APRIL 28, 2025

TIMELINEXPRESS SYSTEM DESIGN DOCUMENT
DRAFT 0.4 FOR REVIEW

ALPHA TEAM
Alexandra Lawler
Dana Tryon
Gitti Esmailzada
Walter Ulicki

TABLE OF CONTENTS

INTRODUCTION

Description of the Application

The Timeline app is a user-friendly, interactive application designed to help people explore historical events in an engaging and accessible way. The app lets users view, compare, and interact with historical timelines, offering a visual and customizable approach to learning about the past.

Events are stored in an integrated database, allowing users to search for key historical moments, create and manage entries, and explore timelines based on themes, periods, or event types.

Purpose of This Design Document

The purpose of this document is to define the design details, and functionality, of the Timeline app. It serves as a shared reference point for the development team, designers, and stakeholders through the implementation phase of the project.

By outlining the app's core features, technical components, and performance expectations, this document ensures that everyone involved in the project has a mutual understanding of what the application will do, and how it will work.

What This Document Contains

This design document includes the following sections:

- Introduction, including a high-level overview of the Timeline app.

- A design level class diagram, including pseudocode for each class.

- Statechart diagrams for each class

- Design level sequence diagrams for each use case identified in the System Requirements Document

Reference Documents

TimelineXpress Project Plan, February 17, 2025.

TimelineXpress System Requirements Document, March 26, 2025
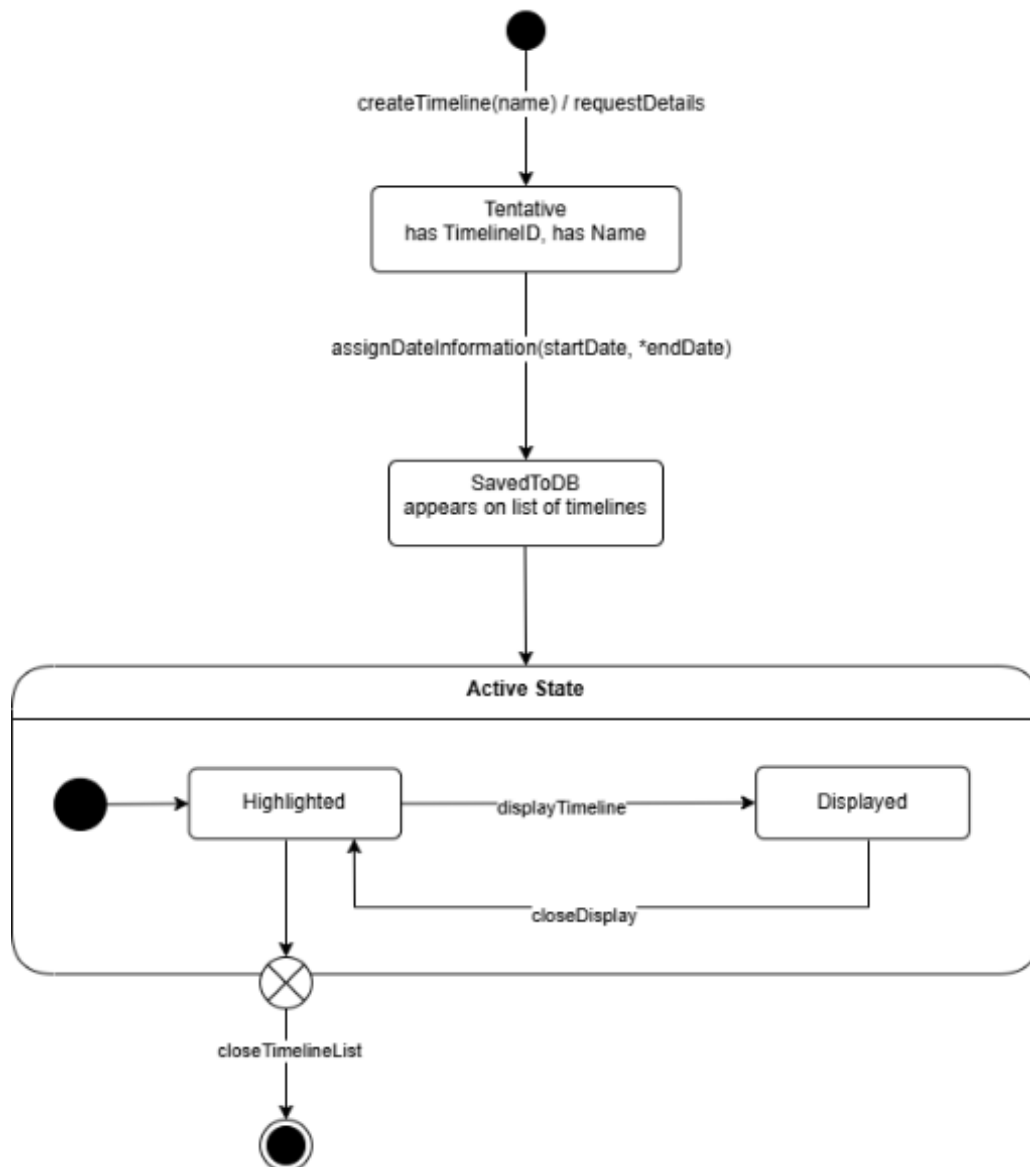
DESIGN LEVEL CLASS DIAGRAM

Refine your Class Diagram from your Analysis phase to contain the following additional information:

- Attributes: data types, initial values, range of values (if appropriate)
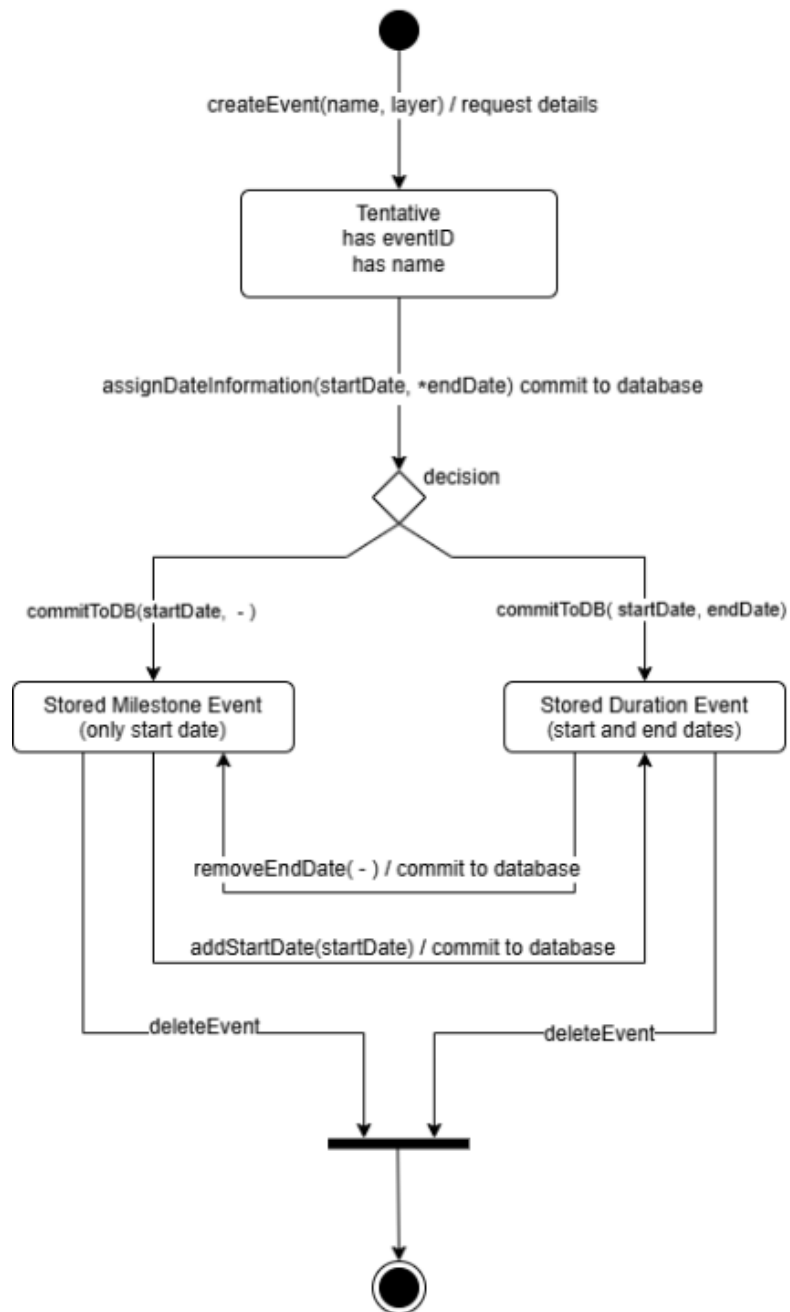- Methods: return types, parameters
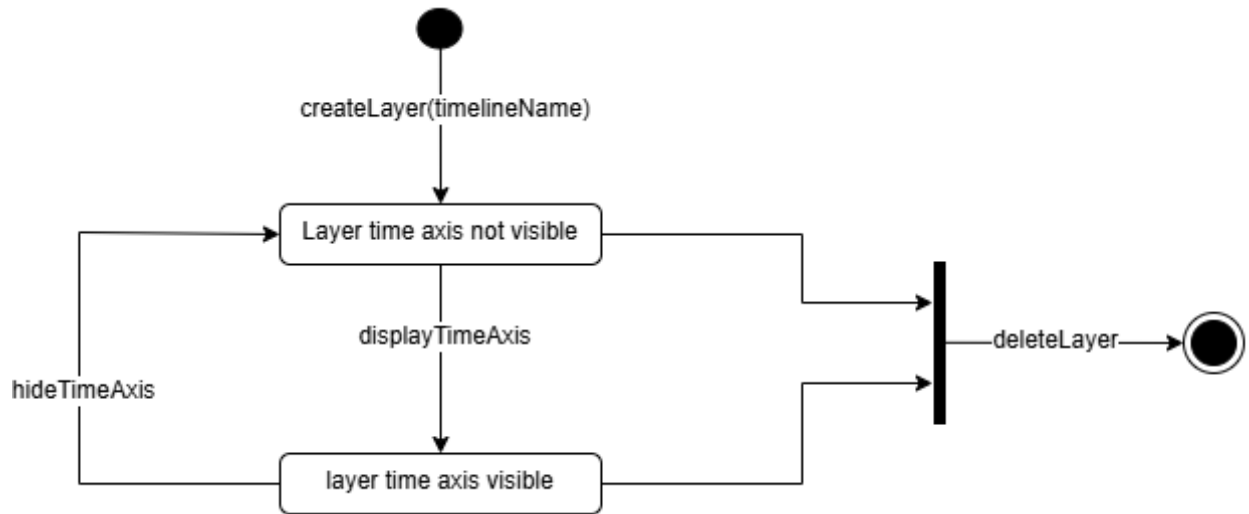- Pseudocode for each method

STATECHART DIAGRAMS

TIMELINE CLASS

EVENT CLASS

createEvent(name, layer) / request details

Tentative
has eventID
has name

assignDateInformation(startDate, *endDate) commit to database

decision

commitToDB(startDate, -)

commitToDB(startDate, endDate)

Stored Milestone Event
(only start date)

Stored Duration Event
(start and end dates)

removeEndDate(-) / commit to database

addStartDate(startDate) / commit to database

deleteEvent

deleteEvent
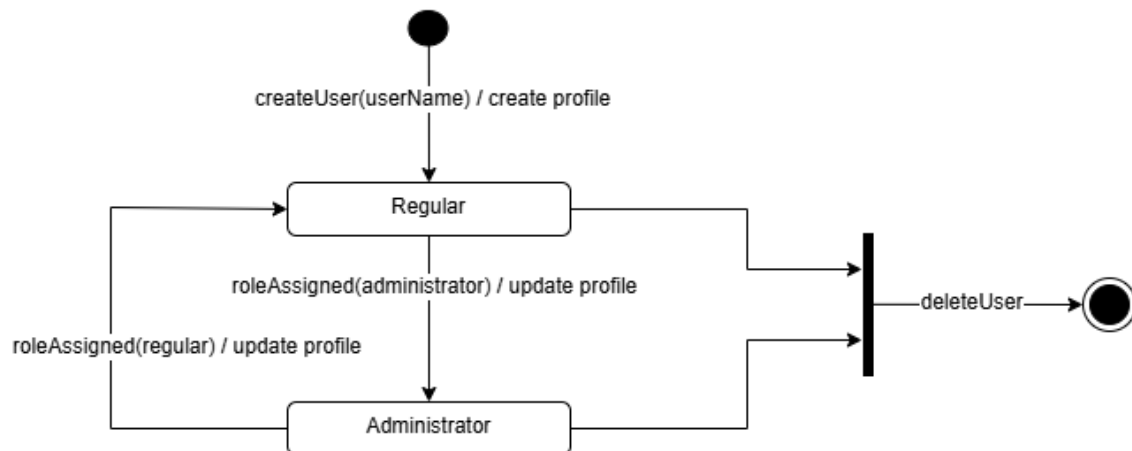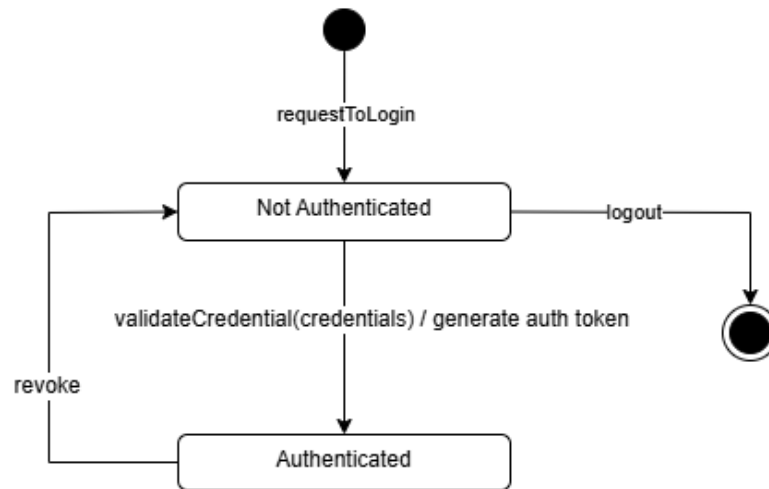
VISUALIZATION CLASS
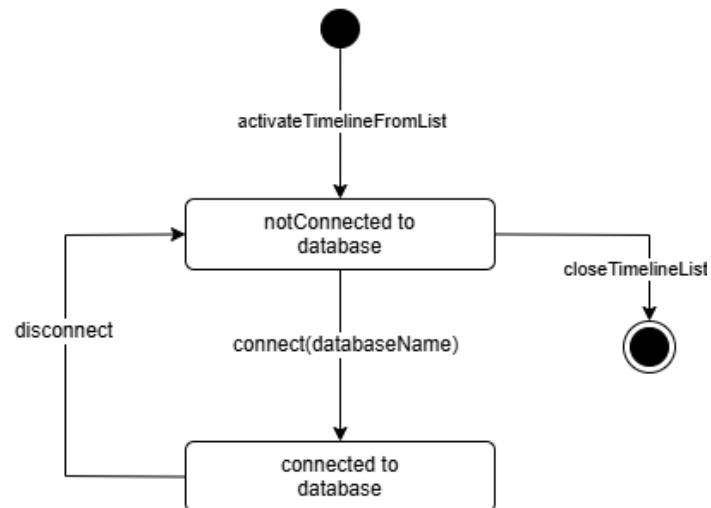

USER CLASS

AUTHENTICATION CLASS

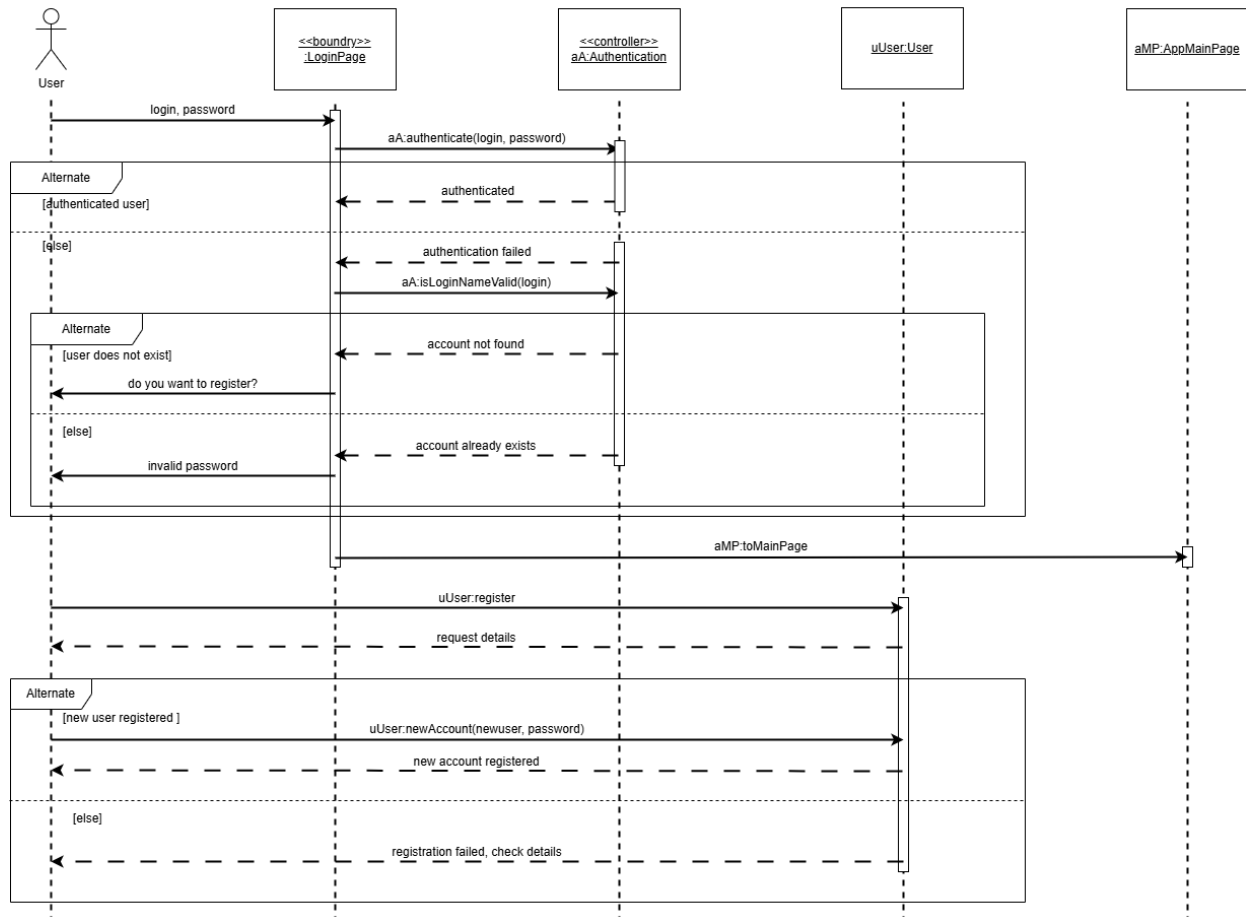# DATABASEMANAGER CLASS

SEARCHENGINE CLASS
The SearchEngine class provides only services, and thus does not have a state machine diagram.

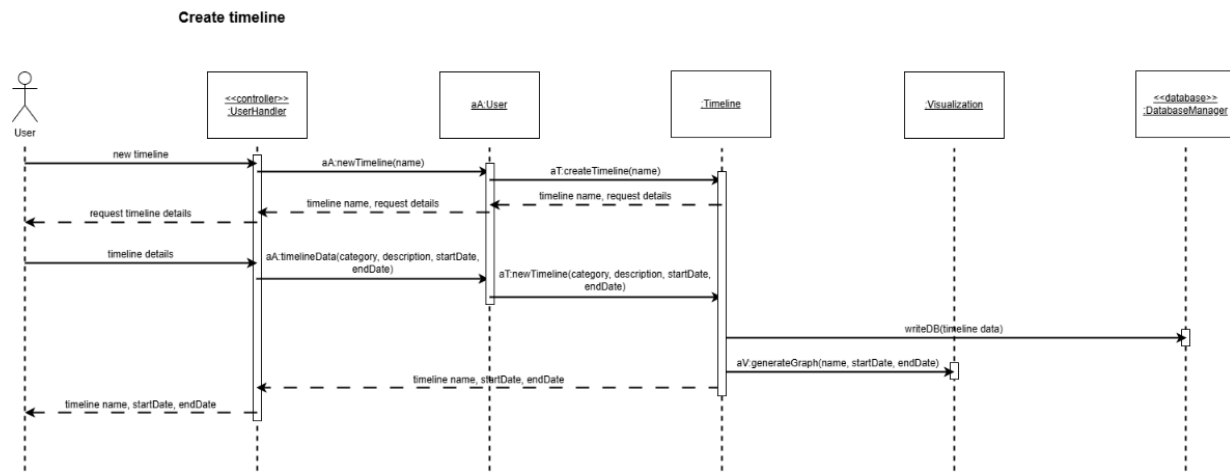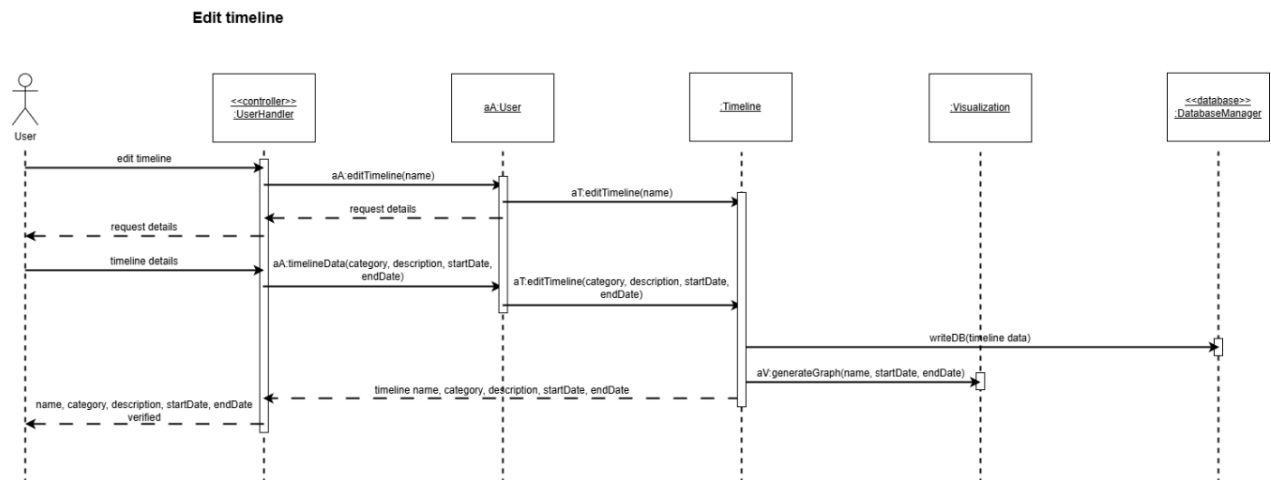# SEQUENCE DIAGRAMS

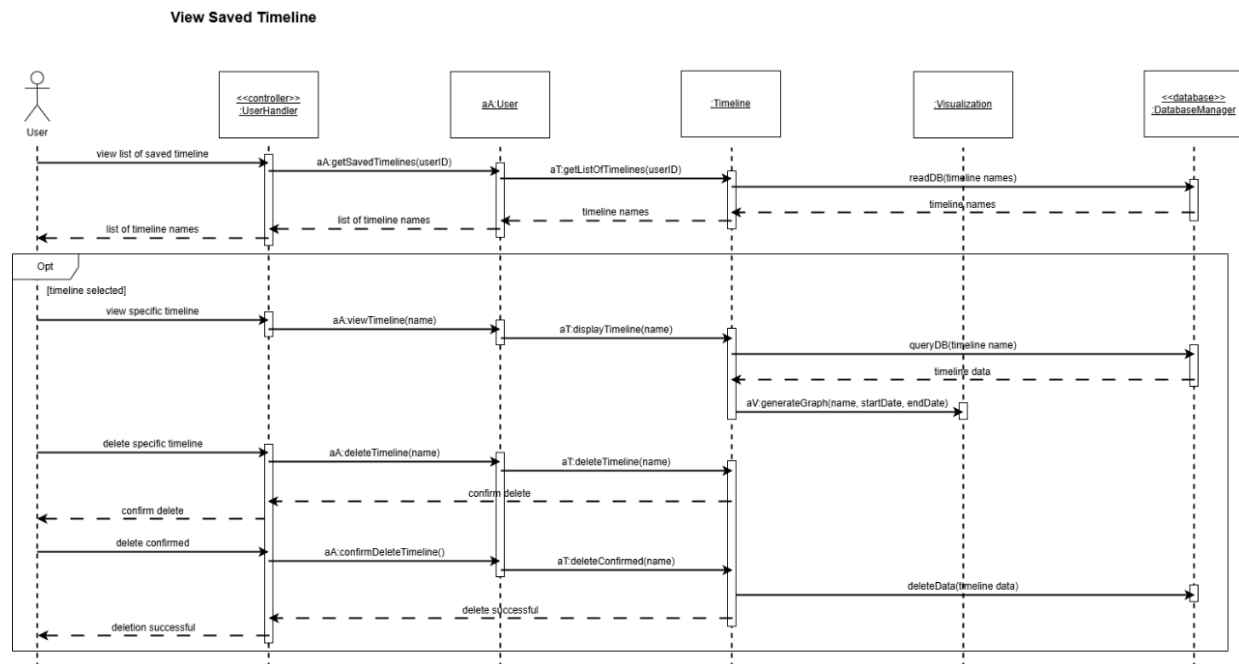## USE CASE 1: LOGIN TO TIMELINEXPRESS

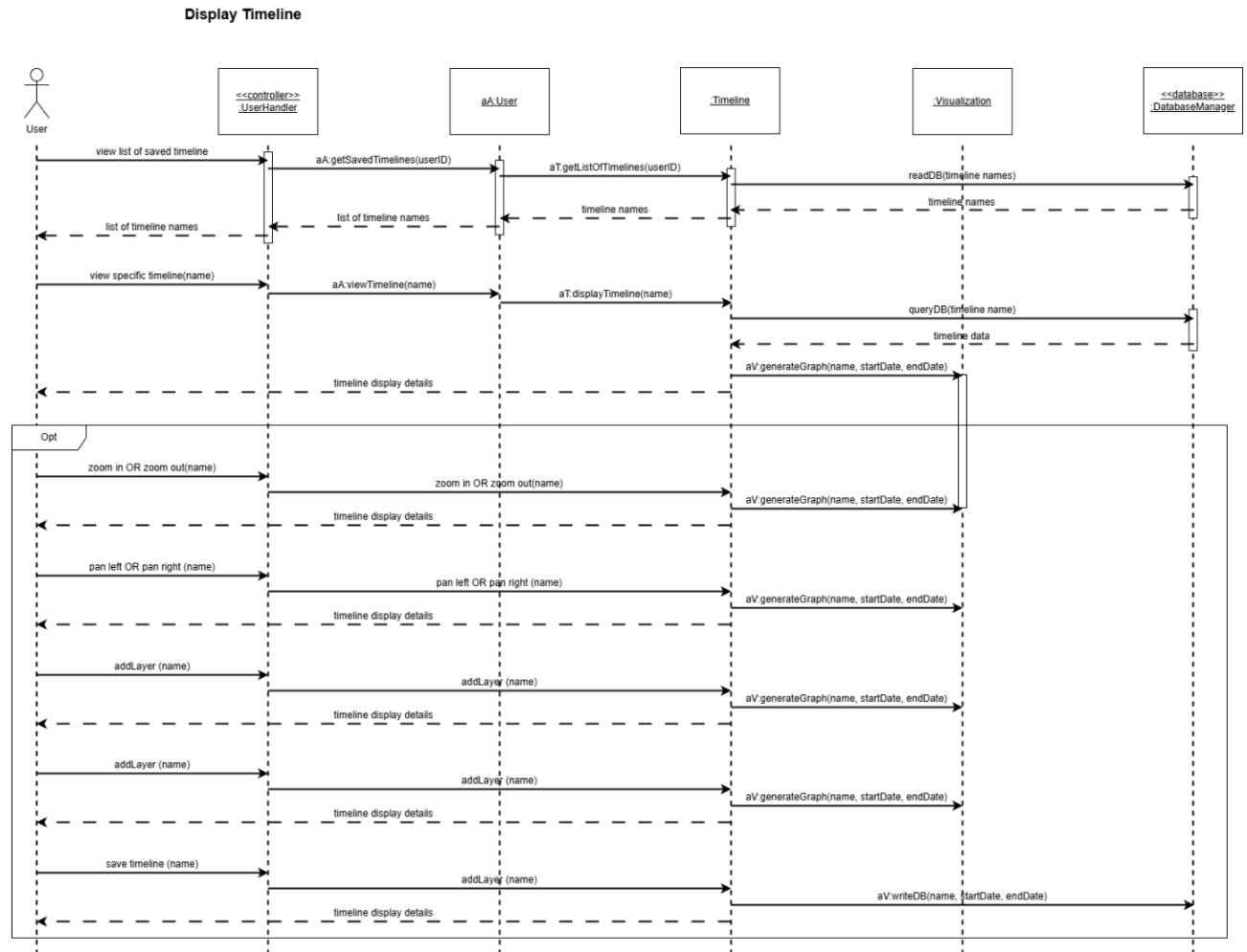
Login to TimelineXpress

USE CASE 2: CREATE TIMELINE

Create timeline

USE CASE 3: EDIT TIMELINE

Edit timeline

USE CASE 4: VIEW SAVED TIMELINES

View Saved Timeline

USE CASE 5: DISPLAY A TIMELINE

Display Timeline

User	<<controller>> :UserHandler	aA:User	:Timeline	:Visualization	<<database>> :DatabaseManager

view list of saved timeline → aA:getSavedTimelines(userID) → aT:getListOfTimelines(userID) → readDB(timeline names) →

timeline names ← list of timeline names ← timeline names ←

list of timeline names ←

view specific timeline(name) → aA:viewTimeline(name) → aT:displayTimeline(name) → queryDB(timeline name) →

timeline data ←

aV:generateGraph(name, startDate, endDate)

timeline display details ←

Opt

zoom in OR zoom out(name) → zoom in OR zoom out(name) → aV:generateGraph(name, startDate, endDate)

timeline display details ←

pan left OR pan right (name) → pan left OR pan right (name) → aV:generateGraph(name, startDate, endDate)

timeline display details ←

addLayer (name) → addLayer (name) → aV:generateGraph(name, startDate, endDate)

timeline display details ←

addLayer (name) → addLayer (name) → aV:generateGraph(name, startDate, endDate)

timeline display details ←

save timeline (name) → addLayer (name) → aV:writeDB(name, startDate, endDate)

timeline display details ←

USE CASE 6: CREATE EVENT

Create Event

USE CASE 7: EDIT EVENT

Edit Event