TIMELINEXPRESS SYSTEM DESIGN DOCUMENT

DRAFT 0.3 FOR REVIEW

ALPHA TEAM

Alexandra Lawler
Dana Tryon
Gitti Esmailzada
Walter Ulicki

TABLE OF CONTENTS

INTRODUCTION

Description of the Application

The ~~History~~ Timeline App is a user-friendly, interactive application designed to help people explore historical events in a~~n more~~ engaging and accessible way. ~~At its core,~~ 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 a~~n~~ integrated ~~connected~~ 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 ~~design~~ document is to define the design details ~~goals,~~ and functionality, ~~and system requirements~~ of the ~~History~~ Timeline App. It serves as a ~~clear and~~ shared reference point for the development team, designers, and stakeholders throughout the ~~planning and building~~ implementation phase of the project ~~process.~~

By outlining the app's core features, ~~user needs,~~ 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, ~~and why it matters~~.

What This Document Contains

This design document includes the following sections:

- Introduction, including a high-level overview of ~~what~~ the ~~History~~ Timeline App ~~is and who it's for~~.

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

- Statechart diagrams for each class

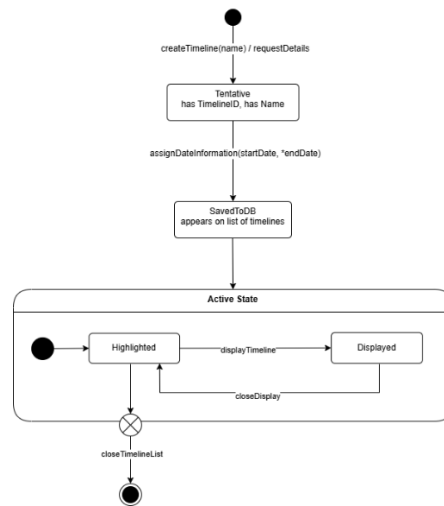- Design level sequence diagrams for each use case identified in the System Requirements Document

DESIGN LEVEL CLASS DIAGRAM

Refine your Class Diagram from your Analysis phase to contain the following additional information:
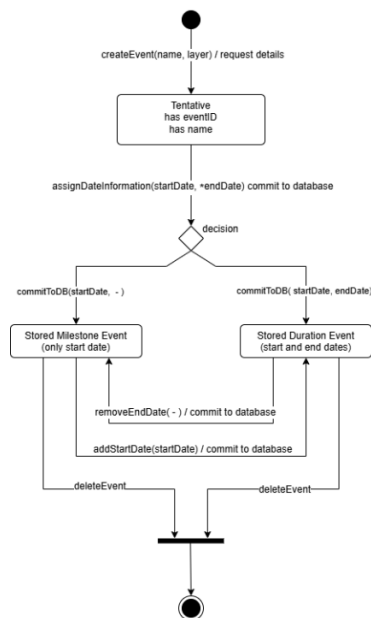
- Attributes: data types, initial values, range of values (if appropriate)
- Methods: return types, parameters
- Pseudocode for each method

STATECHART DIAGRAMS

TIMELINE CLASS

createTimeline(name) / requestDetails

Tentative
has TimelineID, has Name

assignDateInformation(startDate, *endDate)

SavedToDB
appears on list of timelines

Active State

Highlighted — displayTimeline → Displayed

closeDisplay

closeTimelineList

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

createUser(userName) / create profile

Regular

roleAssigned(administrator) / update profile

roleAssigned(regular) / update profile

Administrator

deleteUser

AUTHENTICATION CLASS

requestToLogin

Not Authenticated

logout

validateCredential(credentials) / generate auth token

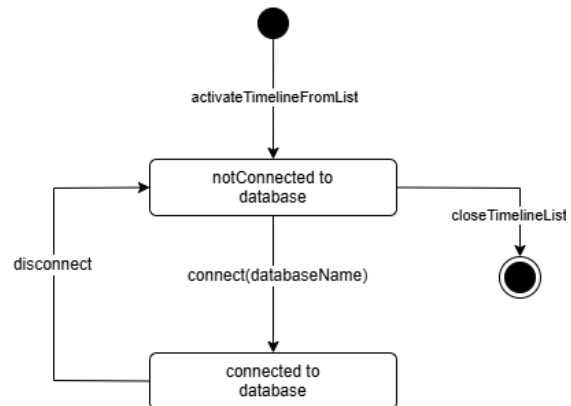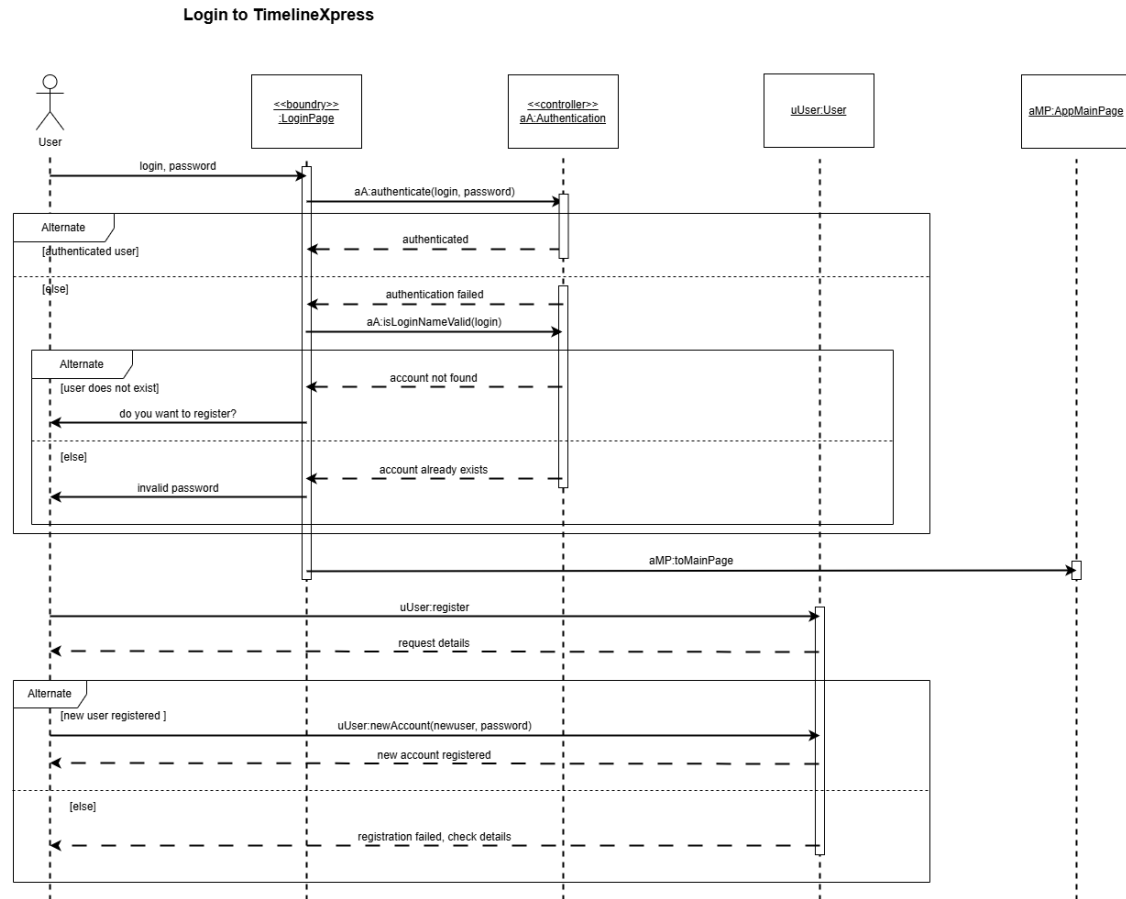revoke
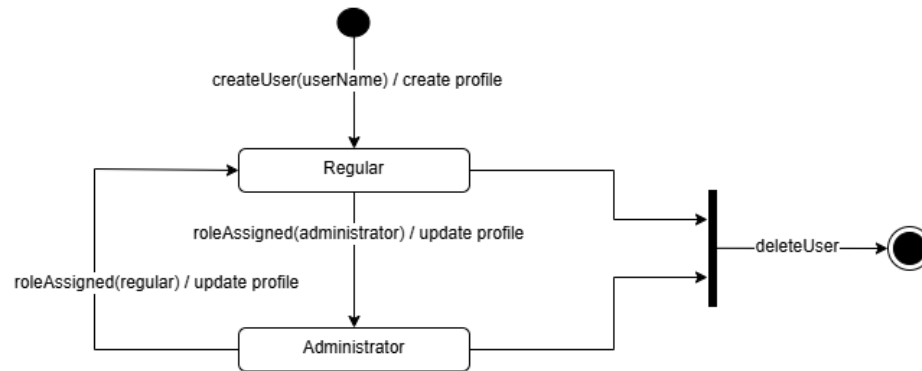
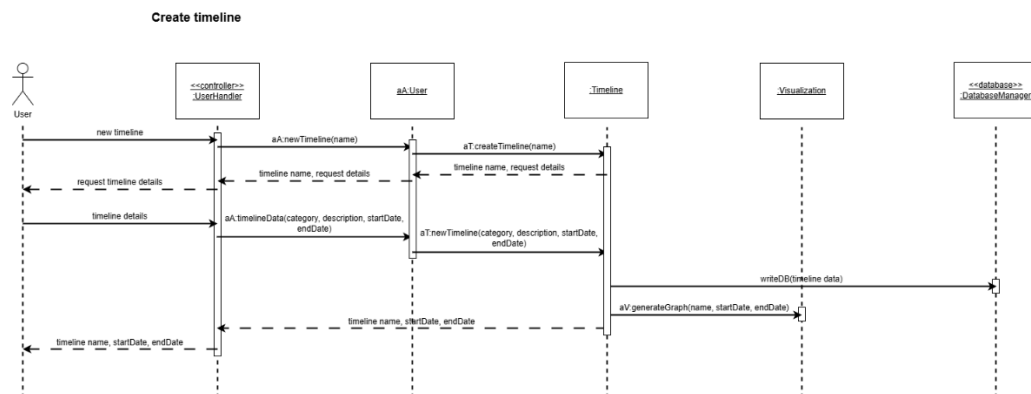Authenticated
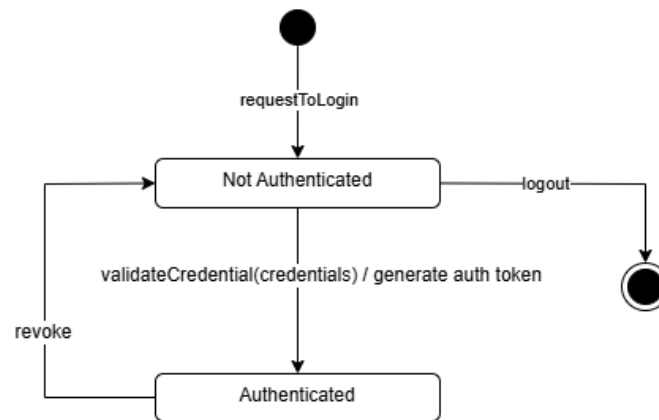
DATABASEMANAGER CLASS

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

SEQUENCE DIAGRAMS

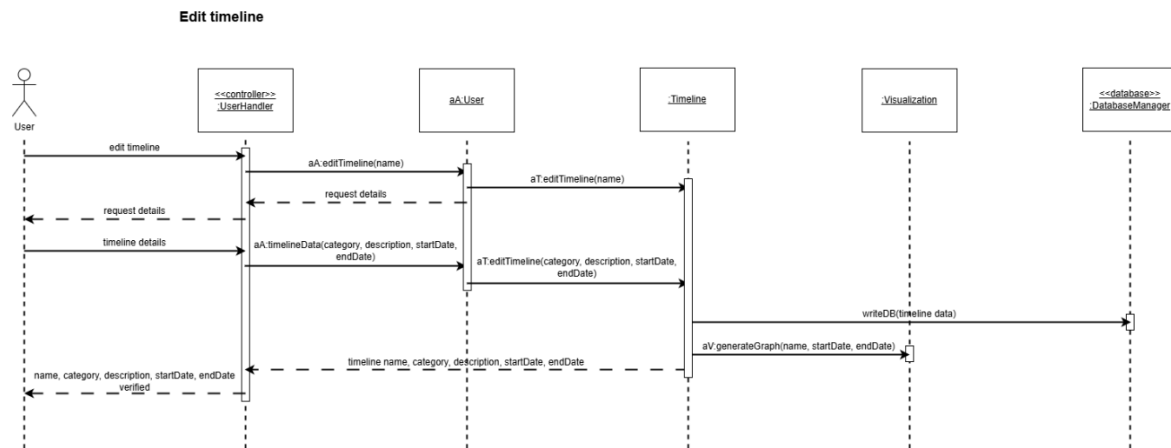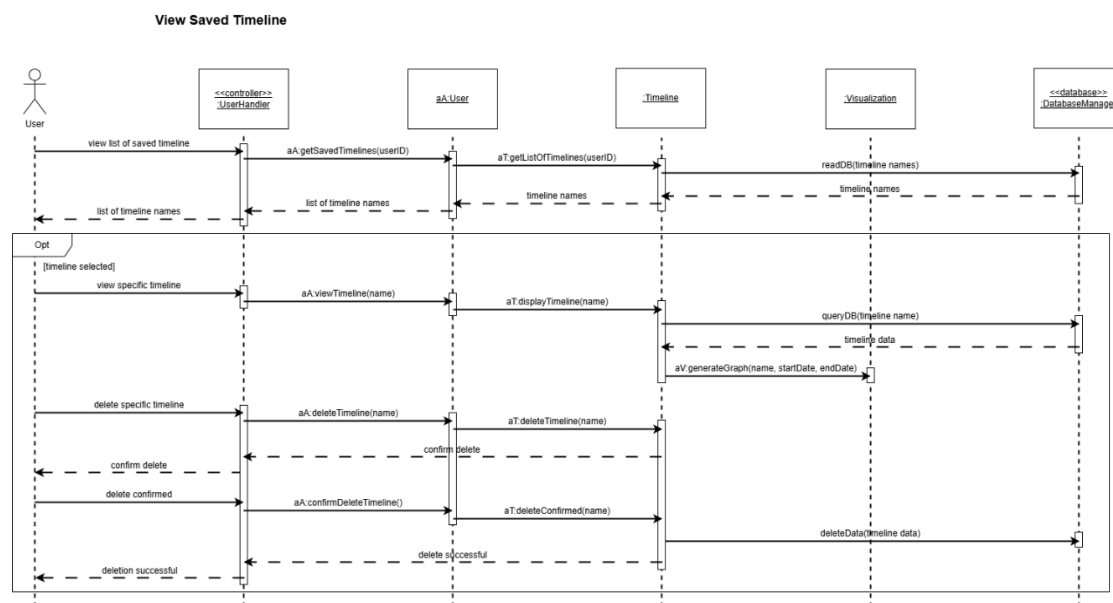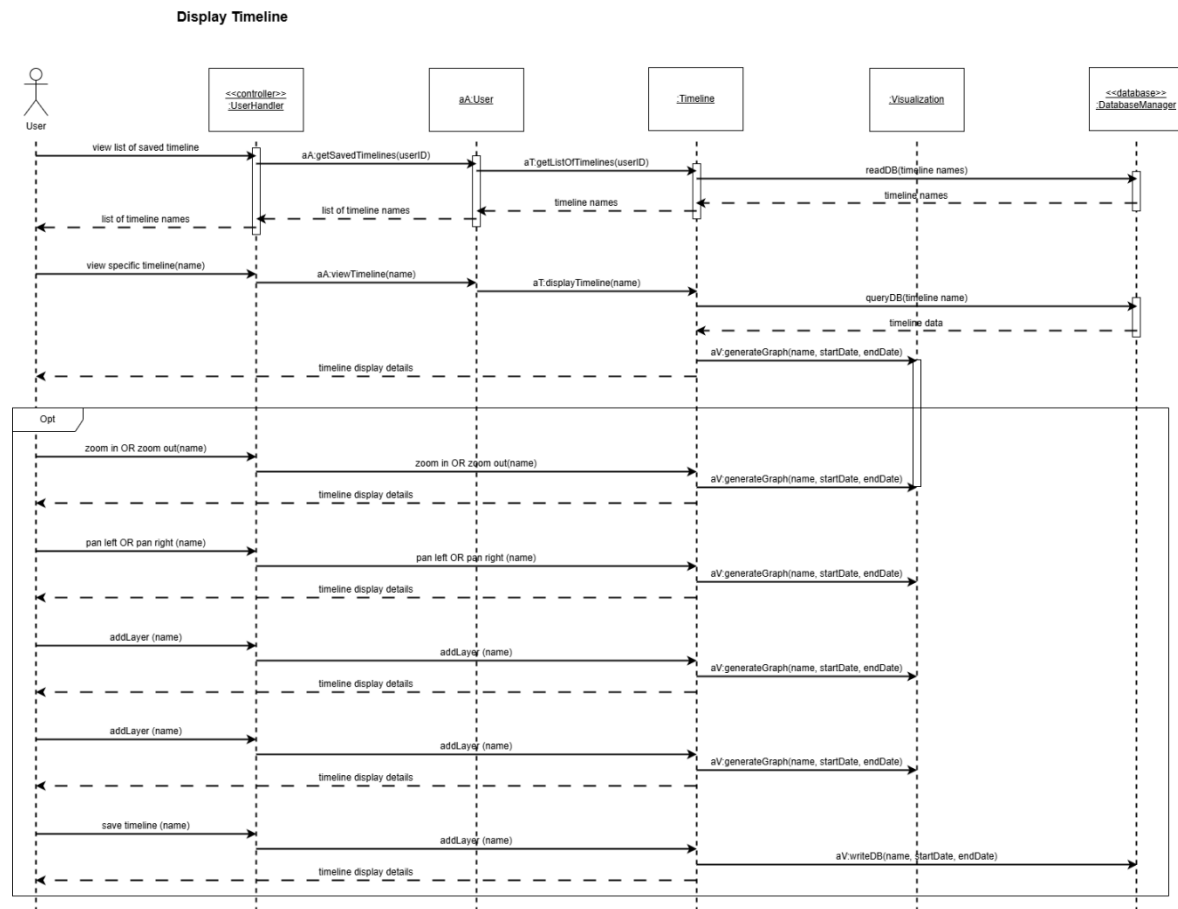## USE CASE 1: LOGIN TO TIMELINEXPRESS

Login to TimelineXpress

User   <<boundry>> :LoginPage   <<controller>> aA:Authentication   uUser:User   aMP:AppMainPage

- login, password
- aA:authenticate(login, password)

Alternate

[authenticated user]
- authenticated

[else]
- authentication failed
- aA:isLoginNameValid(login)

Alternate

[user does not exist]
- account not found
- do you want to register?

[else]
- account already exists
- invalid password

- aMP:toMainPage

- uUser:register
- request details

Alternate

[new user registered]
- uUser:newAccount(newuser, password)
- new account registered

[else]
- registration failed, check details

USE CASE 2: CREATE TIMELINE

Create timeline

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

- new timeline
- aA:newTimeline(name)
- aT:createTimeline(name)
- timeline name, request details
- timeline name, request details
- request timeline details
- timeline details
- aA:timelineData(category, description, startDate, endDate)
- aT:newTimeline(category, description, startDate, endDate)
- writeDB(timeline data)
- aV:generateGraph(name, startDate, endDate)
- timeline name, startDate, endDate
- timeline name, startDate, endDate

USE CASE 3: EDIT TIMELINE

Edit timeline

USE CASE 4: VIEW SAVED TIMELINES

View Saved Timeline

USE CASE 5: DISPLAY A TIMELINE

Display Timeline

USE CASE 6: CREATE EVENT

Create Event

USE CASE 7: EDIT EVENT

Edit Event