# Bilkent University
# Department of Computer Engineering

## Senior Design Project

*T2306*

*Vybe*

## Detailed Design Report

*21902570, Mehmet Berk Türkçapar, berkturkcapar@gmail.com*
*21901784, Yiğit Ekin, ygtekn01@gmail.com*
*21901658, Can Önal, can.onal@ug.bilkent.edu.tr*
*21903452, Harun Can Surav, haruncan.surav@gmail.com*
*21903088, Oğuz Ata Çal, ogz.ata1@gmail.com*

*Cevdet Aykanat*
*Erhan Dolak*
*Tağmaç Topal*

**12.03.2023**

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

## Contents

# 1 Introduction

## 1.1 Purpose of the System

As the COVID-19 pandemic is coming to an end with the help of vaccines, people are returning back to their normal lives. Restaurants and bars are getting more crowded and people are eager to end their long-lasting starvation of socializing. People are going to these venues not only for food or drinks, but also for the atmosphere these places provide. The most significant factor which determines the quality and style of this atmosphere are the songs played in these venues. According to a study, when music that matches the concept of a venue is played, a 9% increase has been observed in the sales of a restaurant [1]. Thus, background music is a crucial factor which enhances the experience of customers. However, customers usually do not have a word on which music is played at a venue. Therefore, users are stuck with the queue.

Back in the 1950s jukeboxes were very popular amongst bars and restaurants and they solved this issue by enabling users to play songs they wish by inserting coins [2]. However, nowadays jukeboxes have become history. There is not a similar medium in which people are able to request and play songs in the venues they are in. This problem is what our application will solve. Vybe will offer a similar experience for customers by creating a platform in which users will be able to request tracks from a predefined playlist of the venue. It will enable a new way of interaction for customers and create a unique experience for them. Also, businesses will be able to offer a more engaging and dynamic environment which will attract new customers. For these reasons, Vybe will transform the whole experience of going to bars, restaurants or cafes by enabling them to become vivid and alive with the help of its customers.

## 1.2 Design Goals

### 1.2.1 Performance

- The application and all systems should start in under 1 second.
- Adding a new song to the music queue should take less than 2 seconds.
- Looking up to friend-ed users on the in-app map should take less than 1 second.
- The latency between the music playing and its representation in the mobile app should be less than 2 seconds.
- Coin transfers should get executed within 1 second of transactions.

### 1.2.2 Usability

- User-friendly UI components will be used such as understandable short texts, big buttons with ripple effect and easy-to-find navigators.
- A predefined color palette will be used to help distinguish the primary and secondary components in the user interface.
- The application will be responsive with regards to the dimension of the device. Also, it will support landscape and portrait modes.

- The application will use the native built-in component according to the operating system that the application runs in. This will make the user use the components that he/she is familiar with.

### 1.2.3 Security

- User credentials should be stored encrypted.
- Cash transfers should be executed in a secure manner.
- Location information should be encrypted.
- 2FA with SMS verification will be used for restraining malicious users.
- Location-based verification will be used to prevent malicious users from crowding the track queue of the venue.

## 1.3 Overview

In its core, Vybe will be an application designed to liven up and bring character to venues by bringing interactivity to music playing and listening. Instead of listening to the same static playlists everyday, users will be able to control the music and dynamically affect the mood of the venue by adding their preferred tracks through a queue system.

Vybe will consist of a web application for venue owners and a mobile application for other users. After scanning the QR code placed on the venues, users will be able to check-in to the session of that venue. After checking in users will have the option to see the music that is currently playing, see the current queue and request tracks to be put on the queue. All users will be given an amount of free requests, decided by the venue, and the subsequent requests will be paid with some amount going to the venue owners. By making this interactivity convenient and easy to use, we aim to bring the venue and its customers closer together.

The web application will serve the venue owners. It will be the platform where the venue owners and designated personnel will be able to control the music by adding tracks and predefined playlists to play when there are no requests. Venue owners will also be able to create restricted playlists for customers to choose their requests from using genres or other parameters. The system will also have the option of generating playlists based on the previous user requests.

The mobile application will also contain social media features designed to increase customer engagement and provide additional features to attract customers to venues. Users will be able to see the venues on a map, inspect the venues, see details about the music played and see the rating/comments regarding the venue’s music. Users will also be able to earn badges and keep track of other statistics in their profile page according to various activities like going to the same venue for consecutive days. These achievements will also be visible to the user's friends. We believe that these features will incentivize users to use our app more and as a result we hope to bring more business to the venues.

## 2 Current Software Architecture

Currently in Turkey, to stream music most of the venues use either Spotify or Youtube. However, these applications are not licensed for commercial use. There are legal alternatives such as Soundtrack your Brand and Touchtunes.

Soundtrack your Brand is one legal way venues can stream commercial music. The application provides the use of playlists, queues and other means to control the music. However, it doesn't have the interactivity and social aspect that Vybe presents.

Another alternative is Touchtunes. It provides both a legal way to play music and the ability for the customers to interact with the music. Yet, it requires venues to purchase additional hardware in order to use their services.

The advantage that Vybe brings to the table against its alternatives is that it has no obligations regarding hardware. A computer is enough for a workplace to stream music with the help of Vybe. In addition, this data will be collected and processed in order to give customers an overview about the music played in a place and other social media features that will give the customers a better experience while also benefiting the venue by attracting more customers.

## 3 Proposed Software Architecture

### 3.1 Overview

Vybe is a complex and big system composed of many external APIs, services and database tables. In order to meet our standards, we have decided to model isolated subsystems that communicate with each other. The subsystems decomposition and how the system interacts with hardware components are explained in more detail in the following two sections below with diagrams and textual descriptions.

The last two sections serve the purpose of explaining more of the logical details of the backend system. Namely, the details of how live information is mapped to persistent storage and the precautions and techniques implemented in order to make a secure application.

## 3.2 Subsystem Decomposition

Below lies the subsystem decomposition diagram of Vybe application.

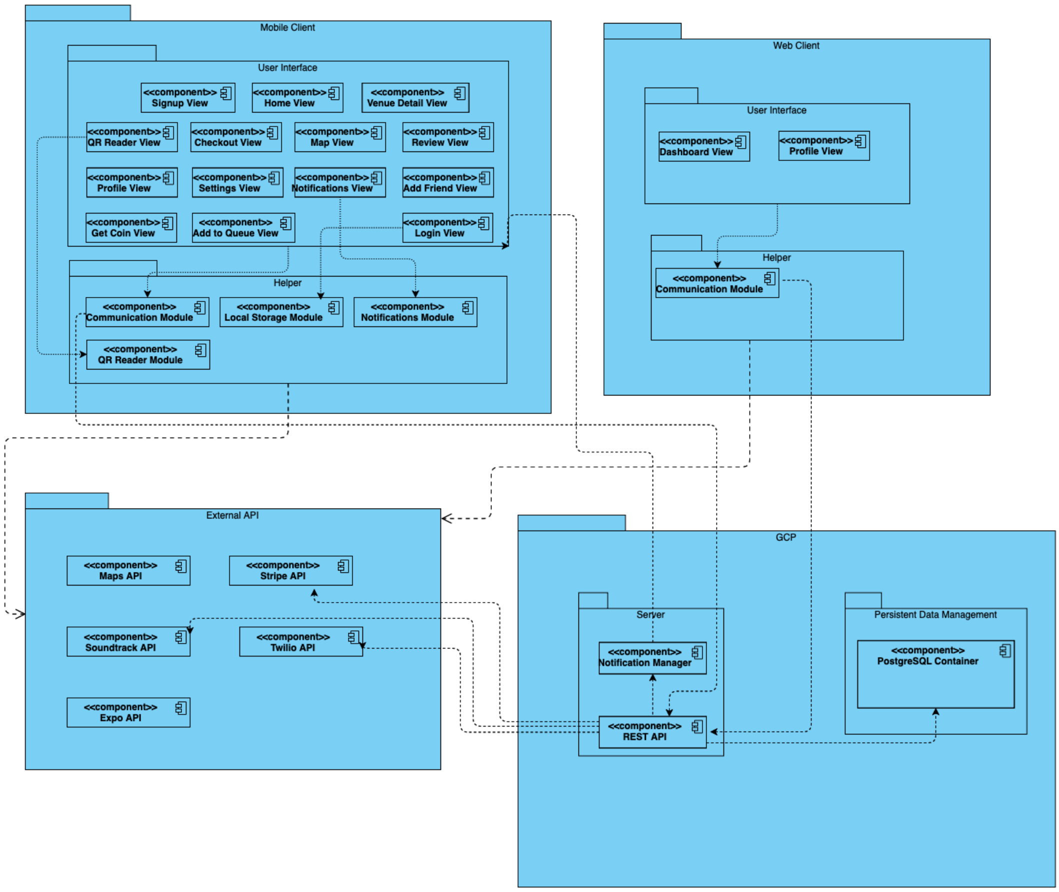

Mobile Client
User Interface
<<component>> Signup View
<<component>> Home View
<<component>> Venue Detail View
<<component>> QR Reader View
<<component>> Checkout View
<<component>> Map View
<<component>> Review View
<<component>> Profile View
<<component>> Settings View
<<component>> Notifications View
<<component>> Add Friend View
<<component>> Get Coin View
<<component>> Add to Queue View
<<component>> Login View
Helper
<<component>> Communication Module
<<component>> Local Storage Module
<<component>> Notifications Module
<<component>> QR Reader Module
Web Client
User Interface
<<component>> Dashboard View
<<component>> Profile View
Helper
<<component>> Communication Module
External API
<<component>> Maps API
<<component>> Stripe API
<<component>> Soundtrack API
<<component>> Twilio API
<<component>> Expo API
GCP
Server
<<component>> Notification Manager
<<component>> REST API
Persistent Data Management
<<component>> PostgreSQL Container

Figure 1 - Subsystem Decomposition Diagram

We decided to build our application on four main components: Mobile Client, Web Client, External APIs and Google Cloud Platform. Mobile Client, which will be used by venue customers, consists of UI and helper modules, responsible for on-device storage management, notifications and QR reading . Web Client will be used by venue admins to see annual statistics. Both clients use communication modules to interact with the backend side.

Backend is hosted by Google Cloud Platform (GCP) and has three components. REST API contains the main backend module written in Java. It also serves as the entry point for the backend side. Notification manager is responsible for directing notifications to different devices. Docker instance of PostgreSQL does persistent data management.

Both GCP and clients use external API's. Maps API is used in mobile client, Expo API is an essential React development API used in mobile client. Twilio is used in the REST API module and is responsible for Two Factor Authentication. Stripe API is the payment manager. SoundTrack API helps us direct music to the venue.

### 3.3 Hardware/Software Mapping

Below is the deployment diagram of the app.

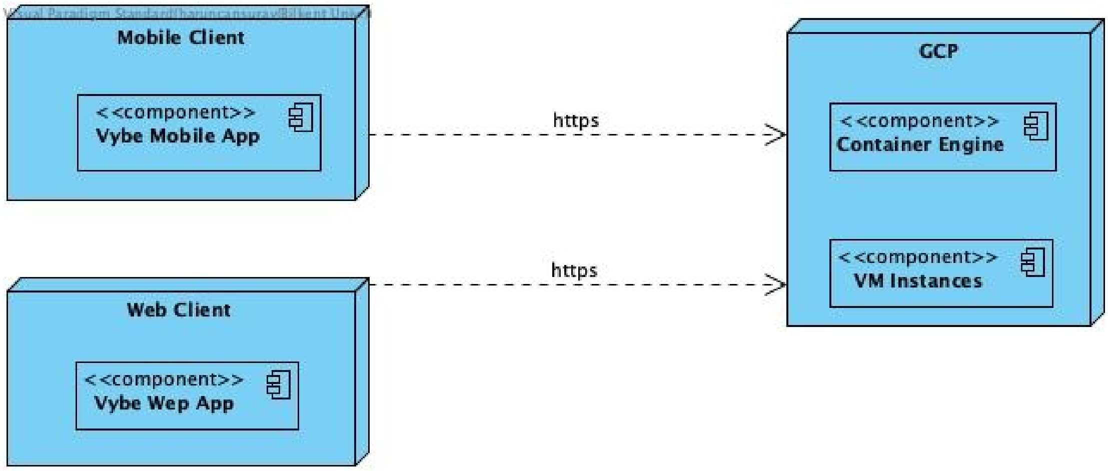


Figure 2 - Deployment Diagram

There are two clients in our system. One is the mobile client that is running on all mobile devices. The other is the web client that is running on the web. Both of the client. Both of the clients render the page and most of the business logic is on the cloud on Google Cloud Platform (GCP). The clients use http requests over the web to the cloud server. This communication will be facilitated using the REST api of the backend.

The cloud server has the container engine and the VM instances. The backend and the database system will be dockerized and the container will be run on the server. For persistent database systems we use a PostgreSQL image.

### 3.4 Persistent Data Management

In Vybe, a Dockerized instance of PostgreSQL will be used. The Docker instance will be containerized and will be maintained within Google Cloud Platform (as seen in Subsystem Decomposition). As our backend application is written in Java SpringBoot, it makes sense to use an SQL database. PostgreSQL has been chosen by the backend team as it has been used before, eliminating the learning curve.

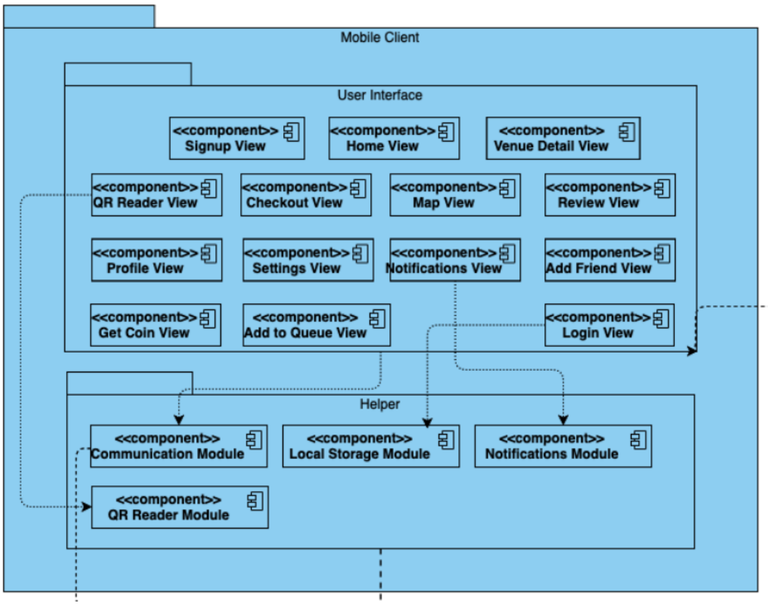
User data (profile information, achievements, venue visits…) and venue data (pictures, scores, analytics…) will be stored in this database. To further increase user privacy, methods that ensure K-Anonymity and Geo-Indistinguishability will be implemented. The Dockerized database instance will be managed through Google Cloud Platform and the REST API that runs on Java SpringBoot framework.

### 3.5 Access Control and Security

Communication between clients and GCP servers will use JSON Web Tokens (JWT) to ensure that the user is authorized to access the pages. Confidential data will remain on servers to prevent on-device security breaches. All confidential data is SHA256 encrypted and any non-encrypted versions of any password data are not stored on the database or anywhere in the system. System authorization keys (API keys, database username & keys…) are shared between team members through encrypted channels. During login & registration, password and user data will be encrypted on-device to prevent man in the middle attacks. To protect our users geolocation data, a k-anonymity level of at least two is present for every user at every restaurant.

# 4 Subsystem Services

## 4.1 Mobile Client



Mobile Client
User Interface
<<component>> Signup View
<<component>> Home View
<<component>> Venue Detail View
<<component>> QR Reader View
<<component>> Checkout View
<<component>> Map View
<<component>> Review View
<<component>> Profile View
<<component>> Settings View
<<component>> Notifications View
<<component>> Add Friend View
<<component>> Get Coin View
<<component>> Add to Queue View
<<component>> Login View
Helper
<<component>> Communication Module
<<component>> Local Storage Module
<<component>> Notifications Module
<<component>> QR Reader Module

Figure 2 - Mobile Client

### User Interface

- **Signup View:** This view contains the sign up screen.
- **QR Reader View:** This view contains the qr reader screen where the native camera module is presented.
- **Checkout View:** This view contains a pop up component which asks the user if he/she is sure of checking out of the venue.
- **Map View:** This view contains the google maps screen where each venue is pointed out in the map.
- **Add Friend View:** This view contains the add friend screen where a list of users are presented with the option to add as friend.
- **Notifications View:** This view contains the notification list made by the system.
- **Login View:** This view contains the login screen where the user is asked to enter their username and password.
- **Add to Queue View:** This view enables the user to search for music and send it as a song request. Also, the user should select the type of request (default or enhanced) and the amount of coin to be sent in the case of an enhanced request.

- **Review View:** This view enables the user to rate and add comments about the venue.
- **Venue Detail View:** This view enables the user to see the most played songs and genres in the venue and contains the profile picture and the background picture of the venue.
- **Home View:** This view displays where the user's friends are currently checked in and the current song at the venue.
- **Settings View:** This view enables the user to change their password and profile picture.
- **Profile View:** This view displays information about the user.
- **Get Coin View:** This view enables the user to acquire coins by either watching ads or directly buying with credit card.

### Helper

- **Communication Module:** This module enables communication with the backend. HTTP request and JSON data will be transferred.
- **Local Storage Module:** User data will be stored for later retrieval.
- **Notifications Module:** Push and local notifications will be sent to the client.
- **QR Reader Module:** This module enables the client to read the QR code in venues and data in the QR Code will be read.

## 4.2 Web Client

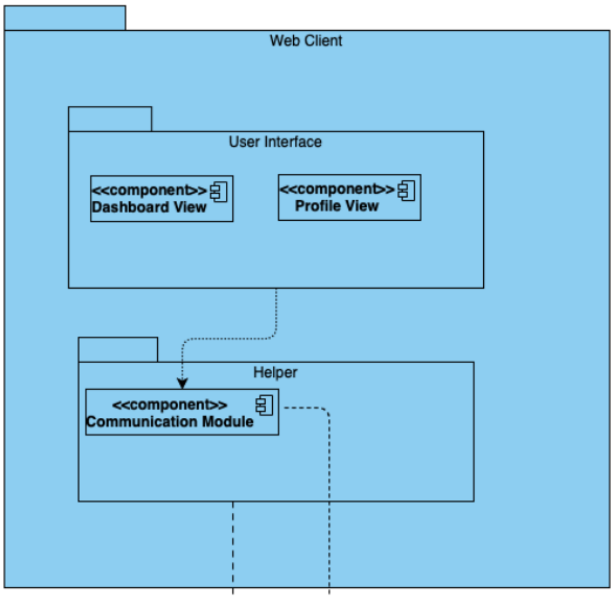


Figure 4 - Web Client

### User Interface

- **Dashboard View:** In this view, the business can see their Vybe analysis. This analysis includes the most listened genre or song in a certain period of time (daily, monthly).

- **Profile View:** In this view the business can edit their information such as profile picture and description.

### Helper

- **Communication Module:** This module enables communication with the backend. HTTP request and JSON data will be transferred.

## 4.3 External API

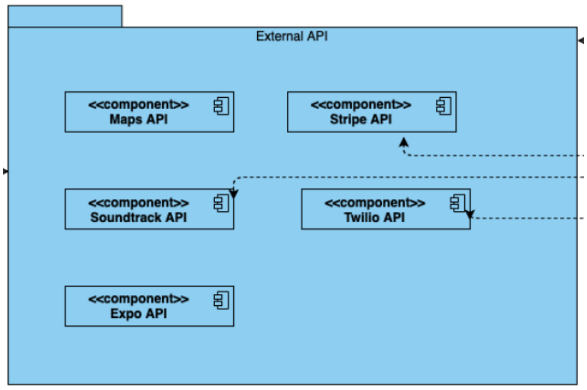


Figure 5 - External API

- **Google Maps API:** A map view and location data will be retrieved.
- **Soundtrack API:** Song requests and currently playing songs will be transferred between client and this API.
- **Stripe API:** Payments will be collected via this API.
- **Twilio API:** 2FA authentications will be done with this API.
- **Expo API:** This API enables developing native apps.
- **Google Ads API:** This API provides the ads.

## 4.4 Google Cloud Platform (GCP)

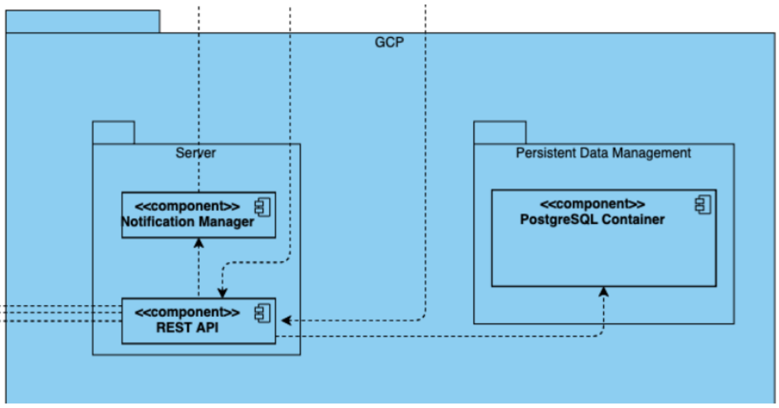

GCP
Server
<<component>> Notification Manager
<<component>> REST API
Persistent Data Management
<<component>> PostgreSQL Container

Figure 6 - GCP

- **Notification Manager:** This module delivers the notifications it retrieves from the REST API to the phones.

- **REST API:** This API enables communication to the backend.
- **PostgreSQL Container:** This is the running instance of the database on the cloud server.

## 5 Test Cases

Test Case ID: 1
Test Category: Functional
Test Title: Create a new account with valid information
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a valid phone number.
4. Click on the "Sign-up" button.
5. Enter two-factor authentication (2FA) code sent to the phone number correctly.
6. Click on the "Continue" button.
7. Enter a valid email address.
8. Click on the "Continue" button.
9. Enter a valid password to the first text input field.
10. Enter the same password to the next input field.
11. Click on the "Continue" button.
12. Enter a valid name.
13. Click on the "Continue" button.
14. Verify that the user is successfully registered and redirected to the Vybe home screen with status "Not Currently Checked In".
15. Verify that the user's account information is displayed correctly in My Profile page.

Expected Result: The user should be able to successfully create a new account with valid information and access the Vybe home screen with their account information displayed correctly.

Severity: Critical

Test Case ID: 2
Test Category: Security
Test Title: Create a new account with an already existing phone number
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a phone number that is already registered in the system in the required field.
4. Click on the "Sign-up" button.

5. Verify that an error message is displayed stating that the phone number is already existing in the system.

Expected Result: An error message should be displayed if the user enters an already existing phone number, and the user should not be able to create an account until they provide a non-existent and valid phone number.

Severity: Critical

Test Case ID: 3
Test Category: Security
Test Title: Create a new account with entering the 2FA code incorrectly
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a valid phone number.
4. Click on the "Sign-up" button.
5. Enter two-factor authentication (2FA) code sent to the phone number correctly.
6. Click on the "Continue" button.
7. Verify that an error message is displayed stating that the 2FA code is entered incorrectly.

Expected Result: An error message should be displayed if the user enters the 2FA code incorrectly, and the user should not be able to create an account until they enter the correct 2FA code.

Severity: Critical

Test Case ID: 4
Test Category: Security
Test Title: Create a new account with already registered email address
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a valid phone number.
4. Click on the "Sign-up" button.
5. Enter two-factor authentication (2FA) code sent to the phone number correctly.
6. Click on the "Continue" button.
7. Enter a valid email address that is already registered in the system in the required field.
8. Click on the "Continue" button.
9. Verify that an error message is displayed stating that the email address is already registered in the system.

Expected Result: An error message should be displayed if the user enters an email address that is already registered in the system, and the user should not be able to create an account until they provide a unique email address.

Severity: Critical

Test Case ID: 5
Test Category: Security
Test Title: Create a new account with passwords that do not match
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a valid phone number.
4. Click on the "Sign-up" button.
5. Enter two-factor authentication (2FA) code sent to the phone number correctly.
6. Click on the "Continue" button.
7. Enter a valid email address.
8. Click on the "Continue" button.
9. Enter a valid password to the first text input field.
10. Enter the same password to the next input field.
11. Click on the "Continue" button.
12. Verify that an error message is displayed stating that the passwords do not match.

Expected Result: An error message should be displayed if the user enters two different passwords in the password fields, and the user should not be able to create an account until they provide matching passwords.

Severity: Critical

Test Case ID: 6
Test Category: Functional
Test Title: Login with valid email and password
Test Procedure:

1. Open the Vybe app.
2. Enter a valid email and password in the login fields.
3. Click on the "Login" button.
4. Enter the 2FA code sent to the phone number registered in the system.
5. Click on the "Submit" button.
6. Verify that the user is successfully logged in and redirected to the Vybe home screen with status "Not Currently Checked In".

Expected Result: The user should be able to successfully log in with valid email, password, and 2FA code and access the Vybe home screen.

Severity: Critical

Test Case ID: 7
Test Category: Security
Test Title: Login with invalid email
Test Procedure:

1. Open the Vybe app.
2. Enter an invalid email (e.g. non-existent email, wrong format) in the email field.
3. Enter a valid password in the password field.
4. Click on the "Login" button.
5. Verify that an error message is displayed stating that the email is invalid.

Expected Result: An error message should be displayed if the user enters an invalid email, and the user should not be able to log in until they provide a valid email.

Severity: Major

Test Case ID: 8
Test Category: Functional
Test Title: Login with invalid password
Test Procedure:

1. Open the Vybe app.
2. Enter a valid email in the email field.
3. Enter an invalid password in the password field (e.g. wrong password).
4. Click on the "Login" button.
5. Verify that an error message is displayed stating that the password is invalid.

Expected Result: An error message should be displayed if the user enters an invalid password, and the user should not be able to log in until they provide a valid password.

Severity: Major

Test Case ID: 9
Test Category: Security
Test Title: Login with incorrect 2FA code
Test Procedure:

1. Open the Vybe app.
2. Enter a valid email and password in the login fields.
3. Click on the "Login" button.

4. Enter an incorrect 2FA code sent to the phone number registered in the system.
5. Click on the "Submit" button.
6. Verify that an error message is displayed stating that the 2FA code is incorrect.

Expected Result: An error message should be displayed if the user enters an incorrect 2FA code, and the user should not be able to log in until they provide a valid 2FA code.

Severity: Major

Test Case ID: 10
Test Category: Usability
Test Title: Open rear camera for QR reader
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the QR-reader button.
3. Verify that the rear camera successfully opens.

Expected Result: The rear camera should successfully open when the user clicks on the QR-reader button.

Severity: Major

Test Case ID: 11
Test Category: Usability
Test Title: Align QR code with outline
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the QR-reader button.
3. Align a valid QR code with the outline displayed on the screen.
4. Verify that the QR code is read successfully.

Expected Result: The QR code should be aligned correctly with the outline and the QR code should be read successfully.

Severity: Major

Test Case ID: 12
Test Category: Functional
Test Title: Check-in with valid QR code and location
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the QR-reader button.

3. Align a valid QR code with the outline displayed on the screen.
4. Make sure that the user's location is within the region of the venue.
5. Verify that the user is successfully checked-in to the venue and the status is updated as "Checked-In".

Expected Result: The user should be able to successfully check-in to the venue with a valid QR code and location, and the status should be updated as "Checked-In".

Severity: Critical

Test Case ID: 13
Test Category: Security
Test Title: Check-in with invalid QR code
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the QR-reader button.
3. Align an invalid QR code with the outline displayed on the screen.
4. Verify that an error message is displayed stating that the QR code is invalid.

Expected Result: An error message should be displayed if the user tries to check-in with an invalid QR code, and the user should not be able to check-in until they provide a valid QR code.

Severity: Major

Test Case ID: 14
Test Category: Security
Test Title: Check-in with invalid location
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the QR-reader button.
3. Align a valid QR code with the outline displayed on the screen.
4. Verify that the user's location is outside the region of the venue.
5. Verify that an error message is displayed stating that the user is not within the region of the venue.

Expected Result: An error message should be displayed if the user tries to check-in with an invalid location, and the user should not be able to check-in until they are within the region of the venue.

Severity: Major

Test Case ID: 15

Test Category: Component
Test Title: Clicking "Make Song Request" button
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Check-in to a venue with a playlist.
3. Click on the "Make Song Request" button.

Expected Result: The "Make Song Request" button should be clickable and should open a search bar to search for a song.

Severity: Major

Test Case ID: 16
Test Category: Component
Test Title: Selecting a song request
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Check-in to a venue with a playlist.
3. Click on the "Make Song Request" button.
4. Type the name of a song in the search bar.
5. Click on the song that the user would like to request.

Expected Result: A popover should appear with options to select the type of song request.

Severity: Major

Test Case ID: 17
Test Category: Functional
Test Title: Song Request with Default Request
Test Procedure:

1. Login to the Vybe app with valid credentials and check in at a venue.
2. Click on the "Make Song Request" button.
3. Search for a song that is in the playlist of the venue.
4. Click on the song and select "Default Request" from the popover.
5. Verify that the user's current balance is checked to make sure they have enough coins for the request.
6. Verify that the user's coin balance is updated accordingly.
7. Verify that the song is added to the end of the current queue.

Expected Result: The song should be added to the queue if the user makes a Default Request and has enough coin balance.

Severity: Critical

Test Case ID: 18
Test Category: Functional
Test Title: Song Request with Enhanced Request
Test Procedure:

1. Login to the Vybe app with valid credentials and check in at a venue.
2. Click on the "Make Song Request" button.
3. Search for a song that is in the playlist of the venue.
4. Click on the song and select "Enhanced Request" from the popover.
5. Select the amount of coin the user wants to use for the request.
6. Verify that the user's current balance is checked to make sure they have enough coins for the request.
7. Verify that the user's coin balance is updated accordingly.
8. Verify that the song is added to the correct position at the queue according to the amount of coins selected.

Expected Result: The song should be added to the queue in the correct position if the user makes an Enhanced Request and has enough coin balance.

Severity: Critical

Test Case ID: 19
Test Category: Usability
Test Title: Song Request Search Bar Functionality
Test Procedure:

1. Login to the Vybe app with valid credentials and check in at a venue.
2. Click on the "Make Song Request" button.
3. Verify that the user is able to search for a song by typing the name of the song to the search bar.
4. Verify that only the songs which are in the playlist of the venue are displayed in the search results.
5. Verify that the user is able to click on the song they would like to request.

Expected Result: The search bar should function properly and only display songs that are in the playlist of the venue.

Severity: Major

Test Case ID: 20
Test Category: Security
Test Title: Insufficient Coin Balance

Test Procedure:

1. Login to the Vybe app with valid credentials and check in at a venue.
2. Click on the "Make Song Request" button.
3. Search for a song that is in the playlist of the venue.
4. Click on the song and select "Enhanced Request" from the popover.
5. Select an amount of coin that exceeds the user's current coin balance.
6. Verify that the user is notified with an error message stating that they do not have sufficient coin balance.
7. Verify that the song is not added to the queue.

Expected Result: The user should be notified with an error message if they try to make an Enhanced Request with an insufficient coin balance and the song should not be added to the queue.

Severity: Critical

Test Case ID: 21
Test Category: Usability
Test Title: Searching for a non-existing song
Test Procedure:

1. Login to the Vybe app with valid credentials and check in at a venue.
2. Click on the "Make Song Request" button.
3. Verify that the user is able to search for a song by typing the name of the song to the search bar.
4. Type the name of a song which doesn’t exist in the venue’s playlist

Expected Result: This search should return an empty list and no song should be displayed on screen.

Severity: Critical

Test Case ID: 22
Test Category: Functional
Test Title: Check out successfully
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the "Checkout" button.
3. Click on the "Yes" button from the popover asking if the user is sure to check out.
4. Verify that the user is redirected to the home page.
5. Verify that the user status is updated to "Not Checked In".

Expected Result: The user should be able to check out successfully by clicking on the "Checkout" button, and the user status should be updated to "Not Checked In".

Severity: Critical

Test Case ID: 23
Test Category: Usability
Test Title: Cancel checkout
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the "Checkout" button.
3. Click on the "I changed my mind." button from the popover asking if the user is sure to check out.
4. Verify that the popover disappears.
5. Verify that the user status is still "Checked In".

Expected Result: The user should be able to cancel the checkout process by clicking on the "I changed my mind." button, and the user status should remain as "Checked In".

Severity: Minor

Test Case ID: 24
Test Category: Security
Test Title: Checkout when out of venue region
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Check-in to a venue.
3. Move outside of the venue region.
4. Verify that the user is automatically checked out.
5. Verify that the user is notified.
6. Verify that the user is redirected to the home page.
7. Verify that the user status is updated to "Not Checked In".

Expected Result: The user should be automatically checked out and notified when they are out of the venue region, and the user status should be updated to "Not Checked In".

Severity: Critical

Test Case ID: 25
Test Category:  Integration
Test Title: View Venues on Map
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Verify that the map is displayed and that the user's location and venues around them are marked on the map.

Expected Result: The user should be able to view the map with their location and the venues around them marked.

Severity: Major

Test Case ID: 26
Test Category: Functional
Test Title: View Venue Details
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Click on a venue on the map.
4. Verify that the details of the selected venue are displayed on the screen.

Expected Result: The user should be able to view the details of a selected venue.

Severity: Critical

Test Case ID: 27
Test Category: Functional
Test Title: Make Venue Review
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Click on a venue on the map.
4. Scroll down to the bottom of the page and click on the "Make a Review" button.
5. Use the slider to rate the venue out of 5.
6. Type in a comment (optional).
7. Click on the "Submit" button.
8. Verify that a notification is displayed confirming that the review is saved and that the user is returned to the venue details page.

Expected Result: The user should be able to rate a venue out of 5 using a slider and leave a comment, and the review should be saved and confirmed with a notification.

Severity: Major

Test Case ID: 28

Test Category: Usability
Test Title: Search Venues
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Click on the search icon located at the top of the screen.
4. Type in a search query for a venue.
5. Verify that the search results are displayed.

Expected Result: The user should be able to search for a venue and see relevant search results.

Severity: Minor

Test Case ID: 29
Test Category: Functional
Test Title: Venue Rating Error Handling
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Click on a venue on the map.
4. Scroll down to the bottom of the page and click on the "Make a Review" button.
5. Use the slider to rate the venue out of 5.
6. Type in a comment (optional).
7. Click on the "Submit" button.
8. Verify that an error message is displayed if there was an issue saving the review, and that the user is returned to the review page.

Expected Result: The user should be notified if there was an error saving their review, and the error message should be displayed.

Severity: Critical

Test Case ID: 30
Test Category: Usability
Test Title: Review submission without rating
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon on the bottom menu.
3. Click on a venue from the map to go to the venue details page.
4. Click on the "Make a Review" button located at the bottom of the screen.

5. Leave the rating slider at the default position (0).
6. Optionally, type a comment in the text box.
7. Click on the "Submit" button.

Expected Result: A notification should be displayed indicating that a rating is required to submit a review. The review should not be submitted and the user should still be on the same page.

Severity: Major

Test Case ID: 31
Test Category: Functional
Test Title: Display Currently Playing Song at Venue Details Page
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located on the left side of the bottom menu.
3. Select a venue from the map.
4. Verify that the "Currently Playing" section is displayed on the venue details page.
5. Check the currently playing song on the venue's sound system.
6. Verify that the currently playing song displayed on the "Currently Playing" section matches the actual song being played in the venue.

Expected Result: The currently playing song displayed on the "Currently Playing" section of the venue details page should match the actual song being played in the venue.

Severity: Major

Test Case ID: 32
Test Category: Functional
Test Title: Search User
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user

Expected Result: The user should be forwarded to the profile page of the user

Severity: Major

Test Case ID: 33
Test Category: Functional
Test Title: Search for Non Existing User
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a non existing username
4. see an error on the screen

Expected Result: an empty array will be returned from the server which will result in a no users found text at the screen.

Severity: Major

Test Case ID: 34
Test Category: Functional
Test Title: Add Friend
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user
5. Click on the Send Friend Request Button

Expected Result: a friend request will be send to the related user and a notification will be send to their logged in device.

Severity: Critical

Test Case ID: 35
Test Category: Component
Test Title: Add Friend Button Text
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user
5. See a Send Friend Request Button

Expected Result: as the user is not a friend, a send friend request button should be seen.

Severity: Major

Test Case ID: 36
Test Category: Component
Test Title: Remove Friend
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user
5. Click on the Remove Friend Button

Expected Result: as the user is a friend, the related user should be removed from the friends list.

Severity: Critical

Test Case ID: 37
Test Category: Component
Test Title: Remove Friend Button Text
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user
5. See a Remove Friend Button

Expected Result: as the user is a friend, a send friend request button should be seen.

Severity: Major

Test Case ID: 38
Test Category: Integration
Test Title: Forwarding of user’s other social media’s
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on the icon at the top right corner
3. Search for a username
4. Click on the the profile tab of the user
5. Click on the Instagram, Facebook or Twitter Logo
6. Directed to the specific platform

Expected Result: The user should be directed to the necessary registered social media of the user.

Severity: Major

Test Case ID: 39
Test Category: Functional

Test Title: Edit First Name
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on edit first name
4. Enter a valid first name
5. Change the first name

Expected Result: The user should be able to see their changed first name

Severity: Major

Test Case ID: 40
Test Category: Functional
Test Title: Edit First Name Error Handling For Empty Input
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on edit first name
4. Enter an empty first name
5. See an error on the screen

Expected Result: The user should be able to see an pop up of invalid input type error

Severity: Critical

Test Case ID: 41
Test Category: Functional
Test Title: Edit Last Name
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on edit last name
4. Enter a valid last name
5. Change the last name

Expected Result: The user should be able to see their changed last name

Severity: Major

Test Case ID: 42
Test Category: Functional

Test Title: Edit Last Name Error Handling For Empty Input
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on edit last name
4. Enter an empty last name
5. See an error on the screen

Expected Result: The user should be able to see an pop up of invalid input type error

Severity: Critical

Test Case ID: 43
Test Category: Functional, Integration
Test Title: Edit Profile Picture with Device Album
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on change profile picture
4. Select from device
5. Select a photo from gallery (.jpg, .jpeg or .png)
6. Change the profile picture

Expected Result: The user should be able to see their changed profile picture

Severity: Major

Test Case ID: 44
Test Category: Functional, Integration
Test Title: Edit Profile Picture with Files
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on change profile picture
4. Select upload a document (.jpg, .jpeg or .png)
5. Select a files
6. Change the profile picture

Expected Result: The user should be able to see their changed profile picture

Severity: Major

Test Case ID: 45
Test Category: Functional
Test Title: Change Profile Error Handling For Unsupported Image Formats
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on change profile picture
4. Select upload an unsupported file (none .jpg, .jpeg or .png) document
5. See an error

Expected Result: The user should be able to see an pop up of invalid image type error

Severity: Critical

Test Case ID: 46
Test Category: Usability
Test Title: Change in Font Sizes for Different Device Sizes for IOS
Test Procedure:

1. Iterate over all of the pages of the application with Iphone SE, Iphone 11, Iphone 14, Iphone 14 Pro Max
2. See whether the font sizes have changed according to the device dimensions

Expected Result: The fonts should be usable related to the device size

Severity: Major

Test Case ID: 47
Test Category: Usability
Test Title: Change in Font Sizes for Different Device Sizes for Android
Test Procedure:

1. Iterate over all of the pages of the application with Samsung Galaxy S23, Google Pixel 7 and Asus Zenfone 9
2. See whether the font sizes have changed according to the device dimensions

Expected Result: The fonts should be usable related to the device size

Severity: Major

Test Case ID: 48
Test Category: Usability
Test Title: Change in Component Sizes for Different Device Sizes for IOS
Test Procedure:

1. Iterate over all of the pages of the application with Iphone SE, Iphone 11, Iphone 14, Iphone 14 Pro Max
2. See whether the padding, margin and component sizes have changed according to the device dimensions

Expected Result: The component should be usable related to the device size

Severity: Major

Test Case ID: 49
Test Category: Usability
Test Title: Change in Component Sizes for Different Device Sizes for Android
Test Procedure:

1. Iterate over all of the pages of the application with Samsung Galaxy S23, Google Pixel 7 and Asus Zenfone 9
2. See whether the padding, margin and component sizes have changed according to the device dimensions

Expected Result: The component should be usable related to the device size

Severity: Major

Test Case ID: 50
Test Category: Documentation
Test Title: Launching Client and Server with cross teams
Test Procedure:

1. Make frontend team launch the server with the documentation written by the backend team
2. Make backend team launch the client with the documentation written by the frontend team

Expected Result: Both client and server should be able to launched with the documentation provided by subteams

Severity: Critical

Test Case ID: 51
Test Category: Functional
Test Title: Accept Friend Request
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right

3. Click on Notification button
4. See possible friend requests
5. Click on accept button for the intended friend request
6. Accept friend request

Expected Result: The user should be able add the user to their friends list.

Severity: Critical

Test Case ID: 52
Test Category: Functional
Test Title: Decline Friend Request
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on profile button at the bottom right
3. Click on Notification button
4. See possible friend requests
5. Click on Decline button for the intended friend request
6. Accept friend request

Expected Result: The user should be able decline the request and not see the notification anymore

Severity: Critical

Test Case ID: 53
Test Category: Component
Test Title: Bottom Drawer Navigation Test
Test Procedure:

1. Login to the Vybe app with valid credentials
2. Click on buttons at the bottom drawer
3. Navigate across pages

Expected Result: The user should be able navigate between pages

Severity: Critical

Test Case ID: 54
Test Category: Component
Test Title: Back to Previous Page Navigation Test
Test Procedure:

1. Login to the Vybe app with valid credentials

2. Click on profile page
3. Click on the back arrow button at the top left corner
4. navigate to the page at the top of the stack

Expected Result: The user should be able navigate the home page

Severity: Critical

Test Case ID: 55
Test Category: Component
Test Title: Error Handling for Non Existing Venue Query on Map
Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the location icon located at the bottom left side of the screen.
3. Click on the search icon located at the top of the screen.
4. Type in a search query for a non existing venue.
5. See an empty map on the screen with an error text

Expected Result: The user should be able to see an empty map with no venues found error text on it

Test Case ID: 56
Test Category: Component, Security
Test Title: Resend One Time Password Button
Test Procedure:

1. Open the Vybe app.
2. Click on the "Create Account" button.
3. Enter a valid phone number.
4. Click on the "Sign-up" button.
5. Click resend code button

Expected Result: The user should be able get another code to their same number through SMS.

Severity: Minor

Test Case ID: 57

Test Category: Functional

Test Title: Verify user can access notifications from the profile page

Test Procedure:

1. Launch the Vybe app and login with valid credentials
2. Click on the profile button on the bottom drawer
3. Click on the notifications button

Expected Result: User should be able to access the notifications page from the profile page. Verify that the notifications page is displayed with all the relevant information.

Severity: Critical

Test Case ID: 58

Test Category: Functional

Test Title: Verify user cannot access notifications without logging in

Test Procedure:

1. Launch the Vybe app
2. Click on the profile button on the bottom drawer
3. Click on the notifications button

Expected Result: User should not be able to access the notifications page without logging in. Verify that the app prompts the user to log in before displaying the notifications page.
Severity: Critical

Test Case ID: 59

Test Category: Usability

Test Title: Verify user can navigate back to profile page from notifications page

Test Procedure:

1. Launch the Vybe app and login with valid credentials
2. Click on the profile button on the bottom drawer
3. Click on the notifications button
4. Click on the back button on the top left corner of the notifications page

Expected Result: User should be able to navigate back to the profile page from the notifications page by clicking on the back button. Verify that the profile page is displayed with all the relevant information.

Severity: Major

Test Case ID: 60

Test Category: Security

Test Title: Verify user cannot access other user's notifications

Test Procedure:

1. Launch the Vybe app and login with valid credentials
2. Click on the profile button on the bottom drawer
3. Click on the notifications button
4. Logout from the app
5. Login with another user's credentials
6. Click on the profile button on the bottom drawer
7. Click on the notifications button

Expected Result: User should not be able to access other user's notifications. Verify that the app displays an error message indicating that the user is not authorized to access the notifications.

Severity: Critical

Test Case ID: 61

Test Category: Component

Test Title: Profile Page Coin Balance Test

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Look for the coin balance on the profile page.

Expected Result: The user should be able to log in successfully with valid credentials, and after clicking on the profile page, the user should be navigated to the profile page where the coin balance is visible. This is a critical functionality of the application and any failure to display the coin balance can significantly impact the user experience.

Severity: Critical

Test Case ID: 62

Test Category: Component

Test Title: Add Coins Button Test

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.

Expected Result: The user should be able to log in successfully with valid credentials, navigate to the profile page, and click on the "Add Coins" button to navigate to the add coins page.

Severity: Major

Test Case ID: 63

Test Category: Component

Test Title: Add Coins Page Test

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Check if the "Add Coins via Watching Ads" option is displayed.
7. Check if the "Add Coins via Credit Card" option is displayed.
8. Check if the "Transaction History" option is displayed.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, and be navigated to the add coins page where they should be able to see options to add coins via watching ads, credit card, and transaction history.

Severity: Major

Test Case ID: 64

Test Category: Functionality

Test Title: Add Coins via Watching Ads Test

Test Procedure:

1. Login to the Vybe app with valid credentials.

2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Click on the "Add Coins by Watching Ads" option.
7. Check if the user is navigated to the ads page.
8. Watch the ads until completion.
9. Check if the user's coin balance is updated.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, select the "Add Coins via Watching Ads" option, watch the ads until completion, and have their coin balance updated accordingly.

Severity: Critical

Test Case ID: 65

Test Category: Functionality

Test Title: Add Coins by Watching Ads - User Closes App

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Click on the "Add Coins by Watching Ads" option.
7. Check if the user is navigated to the ads page.
8. Close the app while the ads are playing.
9. Open the app again and check if the user's coin balance is not updated.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, select the "Add Coins by Watching Ads" option, close the app while the ads are playing, open the app again, and not have their coin balance updated.

Severity: Major

Test Case ID: 66

Test Category: Functional

Test Title: Add Coins via Watching Ads - User Exits Ad Before Completion

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Click on the "Add Coins via Watching Ads" option.
7. Check if the user is navigated to the ads page.
8. Exit the ad before it finishes playing.
9. Check if the user's coin balance is not updated.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, select the "Add Coins via Watching Ads" option, exit the ad before it finishes playing, and not have their coin balance updated.

Severity: Major

Test Case ID: 67

Test Category: Functional

Test Title: Add Coins via Credit Card - Payment Handling - Successful Payment

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Click on the "Add Coins via Credit Card" option.
7. Check if the user is navigated to the credit card payment page.
8. Enter valid credit card information and click on "Pay".
9. Check if the app shows a success message indicating that the payment is completed.
10. Check if the user's coin balance is updated and the correct amount of coins is added to the balance.
11. Check if the transaction history is updated and the payment details are recorded in the history.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, select the "Add Coins via Credit Card" option, and make a successful payment. If the payment is successful, the user's coin balance should be updated with the correct amount of coins, and an appropriate success message should be shown to the user. Additionally, the transaction history should be updated with the payment details.

Severity: Major

Test Case ID: 68

Test Category: Functional

Test Title: Add Coins via Credit Card - Payment Handling - Unsuccessful Payment

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Click on the profile page on the bottom drawer.
3. Check if the user is navigated to the profile page.
4. Click on the "Add Coins" button.
5. Check if the user is navigated to the add coins page.
6. Click on the "Add Coins via Credit Card" option.
7. Check if the user is navigated to the credit card payment page.
8. Enter invalid credit card information and click on "Pay".
9. Check if the app shows an error message indicating that the payment cannot be completed.
10. Check if the user's coin balance is not updated and remains the same.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the profile page, click on the "Add Coins" button, select the "Add Coins via Credit Card" option, and handle payment errors such as invalid credit card information. If the payment is not successful, the user's coin balance should not be updated, and an appropriate error message should be shown to the user.

Severity: Major

Test Case ID: 69

Test Category: Functionality - Transaction History

Test Title: Transaction History - Non-Empty History

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Check if the user is navigated to the home page.
3. Click on the profile page on the bottom drawer.
4. Click on the "Add Coins" button.
5. Check if the user can see the transaction history view on the add coins page.

Expected Result:

The user should be able to log in successfully with valid credentials, navigate to the add coins page, and check the transaction history view without clicking any buttons. If there are any transactions in the history, the app should display them correctly to the user. If the history is empty, the app should display a message saying there are no transactions to show.

Severity: Minor

Test Case ID: 70

Test Category: Functionality - Transaction History

Test Title: Transaction History - Empty History

Test Procedure:

1. Login to the Vybe app with valid credentials.
2. Check if the user is navigated to the home page.
3. Click on the profile page on the bottom drawer.
4. Click on the "Add Coins" button.
5. Check if the user can see the transaction history view on the add coins page.
6. If the history is empty, check if the app displays a message saying "No transactions to show" or similar.

Expected Result: The user should be able to log in successfully with valid credentials, navigate to the add coins page, and check the transaction history view without clicking any buttons. If there are no transactions in the history, the app should display a message saying there are no transactions to show.

Severity: Minor

Test Case ID: 71

Test Category: Usability

Test Title: Login Page Test-Vybe for Business Web App

Test Procedure:

1. Open the Vybe for Business web app
2. Enter valid login credentials
3. Click on the "Login" button

Expected Result: The user should be redirected to the home page of the dashboard

Severity: Critical

Test Case ID: 72

Test Category: Usability

Test Title: Login Error Handling Test-Vybe for Business Web App

Test Procedure:

1. Open the Vybe for Business web app
2. Enter invalid login credentials
3. Click on the "Login" button

Expected Result: The user should receive an error message indicating that the login credentials are incorrect and remain on the login page

Severity: Major

Test Case ID: 73

Test Category: Functional

Test Title: Home Page Test-Vybe for Business Web App

Test Procedure:

1. Login to the Vybe for Business web app
2. Verify that the home page displays weekly/daily/hourly analysis of the venue

Expected Result: The venue admin should be able to view detailed analytics of their venue's usage

Severity: Major

Test Case ID: 74

Test Category: Functional

Test Title: Profile Page Test-Vybe for Business Web App

Test Procedure:

1. Login to the Vybe for Business web app
2. Navigate to the "My Profile" page

Verify that only the venue photo can be changed and other information like venue name and email is read-only

Expected Result: The venue admin should not be able to edit certain information and only able to change the venue photo and password

Severity: Major

Test Case ID: 75

Test Category: Security

Test Title: Password Change Test-Vybe for Business Web App

Test Procedure:

1. Login to the Vybe for Business web app
2. Navigate to the "My Profile" page
3. Click on the "Change Password" button
4. Enter the current password and a new password
5. Click on the "Save" button
6. Log out and log back in with the new password

Expected Result: The venue admin should be able to change their password and successfully log in with the new password

Severity: Critical

Test Case ID: 76

Test Category: Security

Test Title: Unauthorized Access Test-Vybe for Business Web App

Test Procedure:

1. Attempt to access the Vybe for Business web app without valid login credentials

Expected Result: The user should not be able to access the dashboard and should be redirected to the login page

Severity: Critical

## 6 Consideration of Various Factors in Engineering Design

In this section the possible effects of various factors on the engineering design process of the project will be discussed in detail.

### 6.1 Public Health Considerations

Vybe is an application that provides interactivity between the attendants of the venue and the music that is playing in it. It has no additional health risk to the users of the application.

## 6.2 Public Safety Considerations

Because Vybe is only a phone application for the venue attendants and web application for the venue owners and provides no external hardware, it doesn't bear any safety risk to any users.

## 6.3 Public Welfare Considerations

Our target demographic in this application is people attending a restaurant, hotels and other similar venues who have a phone and want to interact with the music currently playing. As such, it is mainly a leisure focus app and it doesn't concern the public welfare.

## 6.4 Global Considerations

The core of our app, which is listening to and interacting with music, is a universal concept and is applicable in all countries and settings. Thus, Vybe is very extendable to other countries provided there is a translation or English is widely used in that country.

## 6.5 Cultural Considerations

Since our app operates as a mediator between the venue customer and the owner, it doesn't affect or make any assumptions about the cultural background of any of the users.

## 6.6 Social Considerations

First of all, for venue protection reasons the app uses the live location of the customer to check-out and restricts the user from reaching a venue if he moves far away from the venue. However, to respect the privacy of the user, the live location of the user is never kept in the system. Streaks, points, and other personal information of the user is also kept in a more protected way than other information.

Another social consideration is the music that is requested by the users. Because users can dictate the music playing in a public environment, it becomes a social issue to ensure that the users do not have complete freedom and play derogatory or discriminatory tracks.

## 6.7 Environmental Considerations

Vybe is an application that provides interactivity between the attendants of the venue and the music that is playing in it. It does not have any direct or indirect environmental effect to be considered.

## 6.8 Economic Considerations

Vybe is both a business to business and a business to consumer product. For businesses, it aims to provide a legal way of streaming music in their venues and an interactive and engaging environment for their customers. The legal public music streaming service Soundtrack Your Brand (30 € per month per business) which will be used for this purpose is the most important expense to be considered. In order to meet with the expenses this service will be provided to the businesses for a fixed fee. This fixed fee needs to be adjusted in a way that the business owners won’t be discouraged to utilize our product and it will cover the

Soundtrack Your Brand expenses. On the other hand, for consumers, the mobile app will provide a medium in which they will be able to see and get insight about businesses and interact with their friends. This mobile application will be free. However, users will need to obtain in app currency to make song requests in venues. This in app currency will be available via purchasing with credit card or watching ads. The pricing of this in app currency is also important. With in app purchases, advertisement income and the fixed fee for the venues the business will be the main source of income for the application.

| | Effect level (1-10) | Effect |
|---|---|---|
| Public Health | 1 | low effect |
| Public Safety | 1 | low effect |
| Public Welfare | 1 | low effect |
| Global Factors | 4 | increasing localization can increase the use of our app |
| Cultural Factors | 1 | low effect |
| Social Factors | 8 | Proper means of moderation both by the system and the venue owner should be implemented |
| Environmental Factors | 1 | low effect |
| Economic Factors | 8 | The cost analysis of the expenses should be done carefully and the pricing of the in-app currency along with the monthly fixed fee charged to the venues should be decided accordingly. |

Table 1 - Factors that can affect analysis and design

## 7 Teamwork Details

### 7.1 Contributing and Functioning Effectively on the Team

In order to make each member function effectively, we do several actions. First of all, we make sure that our group has a shared purpose where everyone is on the same page in terms of implementation details. This is achieved by weekly meetings. In each meeting, attendance is mandatory and we note the discussions that were conducted to prevent any forgetting of details or misinterpretation later on. Another important topic that we focus on is the role clarity. To achieve this, at the end of our weekly meetings, we partition the workload to the team and create the relevant github issues according to it. This allows us to prevent one team member doing the work of another. Finally, in terms of implementation, we try to implement the project with some regulations to allow each member to contribute effectively to the project. First regulation that we do is try to follow the clean code principles of this document [3]. Second regulation that we do is to use Prettier formatter for all files in the repository so that each code segment is of the same format. This has been embedded into each member's IDE with format on save option. These two regulations improve the overall readability and hence allows each member to contribute effectively. Finally, we do code reviews on each pull request in order to prevent any possible bugs for the future.

## 7.2 Helping Creating a Collaborative and Inclusive Environment

Creating a collaborative and inclusive environment is essential for generating a successful product. To achieve this, we use a combination of both synchronous and asynchronous communication. In our team, weekly meetings are conducted in terms of how we will proceed with the future tasks. If the topic of discussion requires a huge amount of peer to peer communication, then the meeting in person is conducted in person. However, if that is not the case, we conduct the meeting through our Vybe discord channel. We believe that unnecessary face to face meetings decrease the interest of the team regarding the project which as a result will decrease the inclusiveness of team members. Hence, we avoid that at all costs. Also, to increase the collaborative nature of meetings and create a chemistry in between, attendance is mandatory to these weekly meetings. We do not have a strict time zone in the week for these meetings and we try to find the best possible time slot that fits each member. Finally, pair programming is conducted while developing the project as team work will both increase the productivity as well as the efficiency in terms of time of the project.

## 7.3 Taking Lead Role and Sharing Leadership on the Team

In our team, we believe in equal distribution of leadership responsibilities. This means that each member is encouraged to take charge when the situation calls for it, and everyone is given an opportunity to lead. We understand that leadership is not just about giving orders, but also about taking responsibility and guiding the team towards a goal. As a result, each team member takes ownership of their tasks and works collaboratively to achieve success. This approach not only promotes a sense of unity and cohesion within the team, but also helps us to achieve our goals more efficiently and effectively. We have divided task leaderships as follows:

- Project Specification Report and Launch of the app is led by Can Önal
- Analysis Requirement and Final reports are led by Harun Can Surav
- Demos and Detailed Design Reports are led by Mehmet Berk Türkçapar
- Yigit Ekin leads the Frontend development processes.
- Oguz Ata Çal leads the Backend development processes.

# 8 References

[1] D. Zymeri, “9 ways background music can influence your restaurant's customers,” *Backbar Academy*. [Online]. Available: https://academy.getbackbar.com/9-ways-background-music-can-influence-your-restaurants-customers. [Accessed: 17-Oct-2022].

[2] Victrola, “Jukebox history: The evolution of the Jukebox,” *Victrola*, 09-Dec-2020. [Online]. Available: https://victrola.com/blogs/articles/jukebox-history-the-evolution-of-the-jukebox. [Accessed: 17-Oct-2022].

[3] 262588213843476, “Summary of 'clean code' by Robert C. Martin,” *Gist*. [Online]. Available: https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29. [Accessed: 09-Mar-2023].