Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2023

Bits Please
OurScene

Harrison Cawood
Bryan Cox
Sanny Tesfay
Vincent Nguyen
Daniel Palma
Shawn Hyder

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	11.29.2023	HC,SH,VN,ST,BC,DP	document creation

CONTENTS

List of Figures

List of Tables

1 Introduction

The digitalization of industries has been transforming the ways we engage with various forms of content, and the music industry is no exception. OurScene is poised to become a pivotal online platform within the local music ecosystem. This platform offers a comprehensive solution for the discovery, promotion, and organization of local music events. By serving as the nexus between venues, artists, promoters, and fans, OurScene provides a seamless, user-friendly experience that enriches the local music scene for all stakeholders involved.

Product Concept

OurScene is an innovative online platform designed to cater to the needs of the local music industry. Its primary aim is to streamline the process of organizing and attending music events, managing bookings, and enjoying local music. The platform's intuitive design and robust features are tailored to facilitate interactions among venues, promoters, artists, and fans, ensuring that each group can accomplish their objectives effectively and efficiently within a single ecosystem.

Scope

The scope of OurScene includes providing users with the ability to manage and track local music events, handle ticketing and bookings, discover new music, and engage with the local music community. The platform extends beyond mere event listing and ticket sales by incorporating features like music uploads, personalized radio, and interactive maps to offer a richer experience and better support the growth of local artists and venues.

Key Requirements

To fulfill its purpose, OurScene must meet the following key requirements:

1. **Shared Calendar System**: Venues and artists need a reliable and real-time system to display and manage their availability for events.

2. **Booking and Ticket Management**: An integrated, secure, and efficient ticketing system that streamlines the process of sales, purchases, and entry management is essential.

3. **Music Uploads**: Artists should have the ability to upload their music easily, allowing for sharing and discovery by fans and promoters.

4. **Personalized Radio and Playlists**: Fans should be provided with tools to create and enjoy personalized listening experiences, encouraging the exploration of new local talent.

5. **Interactive Event Map**: Integration with mapping services like Google Maps is crucial for event discovery and logistics planning.

6. **Key Performance Tracking**: For promoters and venues, tracking event performance through data analytics will support strategic planning and marketing efforts.

7. **Musical Promotion**: The platform should facilitate promotional opportunities for artists, including options for enhanced visibility through in-app purchases.

By addressing these requirements, OurScene will offer a unique and engaging platform that supports the local music industry and fosters a vibrant community of music lovers. The architectural design to follow will provide a roadmap for realizing a platform that not only meets but exceeds these foundational goals.

2 System Overview

For the basic first iteration of the subsystems the x layer alone is for the users to allow themselves to express themselves. With that being said it will allow users to create a page dedicated to them so that they can not only express themselves but also give a way to be contacted if need be.

The Z layer is more towards the data storage format for Ourscene. Here is where all the crutial functions such as ticket tracking, venue booking, etc. will take place.

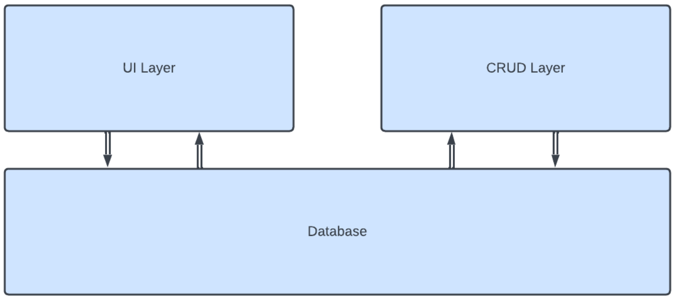

Figure 1: Architectural diagram of all subsystems

2.1 Layer X Description

This subsystem allows users to create their own page that contains information of the user. It also allows users to view other userśprofiles and connect with each other.

2.2 Layer Y Description

The Y layer of OurScene is the CRUD layer, this layer will be responsible for the transfer of CRUD calls between all the necessary components of the site. It will assist in retrieving values from the Z layer as well.

2.3 Layer Z Description

The Z layer of OurScene is the Database Layer, this layer will hold, receive and share all of the data associated with the application and will be interacted with by the API Layer and displayed in the UI or functions in the UI. The ticketing system will require database CRUD calls constantly each user has a unique relationship for how the data will be transferred Venues one way to the ticketing system promoters will send and receive data, bands will send data to the system and the user will both send and receive data. The ticketing system will send data into the KPI system. Which itself is just based on data from many subsystems.

3 Subsystem Definitions & Data Flow

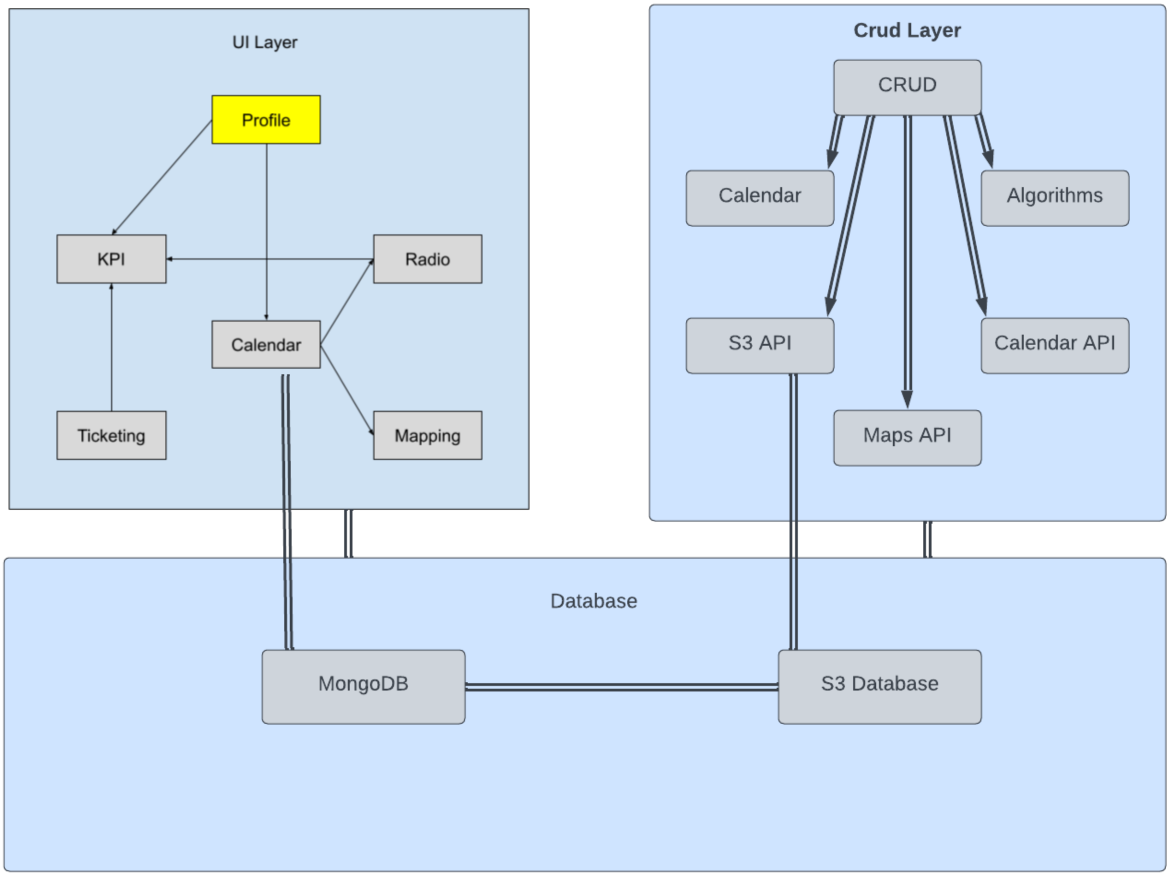

Figure 2: data flow diagram of OurScene

4 UI Layer (X Layer) Subsystems

4.1 Subsystem 1: Profile

This subsystem allows users to create their own page that contains information of the user. It also allows users to view other usersâ profiles and connect with each other.

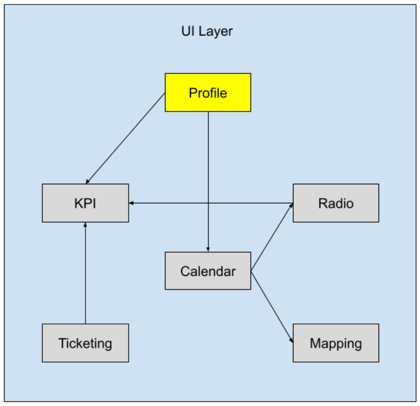

Figure 3: UI Subsystem with profile emphasized

4.1.1 Assumptions

- The user has already created an account on OurScene.
- It is determined whether the user is a promoter, vendor, band, or regular user.

4.1.2 Responsibilities

Profiles should contain information of users; such as their role (promoter, vendor, band, regular user), activity history (events held/attended), and contact information. They serve to allow other users to interact with one another to plan events.

4.1.3 Subsystem Interfaces

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#311	Promoter Profiles contains different information	Calendar KPI Ticketing	N/A
#312	Vendor Profiles contains different information	Calendar KPI	N/A
#313	DBand Profiles contains different information	Calendar KPI Radio	N/A
#314	Regular User Profiles contains different information	Radio Mapping	N/A

4.2 Subsystem 2: Calendar

The calendar allows users to view, arrange, or book upcoming events.

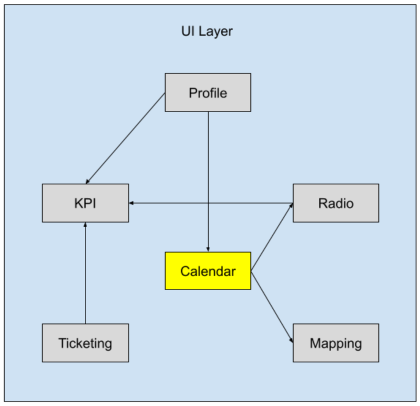

Figure 4: UI Subsystem with Calendar emphasized

4.2.1 Assumptions

- Data of the calendar is provided by Apple/Google calendar.

4.2.2 Responsibilities

The calendar should allow features to different types of users. Vendors should be able add venues that are available at specific dates to the calendar. Promoters and Bands should be able to view available venues from the calendar and can contact the vendor. Users should be able to view and book upcoming events on the calendar.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#321	Promoters can view and contact venues	KPI	Mapping Ticketing
#322	Vendors can add available venues	Profile KPI Mapping	N/A
#323	Bands can view and contact venues	Profile KPI Radio	Mapping
#324	Regular User can view and book upcoming events	Radio	Mapping

4.3 Subsystem 3: KPI

Key Performance Indicator (KPI) provides a measurement of performance over time for objectives. This allows vendors, promoters, and bands to view their progress and help them choose their next steps.

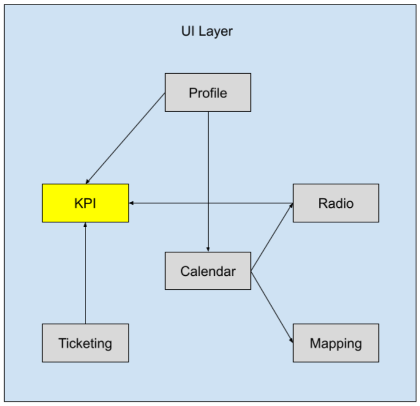

Figure 5: UI Subsystem with KPI emphasized

4.3.1 Assumptions

- Usersâ activities are tracked to provide the measurements for the KPI.

4.3.2 Responsibilities

The KPI should be adjusted accordingly and is dependent on other usersâ activities. It should keep track of usersâ ticket purchases, profile views, radio activity, etc. This data should be available to view by vendors, promoters, and bands for insight.

4.3.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#331	Promoters can view their KPI	Profile Ticketing	N/A
#332	Vendors can view their KPI	Profile Ticketing	N/A
#333	Bands can view their KPI	Profile Radio Ticketing	N/A
#334	Regular User contributes to the KPI	Profile // Radio // Ticketing	N/A

4.4 Subsystem 4: Radio

This subsystem allows users to explore and listen to bands and artists.

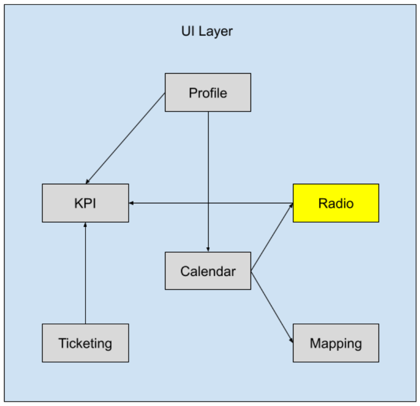

Figure 6: UI Subsystem with Radio emphasized

4.4.1 Assumptions

- The radio uses Spotify to play music.
- The user has a Spotify account.
- There is an algorithm for the public radio

4.4.2 Responsibilities

The radio allows users to explore new bands and artists as well as get familiarized with bands/artists holding concerts that users are planning to attend. There will be two radios users can access: a public radio and a personalized radio. Bands and promoters are able to promote their music on the public radio, allowing listeners to tune in. The personalized radio allows the listeners to choose who they would like to listen to.

4.4.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#341	Promoters can promote their band/artistâs music	Profile	Mapping Ticketing KPI
#342	Vendors can tune into the bands/artists through the radio	N/A	N/A
#343	Bands can promote their music on the radio	Profile	KPI
#344	Regular User can explore and listen to music from the radio	N/A	Profile Mapping Ticketing KPI

4.5 Subsystem 5: Ticketing

This subsystem allows the booking of concerts to take place, as well as the exchange of information of the events.

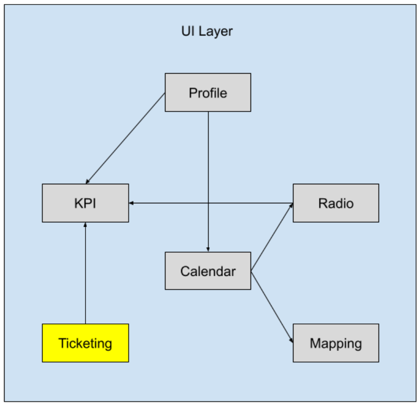

Figure 7: UI Subsystem with Ticketing emphasized

4.5.1 Assumptions

- The venue of which the event is being held is booked
- Users are able to pay through the website

4.5.2 Responsibilities

Promoters are able to set up a ticketing system that would allow users to book to the events. Users should be able to pay through the app to buy the ticket.

4.5.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#351	Promoters can set up a ticketing system	Profile Calendar Mapping	KPI
#352	Vendors can view and contribute to the system	Profile Mapping Calendar	KPI
#353	Bands can view and contribute to the system	Profile Calendar	KPI
#354	Regular User can buy tickets through the system	N/A	Profile Mapping KPI

4.6 Subsystem 6: Mapping

This subsystem provides the location information of where the event is being held.

4.6.1 Assumptions

- Data from the map is provided by Google Maps API

4.6.2 Responsibilities

Mapping allows all users to view locations of local venues and events. Promoters can view the locations of venues and also contact them. Regular users can view current or upcoming events happening near them..

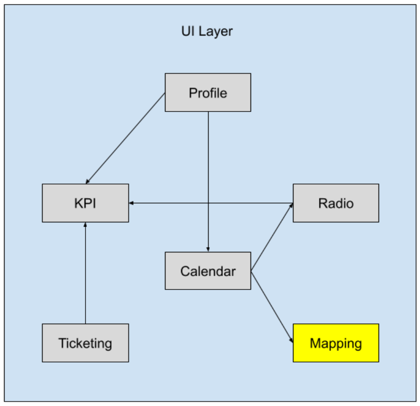

Figure 8: UI Subsystem with Mapping emphasized

4.6.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#361	Promoters can view locations of venues	Profile KPI	Calendar Ticketing
#362	Vendors can have their venues viewable in mapping	Profile Calendar KPI	N/A
#363	Bands can view locations of venues	Profile KPI Radio	Calendar
#364	Regular User can view locations of the events and venues locally	N/A	Calendar

5 Y Layer Subsystems

5.1 CRUD

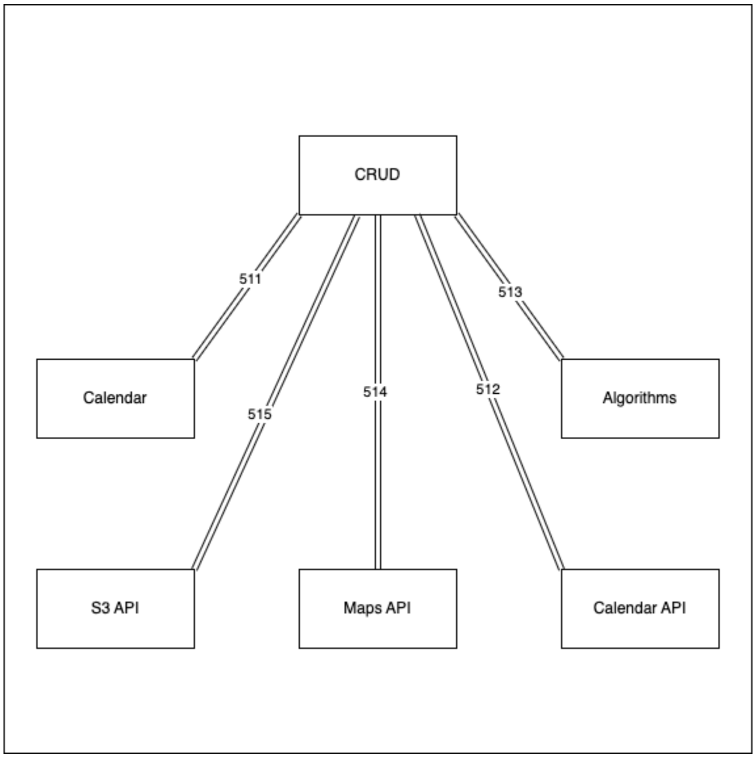

Figure 9: Example subsystem description diagram

5.1.1 Assumptions

Assumptions for CRUD include that users already have accounts with Our Scene as well as any other API that CRUD would interact with.

5.1.2 Responsibilities

CRUD is responsible for creation, reading, updating, and deleting data. It will handle requests from the front-end and return data to be displayed from the back-end as well as any back-end changes.

5.1.3 Subsystem Interfaces

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#511	Data handling	Profile	Profile
#512	Data handling	Calendar API	Calendar API
#513	Data handling	Algorithms	Algorithms
#514	Data handling	Map API	Map API
#515	Data handling	S3 API	S3 API
#516	Data handling	Ticketing	Ticketing
#517	Data handling	Radio	Radio

5.2 Algorithms

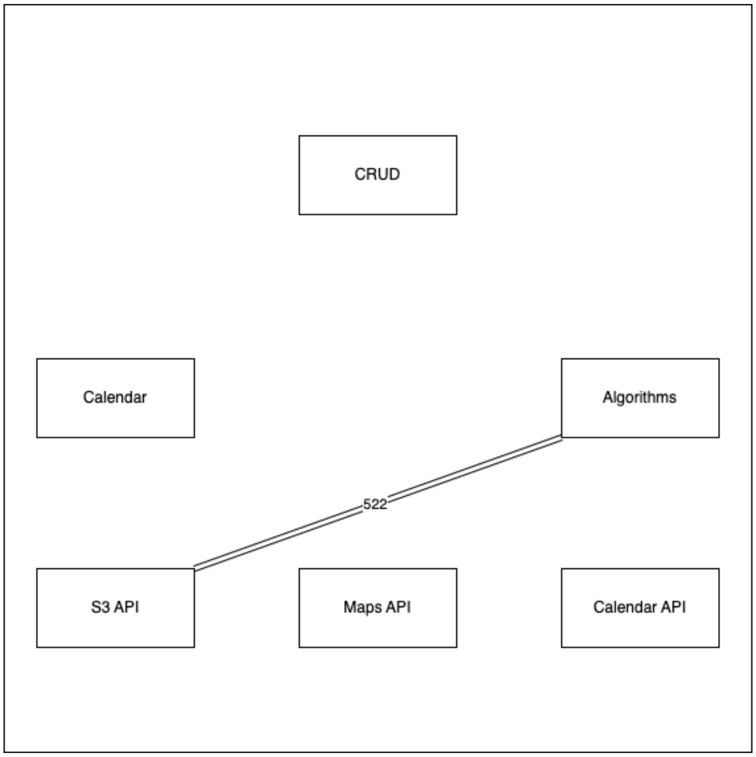

Figure 10: Example subsystem description diagram

5.2.1 Assumptions

Assumptions for the algorithm part of the Y layer are that users utilizing the radio will all be listening to the same song in real time and will have appropriate data speeds to be able to stream the music.

5.2.2 Responsibilities

Algorithm will be responsible for deciding what song to play next, and requesting the song

5.2.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#521	Output to radio	Radio	Radio
#522	Request to S3 API for Radio	S3 API	S3 API

5.3 S3 API

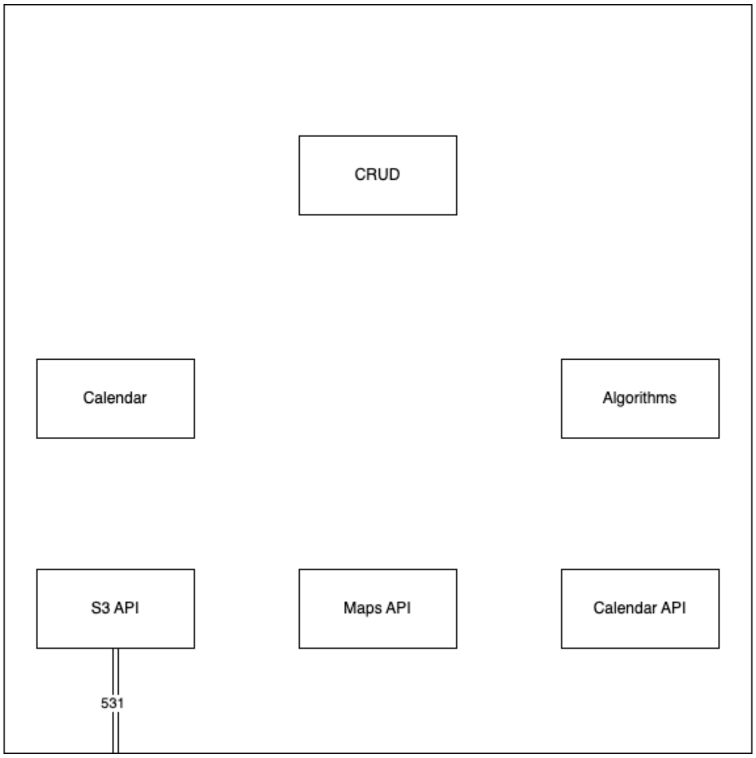

Figure 11: Example subsystem description diagram

5.3.1 Assumptions

The only assumption for the S3 API subsystem is that there will always be adequate funding to be able to sustain its use.

5.3.2 Responsibilities

S3 API is responsible for requesting stored music data and audio files. S3 is high speed which is what is needed to be able to provide a radio service.

5.3.3 Subsystem Interfaces

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#531	Interface for S3 database	Algorithm S3	Algorithm S3

5.4 Maps API

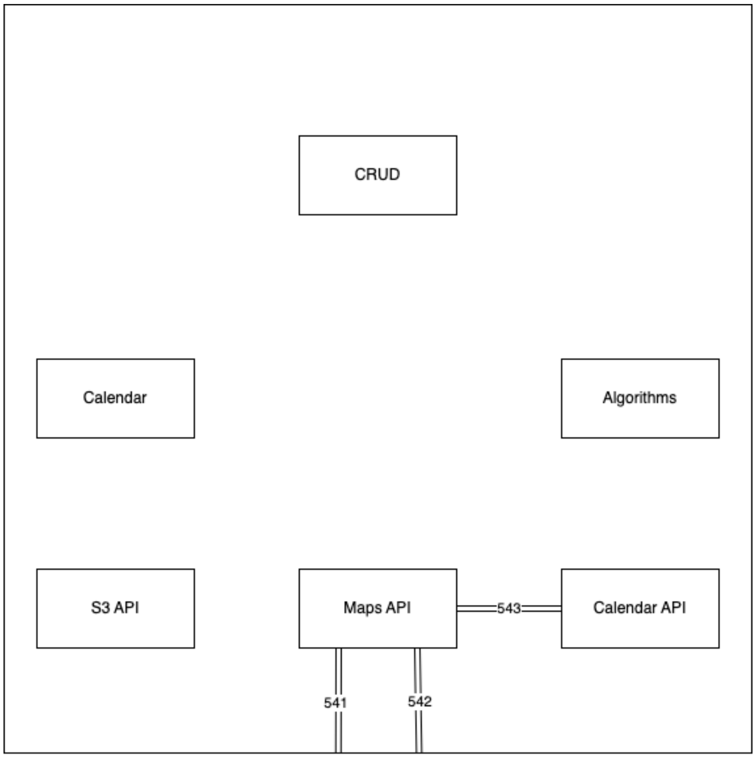

Figure 12: Example subsystem description diagram

5.4.1 Assumptions

Assumptions for the maps API subsystem are that users will already have accounts with third party API and are able to access the appropriate services to be able to utilize the map API subsystem.

5.4.2 Responsibilities

Map API will be responsible for gathering GPS data, making requests to appropriate third party API, and displaying directions

5.4.3 Subsystem Interfaces

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#541	Map data from Google	Google Maps API	Mapping Calendar
#542	Map data from Apple	Apple Maps API	Mapping Calendar
#543	Map Data to Calendar	Google Maps API Apple Maps API	Calendar

5.5 Calendar API

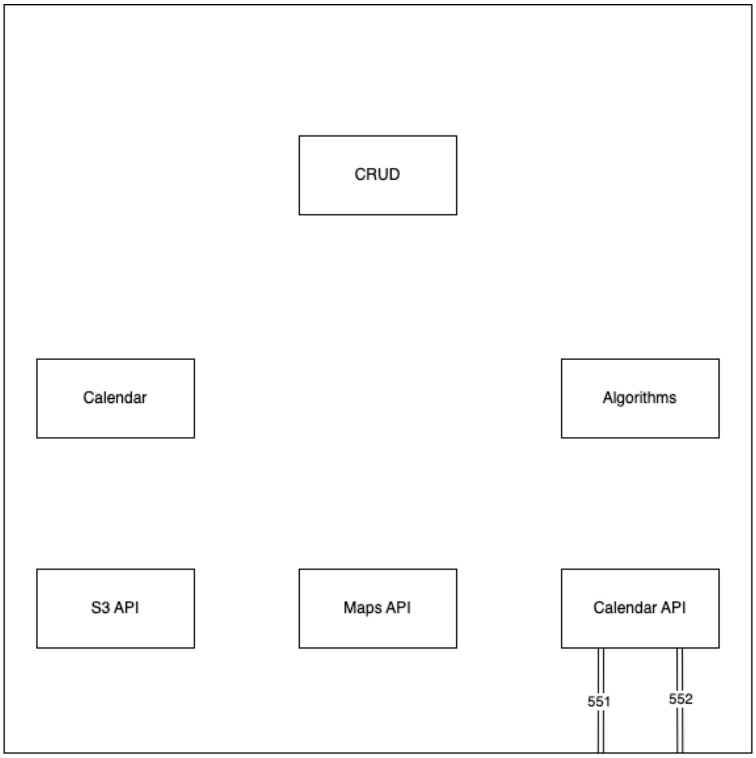

Figure 13: Example subsystem description diagram

5.5.1 Assumptions

Assumptions for the calendar API are that users already have accounts and are able to access the appropriate API to be able to utilize the Calendar subsystem.

5.5.2 Responsibilities

The calendar API subsystem is responsible for storing and displaying calendar data for all Our Scene users.

5.5.3 Subsystem Interfaces

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#551	Dates for events	Calendar Google Calendar API	Calendar Google Calendar API
#552	Dates for events	Calendar Apple Calendar API	Calendar Apple Calendar API

6 Z Layer Subsystems

6.1 CRUD Calls

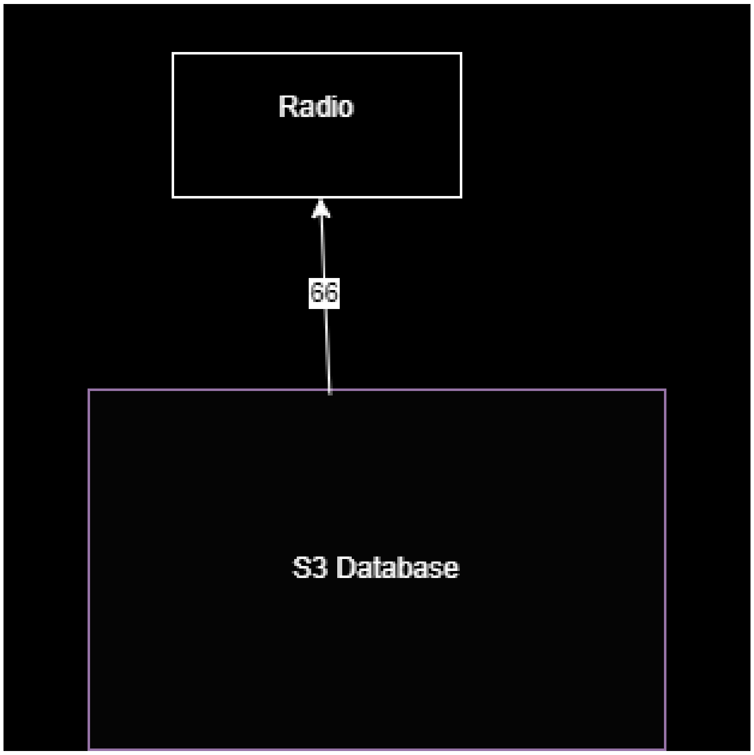

Figure 14: Figure showing connection between CRUD and radio

6.1.1 Assumptions

Assumptions for CRUD Calls would be that there are proper APIs implemented to take data from this layer and affect all of the systems

6.1.2 Responsibilities

The functions will be CRUD calls on the database by each system. Information from the database layer will be created, read, updated, and deleted by the API layers and that info will be shown on the front end to varying degrees on the different services that are affected by the DB. The CRUD calls will affect each section of the app. For KPIs, crud calls will be needed to get and create data to be displayed for the various KPI users. The profile will have data needed to be used to flow to every other service of the application. The calendar system will need data back and forth to update the calendar system with gigs for bands, open nights for promoters, and open nights for venues. The ticketing service will need to use and update the database and the radio will be in need of data from many services in order for the algorithm to play what is coming near the user.

6.1.3 Subsystem Interfaces

Table 13: Subsystem interfaces

ID	Description	Inputs	Outputs
#61	Ticketing CRUD Calls	User info, show info	ticket info, KPI metrics, calendar event, radio stats
#62	KPI CRUD Calls	Ticket info, user info, radio data	KPI Metrics for each user type
#63	Profile CRUD Calls	User data	User data, and metrics
#64	Radio CRUD Calls	show info, user info, map info, calendar information, and	KPI Metrics, algorithm changes
#65	Calendar CRUD Calls	ticketing info, user info	Mapping information, radio information and algo changes

6.2 Amazon S3

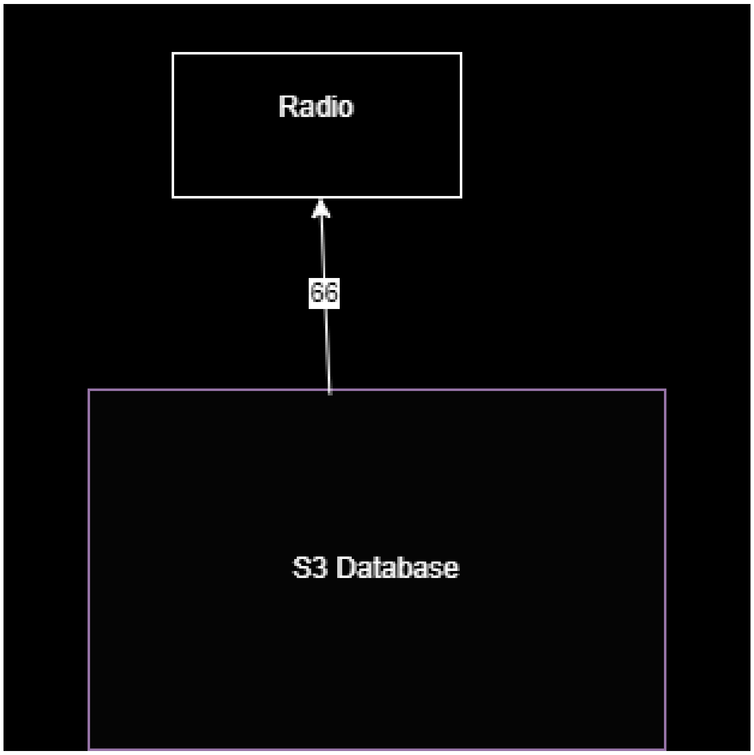

Figure 15: Figure showing connection of CRUD to radio

6.2.1 Assumptions

Assumptions for S3 calls would be that the algorithm uses crud calls on our MongoDB database and will be given what calls to make

6.2.2 Responsibilities

The functions of the S3 calls will just be for music only as S3 is optimized for multimedia. These calls will affect the radio and all profile types

6.2.3 Subsystem Interfaces

Table 14: Subsystem interfaces

ID	Description	Inputs	Outputs
#66	Radio Calls to S3 Database	User info, Show info, Radio info	Music for artists pages allowing for all users to listen