JetLAG

Student Organization Software
Software Architecture Document

Version 1.1

Revision History

Date	Version	Description	Author
10/23/2022	1.0	First iteration of document.	JetLAG Team
10/28/2022	1.1	Second iteration	JetLAG Team

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This Software Architecture Document covers the architectural decisions that have been made in creating the Student Organizational Software.

1.2 Scope

This document provides insight into the architectural design of the Student Organizational Software, including the Goals and Constraints which guided development. Other documents are referenced to provide further information.

1.3 Definitions, Acronyms, and Abbreviations

See the glossary provided in the Supplementary Requirements Specifications document.

1.4 References

1. SOS Supplementary Specifications Document – Glossary
2. SOS Use Case Realization Document
3. SOS Software Requirements Specifications
4. … (Adding as Referenced)

1.5 Overview

Sections 2 through 8 cover the various views of the design of SOS, goals and constraints, as well as performance and quality goals.

2. Architectural Representation

The views and models presented in this document as well as the Use Case Realization document are represented in Unified Modeling Language.

3. Architectural Goals and Constraints

SOS is designed to be an online software solution users can access in browser and on mobile devices. SOS is composed of three main components: a Flutter front-end app, a data-broker, and a back-end Cosmos DB database to store information.

4. Use-Case View

The Use-Case View models the scenarios and use cases that are the focus in development. They show what scenarios and use cases have been taken under consideration in the architectural design of SOS.

4.1 Use-Case Realizations

Refer to Use Case Realizations document.

5. Logical View

5.1 Overview

In our diagram you can see we have 3 packages representing the 3 components of our system, the mobile app, data Broker, and database. In general the mobile app makes api requests to the data broker, from there the data broker determines what data I needs to save and retrieve from the database.

5.2 Architecturally Significant Design Packages

Figure 1:High Level Package Diagram

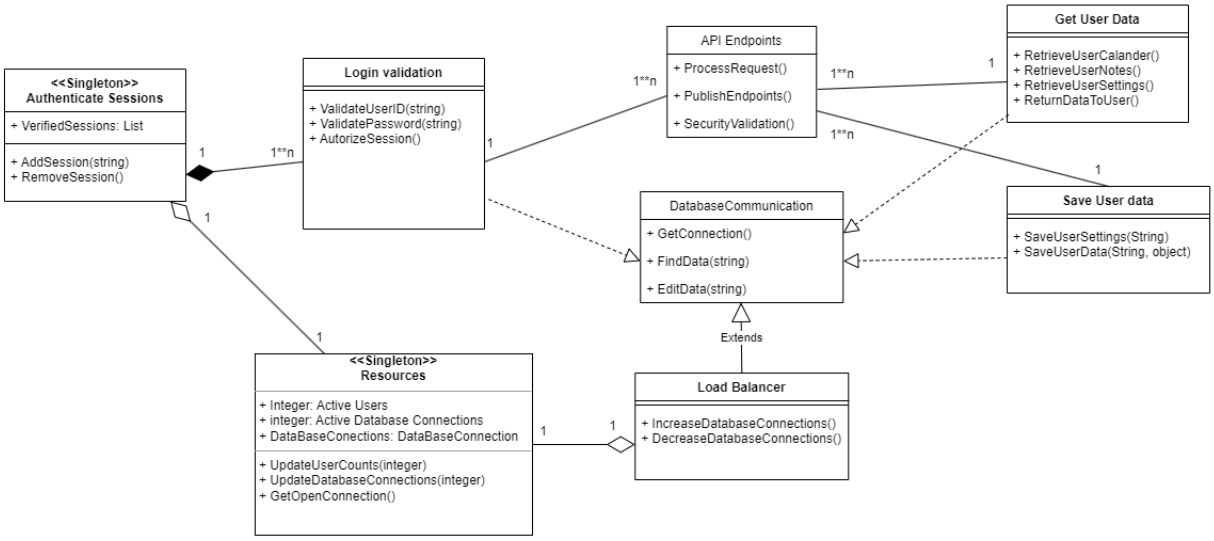

Figure 2:Backend Data Broker Software Architecture

Property	Description
Name	API Endpoints
Description	Handles the creation and use of endpoints
Responsibilities	Create endpoints and communicates with any user interface that connects to the data broker
Relations	Association with login validation, Get User Data, and save user data
Methods	ProcessRequest()-when a request is received, directs traffic to appropriate module PublishEnpoints()-Creates an open endpoint for UI's to use SecurityValidation()-when a request is received, it checks to make sure the request is from a valid UI
Attributes	none

Property	Description
Name	Get User Data
Description	When a request for data is received, it fetches the data from the database
Responsibilities	Retrieve correct data for ui from the database, and returns to correct endpoint
Relations	Association with api endpoint and implements database communication
Methods	RetriveUserCalander()-gets the calendar for the corresponding user RetriveUserNotes()-gets the user notes for the corresponding user RetriveUserSettings()-gets the settings for the corresponding user ReturnDataToUser()-returns the fetched data to the user
Attributes	none

Property	Description
Name	Save User Data
Description	Saves data from a user to the database upon request from the UI
Responsibilities	Saving user data to the database
Relations	Association with API endpoints and implementation of database communication
Methods	SaveUserSettings()-saves settings that the user wants to the database SaveUserData()-saves all other data besides user settings to the database
Attributes	None

Property	Description
Name	Database communication
Description	responsible for creation and maintain communication with the database
Responsibilities	Any communication between the data broker and database
Relations	Implemented by get user data and save user data and login validation
Methods	GetConnection()-gets a new connection to the database FindData(string)-finds data in the database and returns it to the method that called it EditData(string)-changes, adds, or deletes data in the database
Attributes	None

Property	Description
Name	Load Balancer
Description	Scales open database connections with UIs connected
Responsibilities	Add and remove database connections
Relations	Extends database connections, is a component of reasorces
Methods	IncreaseDatabseConnections()-add aa database connection DecreaseDatabseConnections()-removes a database connection
Attributes	None

Property	Description
Name	Resources
Description	Monitors the current state of the dataBroker
Responsibilities	Monitors user usage and adjusts database connections accordingly
Relations	Composed of the load balancer and authenticate sessions
Methods	UpdateUserCounts(integer)-keeps a count of the number of active users using the app UpdateDatabaseConnections(integer)-keeps a count of active database connections GetOpenConnection()-returns an unused database connection
Attributes	ActiveUsers-number of users using the ui ActiveDatabaseConnections-the number of database connection open DatabaseConnections- List of open Database connections

Property	Description
Name	Login Validation
Description	Used for validate login attempts
Responsibilities	Validating login attempts
Relations	Composed of authenticate Sessions associated with API endpoints, implements database communication
Methods	ValidateUserID(string)- checks to make sure the userID exists ValidatePassword(string)-checks to make sure password matches corresponding user AuthorizeSession()-if both password and user are valid, add the session to authenticate sessions
Attributes	

Property	Description
Name	Authenticate Sessions
Description	Handles what sessions are valid
Responsibilities	Add, remove, and validate sessions
Relations	Component of login Validation and Reasorces
Methods	AddSession(string)- adds a valid session to verified session RemoveSession()-removes a session from verified session
Attributes	VerifiedSession-List of session that are Verified at that given point

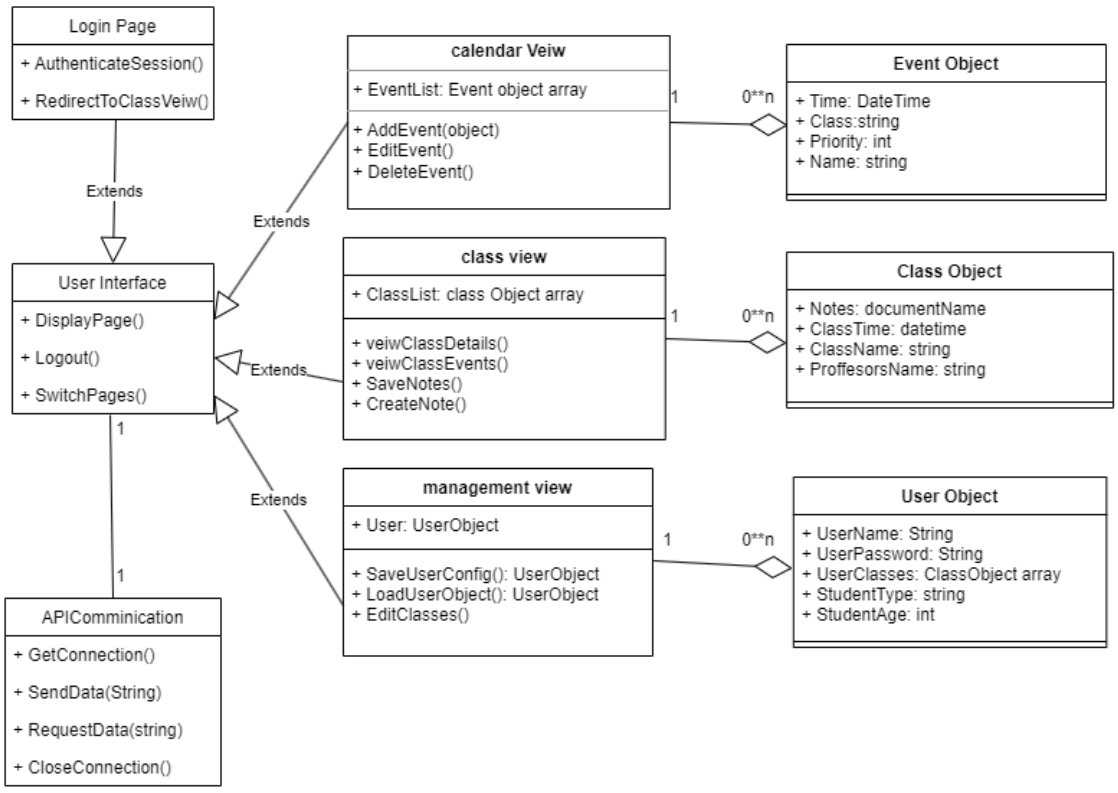

Figure 3:User Interface software Architecture

Property	Description
Name	Login Page
Description	Page that displays when app is opened on a device to authenticate users
Responsibilities	“landing” page for user, handles authentication
Relations	Extends the user interface
Methods	AuthenticateSession()-Sends a request to the Data broker to authenticate that session RedirectToClassVeiw()- after user is authenticated, it redirects to the class view
Attributes	None

Property	Description
Name	User Interface
Description	Encompasses all elements that the user will directly interact with
Responsibilities	Responsible interacting with the user
Relations	Association with API Communication module to get and receive data from the Data Broker
Methods	DisplayPage()-Displays elements required for the given page Logout()-Initiates ending the session SwitchPages()-switches from the given page to the user desired selected page
Attributes	None

Property	Description
Name	API Communication
Description	Talks to the data broker to send and receive user data
Responsibilities	communication with the data broker
Relations	Association with User Interface module to get and receive data from the Data Broker
Methods	GetConnection() - Opens a connection to the data broker SendData(string) – Sends data to Data Broker containing current session information RequestData(string)- Sends a data request to the data broker CloseConnection- ends connection with the data broker
Attributes	None

Property	Description
Name	Calander Veiw
Description	Handles managing the calendar page in the UI
Responsibilities	Manage calendar for the user
Relations	Extends User interface, in composed of events
Methods	AddEvent()-adds events to the calendar EditEvent()-change existing events in the calendar DeleteEvent()-remove an event from the calendar
Attributes	EventList- List of event objects

Property	Description
Name	EventObject
Description	Holds Data for a given event
Responsibilities	Holding data of an even
Relations	Is a component of calendar veiw
Methods	None
Attributes	Time-the time of the event Class- the class the event is for Priority-the importance of the event to the user

	Name-the name of the event

Property	Description
Name	Class Veiw
Description	Responsible for displaying class data and notes to the user
Responsibilities	Display nots and allow the user to add tasks to a class
Relations	Extends user interface, composed of class objects
Methods	VeiwClassDetails()-display details about the class VeiwClassEvents()-display events part of the class SaveNotes() sync the current version of the notes with the data broker CreateNote() create a new note for a class
Attributes	ClassList- A list of class objects that apply to the user

Property	Description
Name	Class Object
Description	Stores class data
Responsibilities	Hold data about a given class
Relations	Is a component of class view
Methods	None
Attributes	Notes-contains the name of the document in the database Classname-the name of the class ClassTime-the time the class is scheduled to meet ProffesorsName: the name of the professor teaching the class

Property	Description
Name	Management Veiw
Description	Handles the user interaction with management aspects of the app
Responsibilities	Allow the user to enter and change parameters for their use for the app
Relations	Extends user interface, composed of user objects
Methods	SaveUserConfig()-saves the current fields to the database LoadUserObject()-loads the current user parameters into the fields EditClasses-alows the user to add, edit, and remove classes from their profile
Attributes	User- UserObject with user parameters

Property	Description
Name	UserObject
Description	Holds data about the user
Responsibilities	Track data about the user
Relations	Component of management veiw
Methods	None
Attributes	UserName- username for the user UserPassword- the users password for their account UserClasses- array of class objects for a given user StudentType-the type of student using the app Student age- the age of the student

6. Interface Description

The user interface talks to a data broker that handles saving and retrieving data from the data broker. The data broker and User interface talk via an API hosted in Microsoft azure cloud. The Data Broker talks to the database over jdbc connections that will scale with user loads, the database is hosted in also hosted in Microsoft azure cloud

7. Size and Performance

SOS architecture is designed to meet size and performance goals set in the Software Requirements Specifications and the Supplementary Specifications Requirements. Those requirements were identified to offer quick performance while minimizing disk storage size.

8. Quality

SOS architecture is designed to meet quality standards set forth in the Software Requirements Specifications and the Supplementary Specifications Requirements.