System Specifications

Table of Contents

- S1: User Interface
  - S1.1 Building Panel
  - S1.2 Elevator Panel
- S2: Control Logic
  - S2.1 Elevator State Machine
  - S2.2 Dispatch & Async Tasks
  - S2.3 Input Validation
- S3: Localization
  - S3.1 Language Management
  - S3.2 Text Resources
  - S3.3 Unit Tests
- S4: Theming
  - S4.1 OS Theme Detection
  - S4.2 Runtime Toggling
  - S4.3 Style Application
- S5: ZeroMQ External API
  - S5.1 Command Ingestion
  - S5.2 Event Publishing
  - S5.3 API Integration Tests
- S6: Runtime Configuration
  - S6.1 Configuration API
  - S6.2 Configuration UI
  - S6.3 Dynamic Application
  - S6.4 Configuration Tests

S1: User Interface

S1.1 Building Panel

- S1.1.1 Provide up/down call buttons per floor with highlight.
- S1.1.2 Send call requests on click.

- S1.1.3 Clear call highlight on completion.
- S1.1.4 Reset buttons on arrival.

S1.2 Elevator Panel

- S1.2.1 Provide floor selection buttons.
- S1.2.2 Provide door open/close buttons.
- S1.2.3 Update status display in real time.

S2: Control Logic

S2.1 Elevator State Machine

- S2.1.1 Move elevator asynchronously and emit floor-change events.
- S2.1.2 Operate doors asynchronously and emit state-change events.
- S2.1.3 Manage current, next, and future target queues.
- S2.1.4 Commit floor requests and await arrival.
- S2.1.5 Commit door actions and await completion.

S2.2 Dispatch & Async Tasks

- S2.2.1 Deduplicate and run async tasks.
- S2.2.2 Assign elevators using GREEDY or OPTIMAL strategy.
- S2.2.3 Handle hall calls and emit completion.
- S2.2.4 Handle car selections and emit arrival.
- S2.2.5 Adjust elevator count/config dynamically.
- S2.2.6 Calculate `estimate_total_duration` per request.
- S2.2.7 Implement `optimal_reassign` to minimize wait time.

S2.3 Input Validation

- S2.3.1 Ignore duplicate hall calls.
- S2.3.2 Ignore duplicate car selections.

S3: Localization

S3.1 Language Management

- S3.1.1 Load and install translation files.
- S3.1.2 Switch language and notify UI.

S3.2 Text Resources

- S3.2.1 Dynamically update all UI texts.

S3.3 Unit Tests

- S3.3.1 Cover language switching and text updates.

S4: Theming

S4.1 OS Theme Detection

- S4.1.1 Detect system light/dark mode.

S4.2 Runtime Toggling

- S4.2.1 Switch theme and emit update event.

S4.3 Style Application

- S4.3.1 Provide theme-specific styles.
- S4.3.2 Apply stylesheet on theme change.

S5: ZeroMQ External API

S5.1 Command Ingestion

- S5.1.1 Receive and forward commands.

S5.2 Event Publishing

- S5.2.1 Publish door and arrival events.

S5.3 API Integration Tests

- S5.3.1 Validate via integration tests.

S6: Runtime Configuration

S6.1 Configuration API

- S6.1.1 Expose `set_config` and `set_elevator_count` as async tasks.

S6.2 Configuration UI

- S6.2.1 Provide live parameter controls.

S6.3 Dynamic Application

- S6.3.1 Apply new settings immediately.

S6.4 Configuration Tests

- S6.4.1 Cover config logic in unit tests.