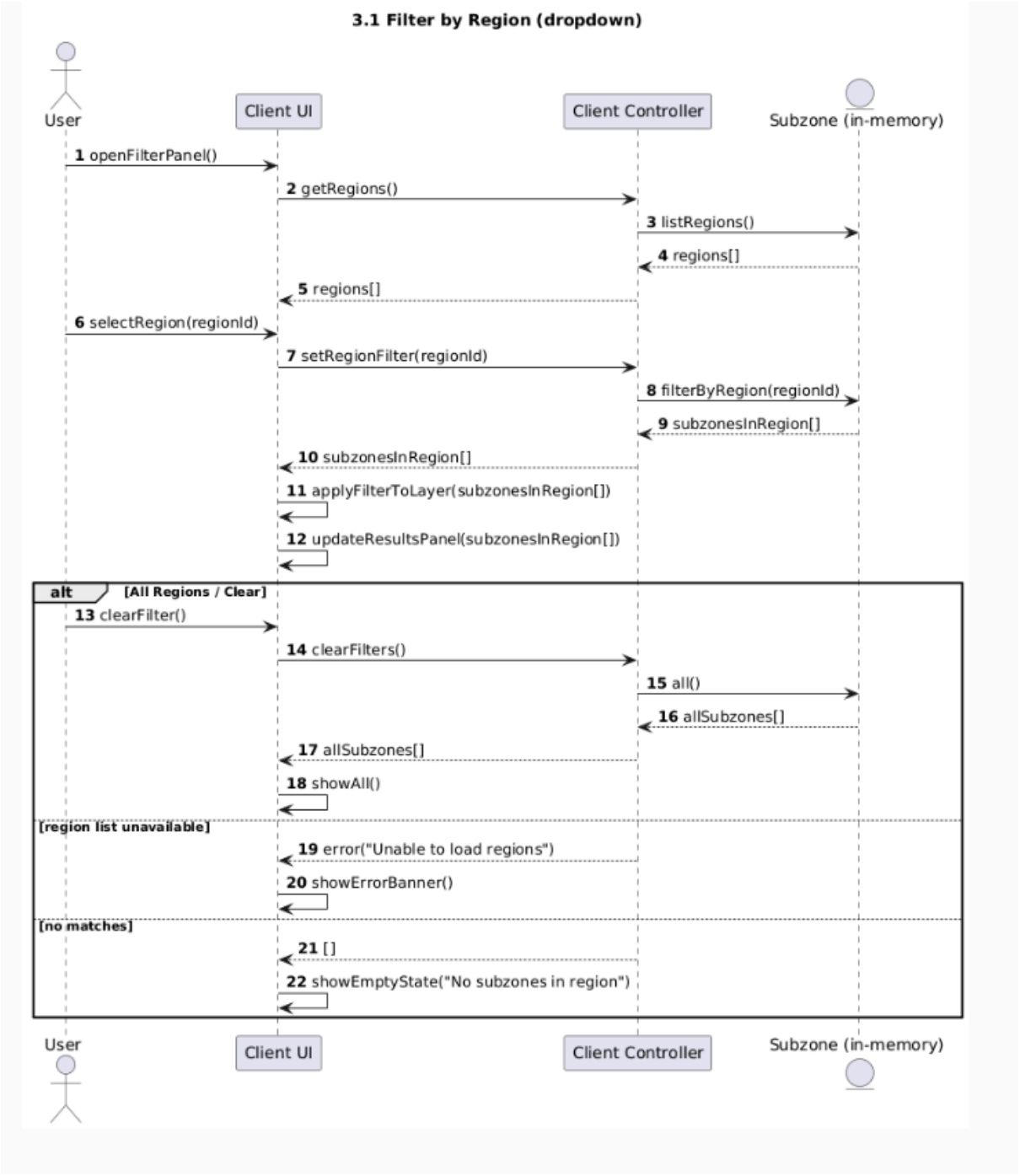


2.2 ShowSubzoneRankPercentile

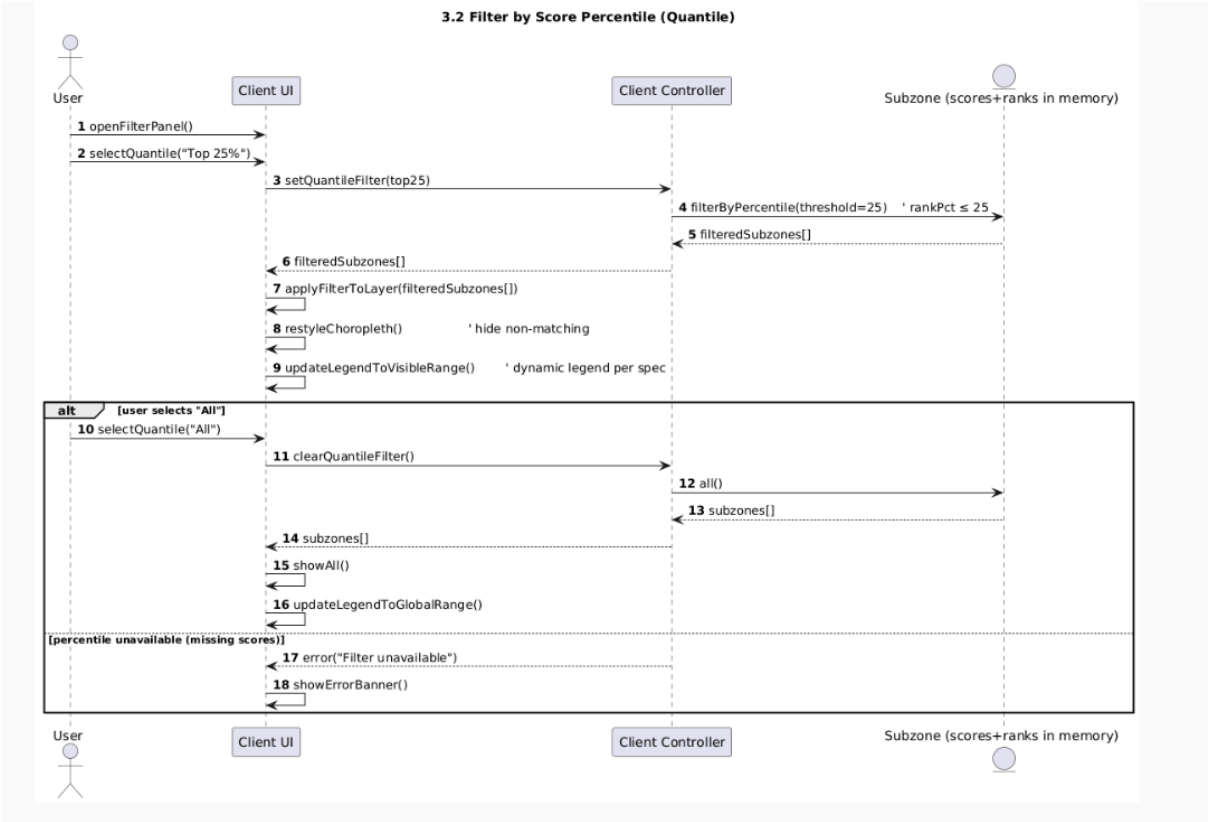


III. For Use Cases Under 3. (Filtering and search)

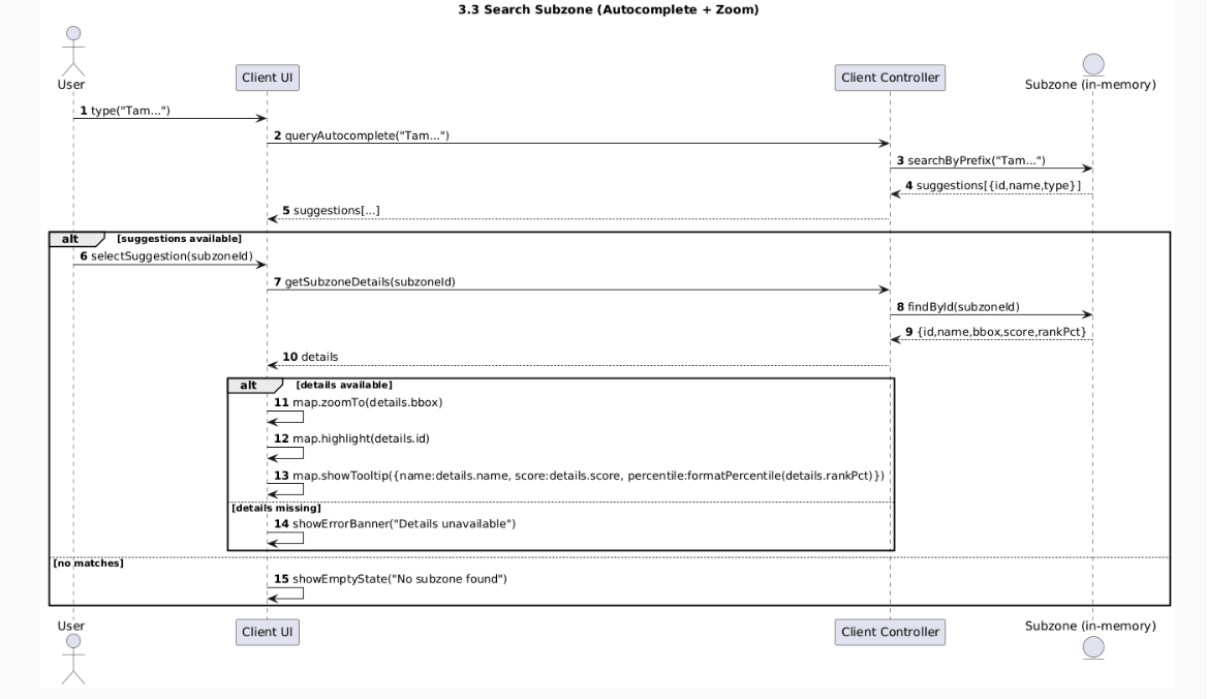
3.1 FilterByGeography



3.2 FilterByScoreQuantile

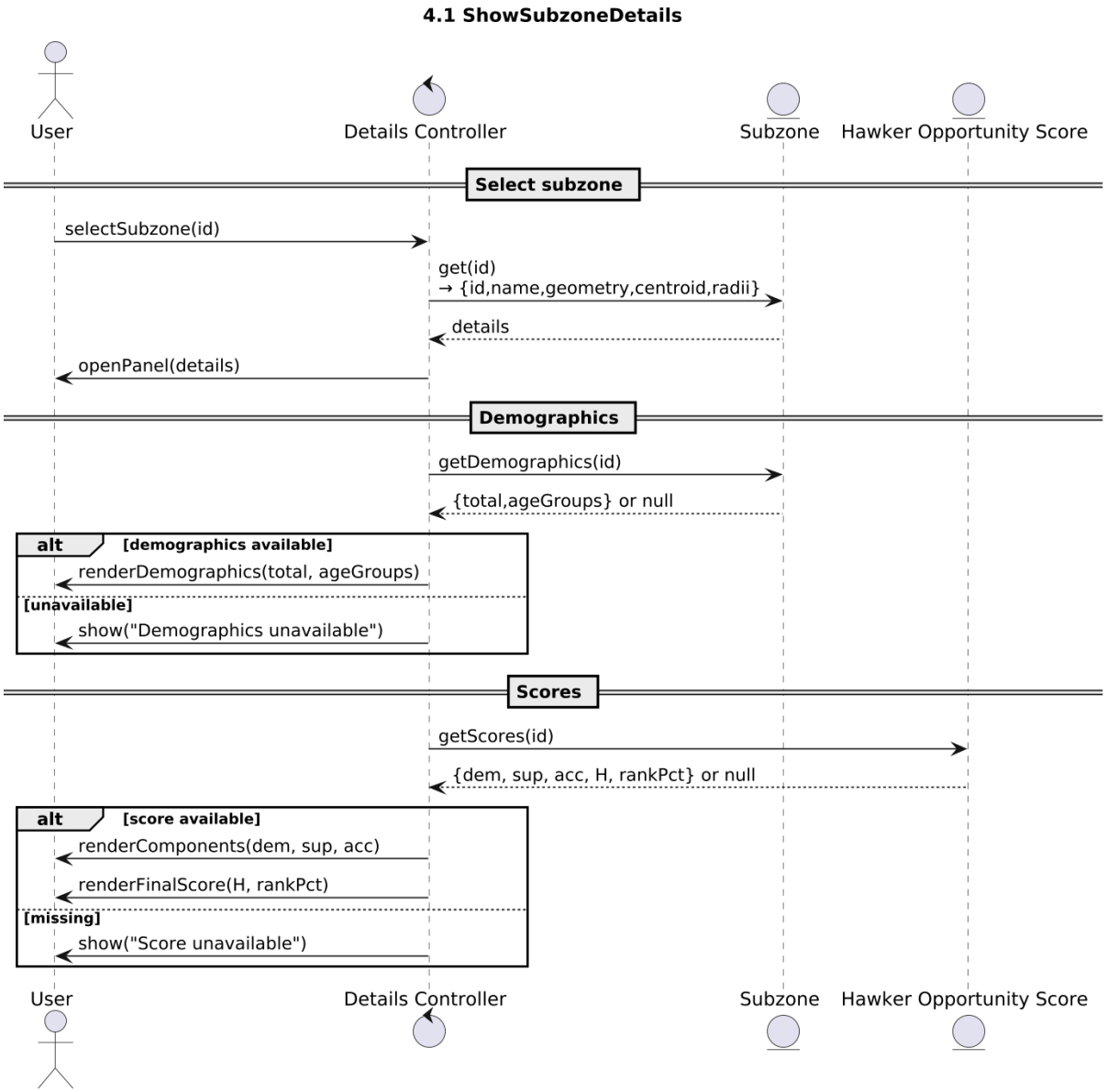


3.3 SearchBySubzoneName

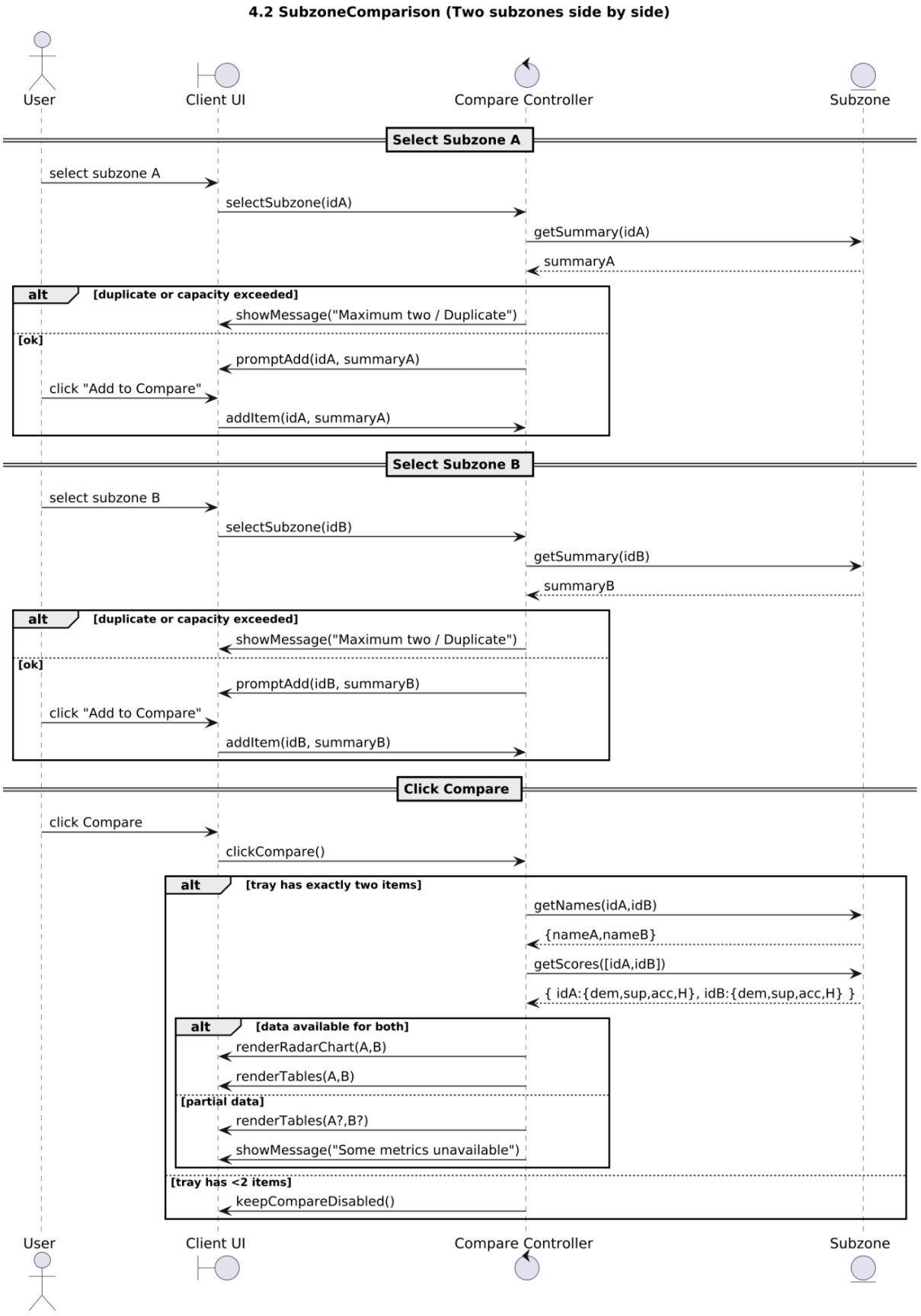


IV. For Functional Requirement #4.

4.1 ShowSubzoneDetails

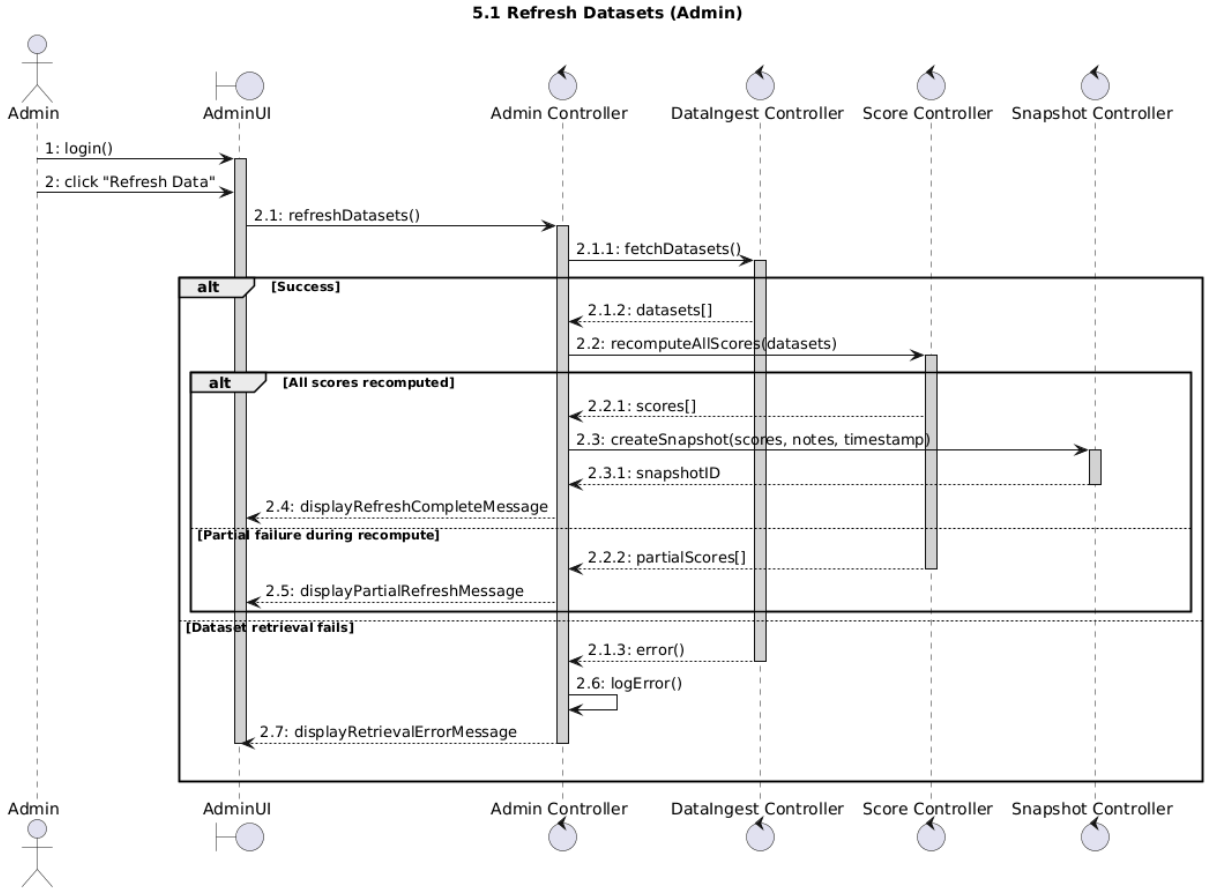


4.2 SubzoneComparison (Two subzones side-by-side)

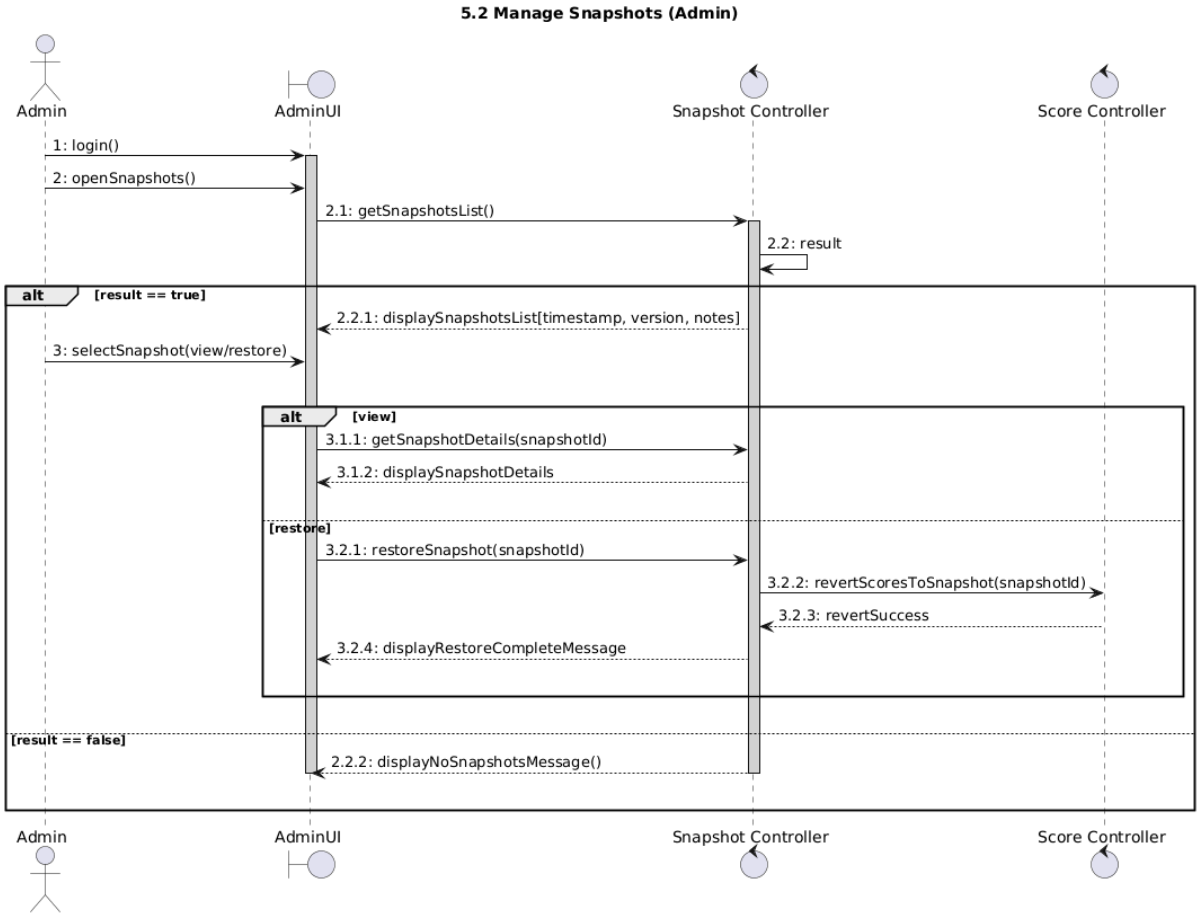


V. For Functional Requirement #5

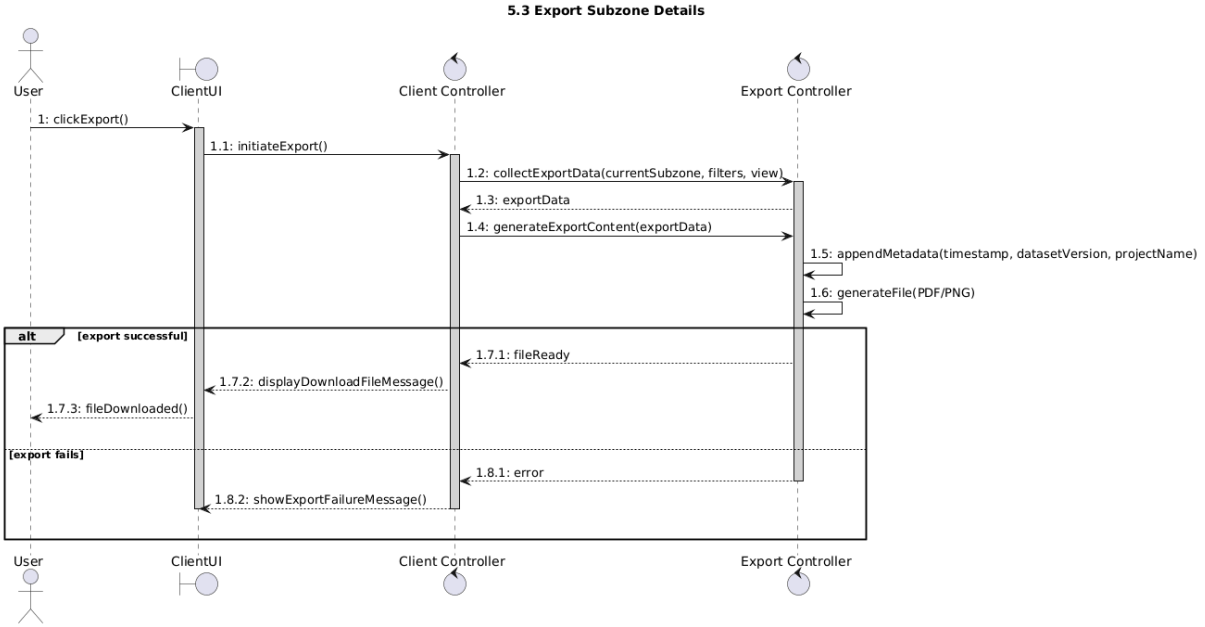
5.1 RefreshDatasets (Admin)



5.2 ManageSnapshots (Admin)

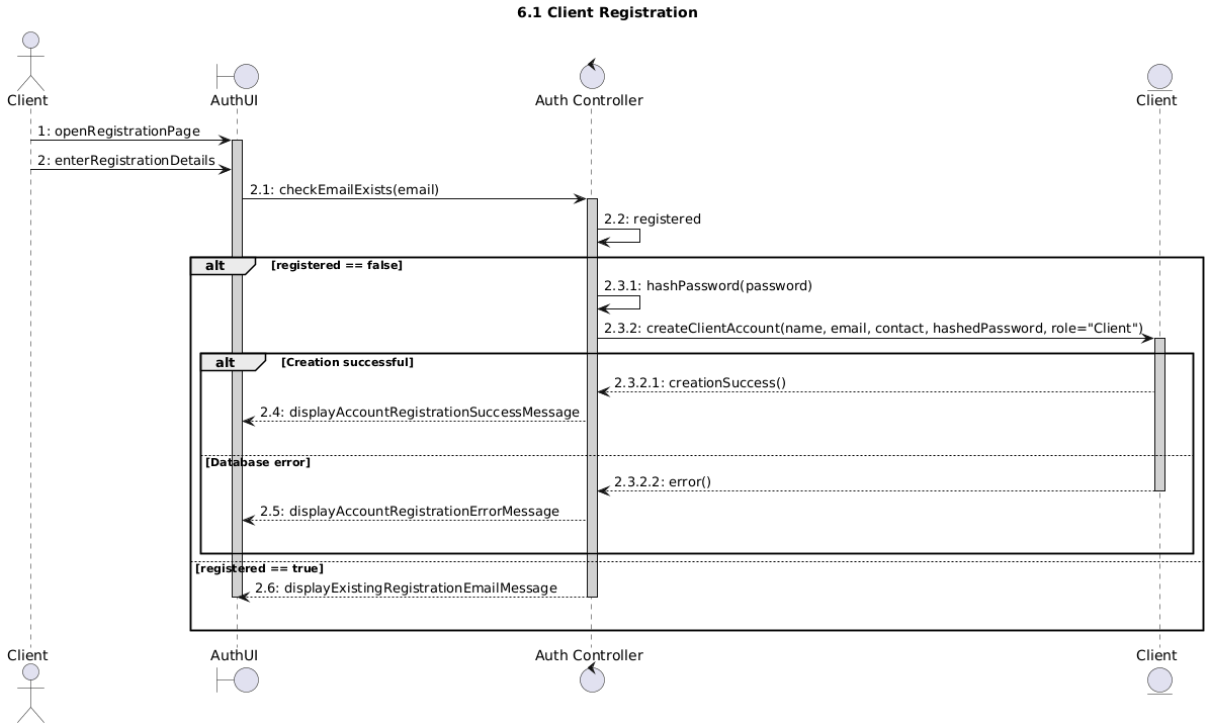


5.3 ExportSubzoneDetails

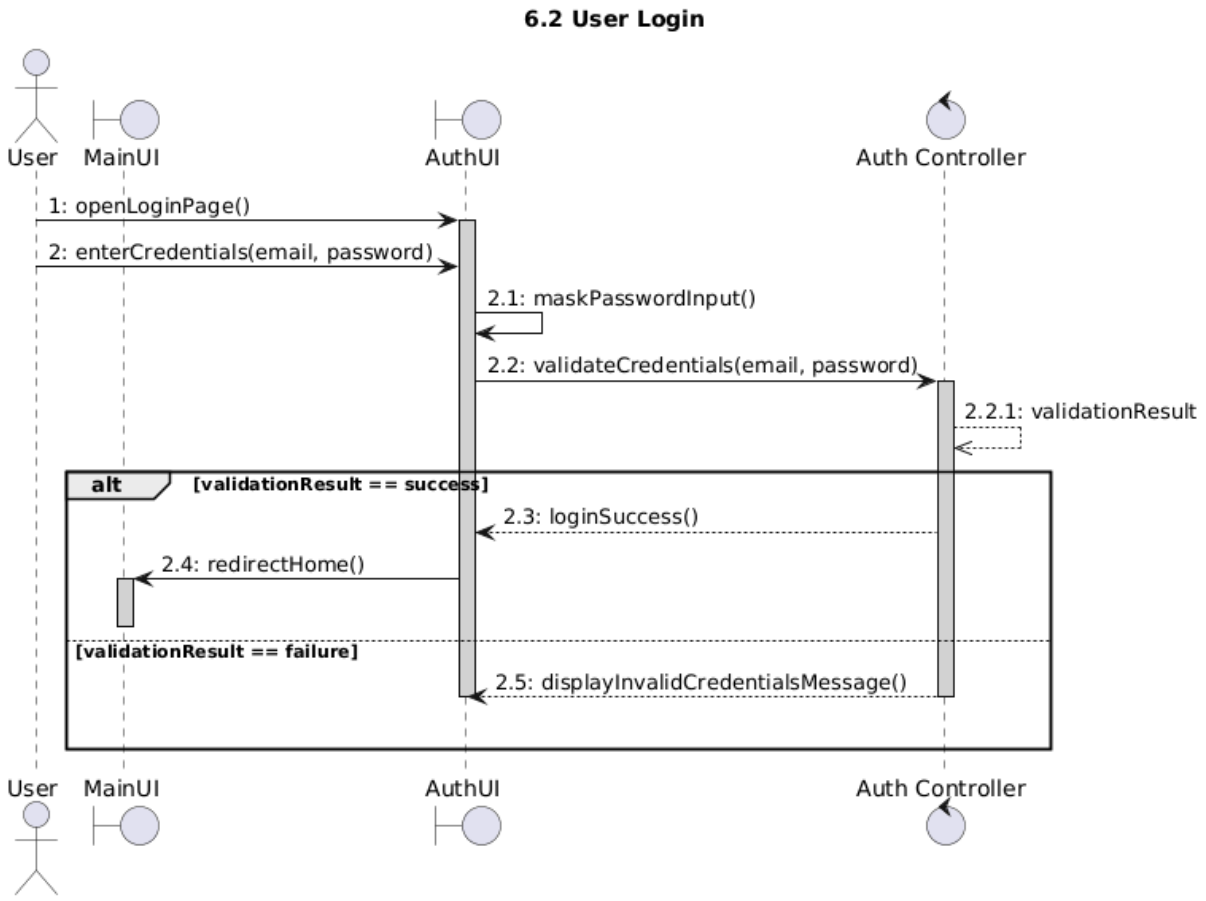


VI. For Functional Requirement #6

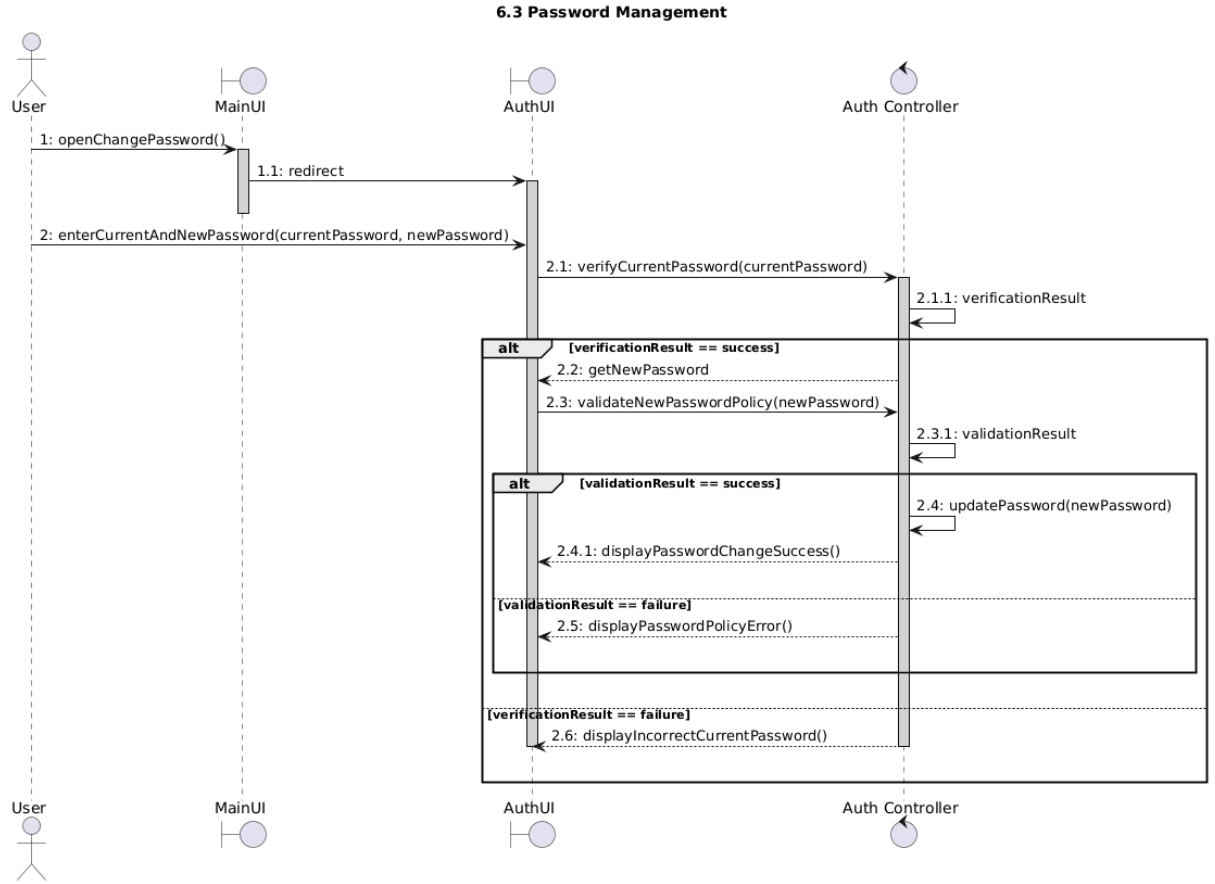
6.1 ClientRegistration



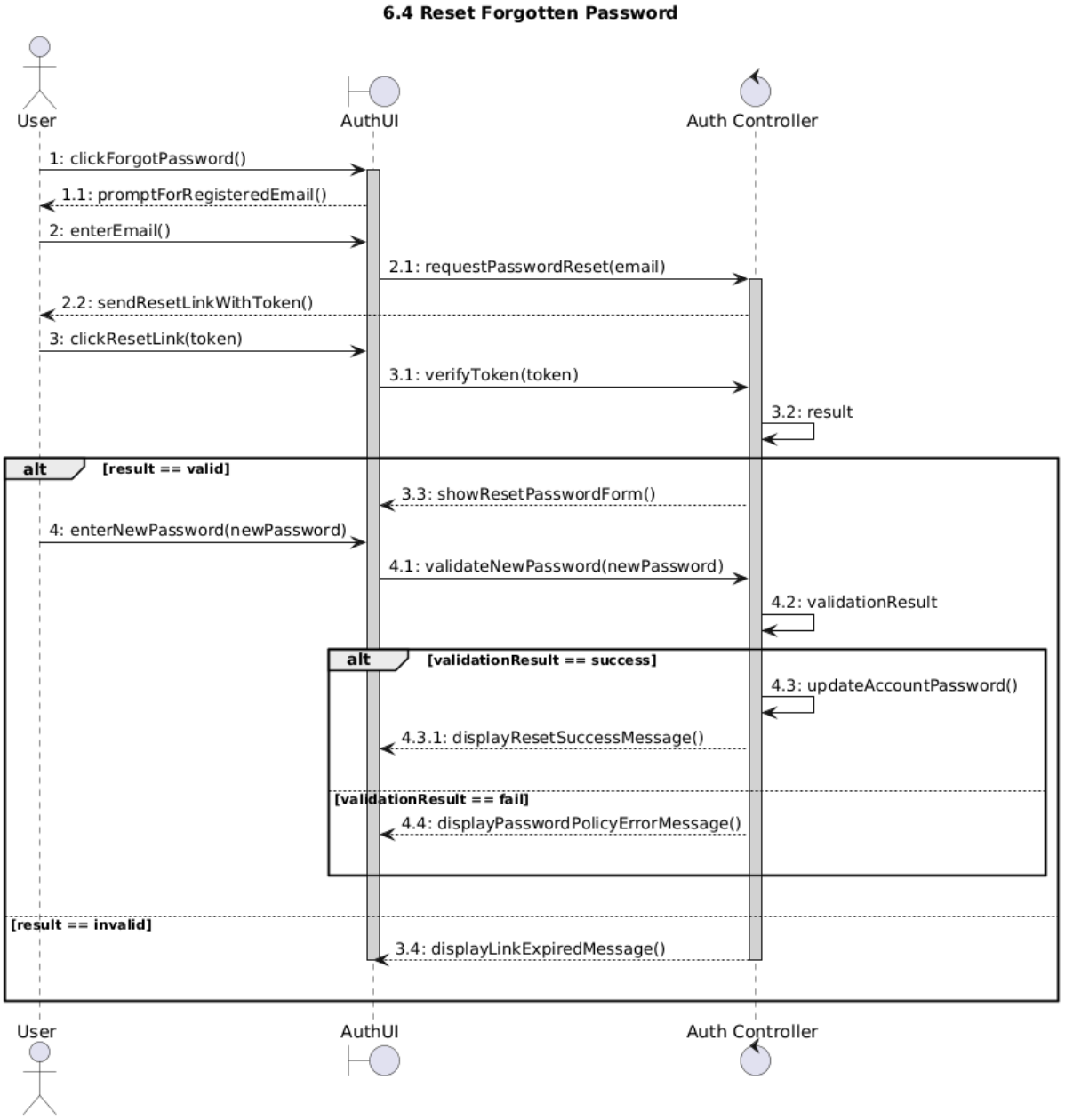
6.2 UserLogin



6.3 PasswordManagement

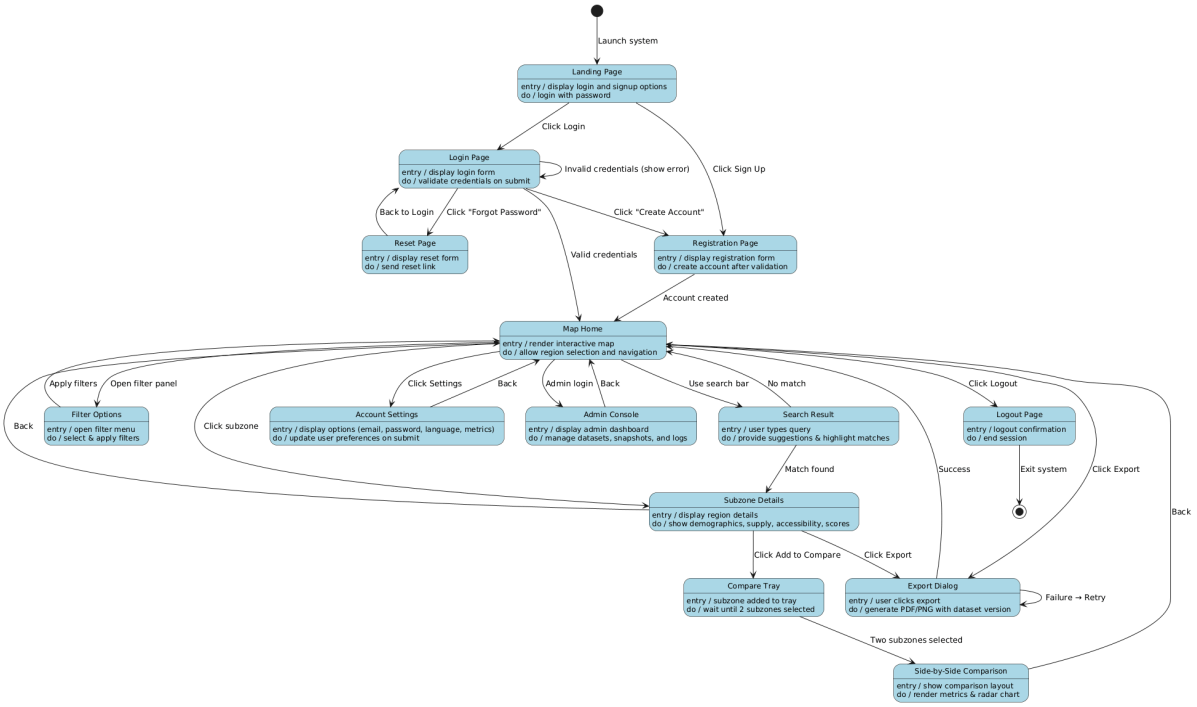


6.4 ResetForgottenPassword



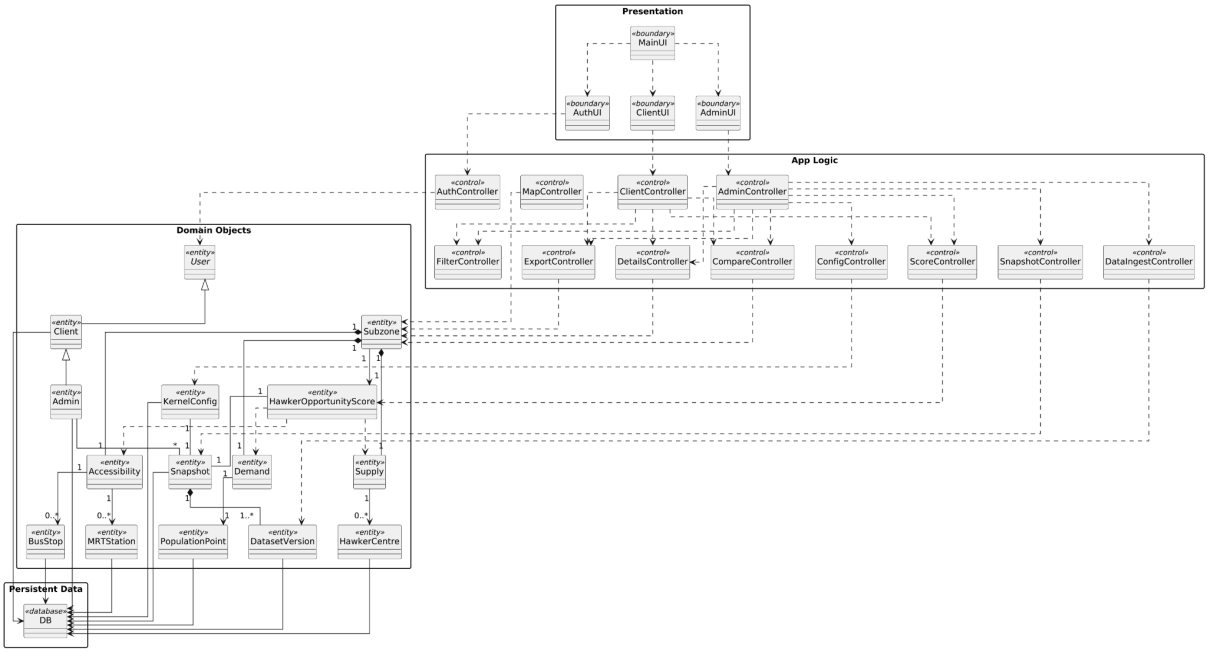
C. Initial Dialog Map

If the image is unclear, please refer to the img file that is uploaded together with this document.



3. System Architecture

If the image is unclear, please refer to the img file that is uploaded together with this document.



4. Application Skeleton

For clarity, please check README.md file in folder Website under the repository

```
sc2006-proj/
├── backend/                              # FastAPI backend (Python)
│   ├── requirements.txt                  # Backend dependencies
│   ├── sql/
│   │   └── 001_init.sql                  # DB schema (snapshots, subzones, users, tokens)
│   └── src/
│       ├── main.py                       # Server entrypoint, CORS, router mounting
│       ├── db/
│       │   └── __init__.py               # SQLAlchemy engine + get_session()
│       ├── controllers/                  # Orchestrates use-cases across services/repos
│       │   ├── admin_controller.py       # Refresh/list/restore snapshots + export
│       │   ├── data_controller.py        # Assemble FeatureCollection, list subzones
│       │   └── auth_controller.py        # Register/login/refresh/logout/me
│       ├── repositories/                 # Data access layer (DB CRUD/queries)
│       │   ├── snapshot_repo.py
│       │   ├── subzone_repo.py
│       │   └── user_repo.py
│       ├── models/
│       │   ├── db_models.py              # SQLAlchemy models: Snapshot, Subzone, User, RefreshToken
│       │   └── kernel_config.py          # (existing)
│       ├── routers/                      # HTTP endpoints
│       │   ├── admin_router.py           # /admin/* (JWT admin only)
│       │   ├── data_router.py            # /data/* (file + DB endpoints)
│       │   ├── auth_router.py            # /auth/* (login/register/...)
│       │   └── deps.py                   # FastAPI deps (DB session, JWT guards)
│       ├── schemas/                      # Pydantic request/response DTOs (expand as needed)
│       └── services/                     # Business logic
│           ├── snapshot_service.py       # Ingest/export snapshots
│           ├── auth_service.py           # Hash/verify, JWT, refresh tokens
│           └── scoring_service.py        # (optional) server-side compute
├── frontend/                             # React + Vite + TypeScript frontend
│   ├── index.html
│   ├── package.json
│   ├── vite.config.ts                    # Dev server + proxy to backend (/data,/auth,/admin)
│   └── src/
│       ├── components/
│       │   └── Map/                      # MapView & layers (Subzones/Hawkers/MRT)
│       ├── contexts/
│       ├── screens/
│       │   ├── MainUI/                   # Main map & exploration UI
│       │   ├── Compare/                  # Side-by-side ComparisonPage
│       │   └── Admin/                    # AdminPage (login, upload, snapshots)
│       ├── services/                     # API client wrappers (data + admin)
│       └── utils/                        # Geo helpers, color scales
├── content/                              # Datasets & the exported GeoJSON used by the map
│   ├── HawkerCentresGEOJSON.geojson
│   ├── LTAMRTStationExitGEOJSON.geojson
│   ├── MasterPlan2019SubzoneBoundaryNoSeaGEOJSON.geojson
│   └── out/
│       └── hawker_opportunities_ver2.geojson   # "current" snapshot export
├── README.md
├── ScoreDemo.py                          # Scoring demo / notebook-style script
└── solve.py                              # Utility script(s)
```