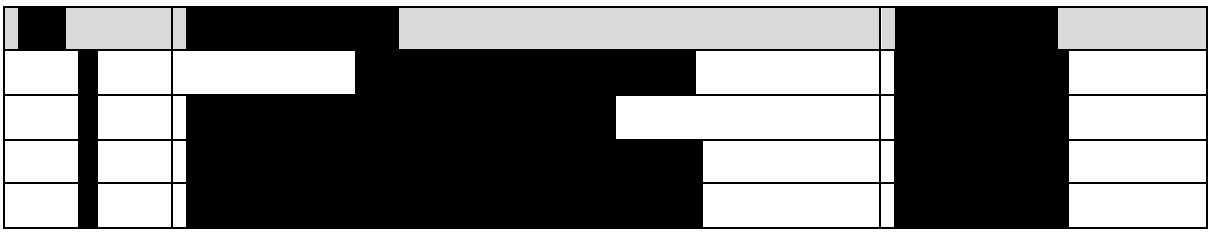
[redacted]

Software Quality Assurance and Testing

Trident Music

[redacted]



[redacted]

Software Test Plan

for

Trident Music

Version 1.0 approved

[redacted]

[redacted]

[redacted]

TM Industries

26-12-2020

Table of Contents

Revision History

Version	Updated by	History	Description
122011.0917.01	[redacted]	11th December,2020	Launch of initial version of project.
122031.1020.10	[redacted]	31th December,2020	Major fix on system feature and release of final version.

1. TEST PLAN IDENTIFIER: TM-FTP01.0

2. REFERENCES

- Software Requirement Specification (SRS) Document

3. INTRODUCTION

3.1 Background to the Problem

Trident Music is a lossless flac music streaming service for solo artists to thrive. Although music streaming services like Spotify, Tidal and Deezer are already available in the market but none of them pays well to the artists. Trident Music tries to solve this issue by supporting solo artist without having to deal with a record label. When an artist signs a deal with a record label, they get only 40% - 50% of the money of each track sold or streamed. Also, there are no platforms for good solo artists to make money by selling music unless they sign a deal with a record label. Furthermore, subscribers can buy DRM free music or stream which is currently unavailable on other platforms like Spotify and Tidal. Trident Music tries to eradicate these issues.

3.2 Solution to the Problem

As the artists are not being well paid on other platforms, a new platform was necessary. Changing the whole strategy of Spotify and Tidal to meet the needs of solo artists is not currently possible. Therefore, it would be more sensible to create a totally new platform. The business objective is to get as many subscribers on the platform as possible and with the features discussed below will surely play a role in gaining subscribers.

On Trident Music, Solo artists can create account as well as record labels and upload tracks. Subscribers of Trident Music can listen to each and every track uploaded by the solo artists and the record labels by paying a small amount of fee each month. Tracks downloaded via subscription can be only played by the Trident Music but subscribers can also buy DRM free music that can be played with any type of music player on any device. Solo artists can sell their music and for the record labels, they will get a small amount of money each time their music is streamed.

Music streaming services like Spotify and Tidal are two major players but these services are targeted towards record labels and pays a little amount of money to the artists.

The features that makes Trident Music unique from Spotify and Tidal are,

- Subscribers can buy DRM free music that can be played on wide range of devices and music players.
- Artists can release and sell tracks on their own. But they will have to create an artist account by paying a small amount of fee first.
- Subscribers will have access to millions of songs with the highest possible quality (lossless flac). But in order to enjoy the highest quality, an adequate internet speed is required.
- Artists and Record labels will be able to organize virtual concerts (VR/Normal).

This project does not duplicate other major streaming services rather extends their features being a standalone music streaming service of its own.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. **App Login and Signup**

    1.1 The app will have three types of users *(Consumer, Artist and Record Label).*
    1.2 Users will be able to login to the app with a valid email and password.

1.3 If the email/password is incorrect the app will prompt to try again or create a new account if the app fails to find the inputted email address in *'Trident'* database.
1.4 Users should be able to enable Tow-Factor Authentication that will send a unique code to their email each time upon login to set an extra layer of protection to the account.
1.5 The app will check if the user is logging in from a valid region *(Where trident legally available).*
1.6 Artists and Record labels must verify themselves as authentic while signing up.
1.7 Artist account can be linked to a Record-label while creating. *(Record-label will have to approve)*
1.8 Users should be able to choose to remember password for faster future logins.
1.9 Users should be able to reset password.

Priority level: High

Precondition(s): User email address and password should be valid.

Cross-reference: None

2. Consumer Side Features

2.1 Consumers should be able to browse the whole catalog without buying a subscription.
2.2 Music can be listened to without buying or subscription with limited functionality *(No skips)* and in-app ads.
2.3 Consumer should be able to listen to the whole catalog without any interruptions if they purchase a subscription.

Priority level: High

Precondition(s): Must be logged in.

Cross-reference: 5

3. Artist Side Features

3.1 Individual artists should able to release tracks and albums at their own pace.
3.2 Artists should be able to listen to whole catalog without buying a subscription.
3.3 Artists should be able to see statistics of how their tracks and albums are doing. (Times streamed, Times bought)

Priority level: High

Precondition(s): Must be logged in.

Cross-reference: 5

4. Record-label Features

4.1 Record-labels should be able to add Artists.

4.2 Record-labels must have the option to upload/release music.

4.3 Record-labels should be able to organize VR concerts.

4.4 Record-Labels should be able sell concert tickets. (Price is determined by the Record-label)

Priority Level: High

Precondition(s): Must be logged in using Record-Label account.

Cross-reference: None

5. Player Features

5.1 Users should be able to buy DRM free music. *(Valid payment method is required)*

5.2 Users should be able to purchase VR music tickets if their system meets the minimum requirements.

5.3 Users should be able to create playlists which can be added to 'Favorites' by other users.

5.4 Users must be able to follow each other.

5.5 Music and Album can be added queues, favorites and playlists by users.

5.6 Users will have the 'Music Library' feature. Which will allow users to add and synchronize songs on their devices and playlists.

5.7 Users can share music and album link via social media.

5.8 Users can link their social media accounts with Trident account.

5.9 There will be an Ai-based recommendation algorithm that selects songs for users according to their genre tastes.

5.10 Users can see listening history of other users. *(If turned on and linked with social medias)*

Priority Level: High

Precondition(s): Some feature needs a subscription and a valid payment method.

Cross-reference: 6

6. Features

6.1 System will learn about music genres by analyzing text, music tags and similar terms

associated with artist across the trident database.

6.2 There will be a 'Push notification' option. Which will notify the users when their favorite artists or record label releases songs, live concerts etc. *(If enabled)*

6.3 Only lossless flac files should be uploaded to the system. *(24Khz, >=940kbps bit rate)*

6.4 System should be able to pre-cache frequently played music for faster loading.

6.5 System will automatically determine the music quality based on user's internet speed while streaming. *(Can be changed manually in settings)*

6.6 Interactive VR concerts must be developed with Unreal Engine.

6.7 Users should be able to choose the number of devices signed at the same time using the same account.

Priority Level: High

Precondition(s): HTC Vive, Occulus Rift and PS VR are needed for Interactive VR concerts.

Cross-reference: None

7. Billing and Payments

7.1 Users should be able to add payment methods in the app settings. (Credit card/ Gift Card)

7.2 The app should be able to update payment information in real-time.

Priority Level: High

Precondition(s): Must be logged in first.

Cross-reference: None

8. Sign out Functionality

8.1 Users should be able to sign out anytime they want.

8.2 Users will have the feature to sign out of every device they have signed in with a single click.

Priority Level: High

Precondition(s): Must be logged in first.

Cross-reference: None

4.2 System Quality Attributes

1. **Availability**

   System should be up and running 99.8% of the time.

   Priority Level: High

   Preconditions(s): None

   Cross-reference: None

2. **Integrity**

   a. Users should not be able to decrypt downloaded DRM protected music files.
   b. User emails must be authenticated before signing up.
   c. Users should have a least amount of money before they can add their credit card in payment details.

   Priority Level: High

   Preconditions(s): None

   Cross-reference: None

3. **Maintainability**

   a. Any kind of maintenance that directly hampers user experience should be completed under 40min to 1hour.

   Priority Level: Medium

   Preconditions(s): None

   Cross-reference: None

4. **Robustness**
    a. System should be able to recover state of currently playing music when user re-runs the application after an accidental exit.
    b. System should be able to pause currently downloading contents before it crashes and resume when user re-runs the application.

    Priority Level: High

    Preconditions(s): None

    Cross-reference: None

5. **Usability**
    a. User interfaces should be self-explanatory so that new users can feel right at home without going through tedious documentation and other user guides.

    Priority Level: Medium

    Preconditions(s): None

    Cross-reference: None

6. **Interoperability**
    a. Payment gateway should be integrated in the app and it should be able exchange data between card and the system.

    Priority Level: Medium

    Preconditions(s): None

    Cross-reference: None

4.3 Project Requirements

1. A working system should be ready within 20 weeks.

2. The application should not consume more than 250mb storage space after installation.
3. The desktop application will be built using Electron framework.
4. Developers are free to use their preferred Code Editors.
5. BitBucket will be used as the default version control system and code management tool.
6. JIRA will be used as the default platform for bug tracking and agile process management.
7. Continuous Integration, Delivery and automated tests will be done by using Bamboo.
8. Interactive prototype will be done using Azure.
9. Illustrative prototype will be done using Affinity Designer.
10. Budget for the complete application is $50,000.

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed.

- VR compatibility in the quality of streaming

6. TESTING APPROACH

6.1 Testing Levels

The testing for the Trident music will cover of Unit, System/Integration and Acceptance test levels. Most testing will be done by the test manager with the development teams' participation.

- **UNIT** Testing will be done by the developer and will be approved by the development team leader. It will be guided by Rumi, Fatematuz Johora.

- **SYSTEM/INTEGRATION** Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. It will be guided by Priyanka, Afreen Hasan.

- **ACCEPTANCE** Testing will be performed by the actual end users with the assistance of the test manager and development team leader. It will be guided by Hamim, Faieq.

6.2 Test Tools

The only test tools to be used are the standard AS/400 provided utilities and commands.

- The Program Development Manager (PDM) will be used as the source version configuration management tool in conjunction with the in-house check-in/check-out control utility. The check-in/out utility is part of each developer's standard AS/400 access menu.

- The initial prototypes for the new screens will be developed using the AS/400 Screen Design Aid (SDA). The initial layout and general content of the screens will be shown to the sales administration staff prior to proceeding with testing and development of the screens.

6.3 Meetings

The test team will meet once every two weeks to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well.

7. TEST CASES/TEST ITEMS

<table>
<tr><td colspan="3">Project name: Trident Music
Test Case: FR_1</td><td colspan="2">Test Designed by: [redacted]
Test Design date: 31/12/20</td></tr>
<tr><td colspan="3">Test Priority: High
Module Name: App Login and Signup</td><td colspan="2">Test Executed by: [redacted]
Test Execution date: 31/12/20</td></tr>
<tr><td colspan="5">Test Title: To test the login and signup functionality.
Test Description: Verify if logging in and signing up with a valid email and password works.</td></tr>
<tr><td colspan="5">Precondition: User email address and password should be valid.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status</td></tr>
<tr><td>1.Open app.
2. Enter username/email.
3.Enter valid password.
4.Click login/signup.</td><td>Email: example@e.c
Password: 123</td><td>User should logged in or signed up to the app.</td><td></td><td></td></tr>
<tr><td colspan="5">Post Condition: User is validated with details from database for login or user data has been written to the database.</td></tr>
</table>

Project name: Trident Music Test Case: FR_2			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: Medium Module Name: Consumer Side Features			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test consumer side features. Test Description: Verify if all consumer side features work as expected.				
Precondition: Must be logged in.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Browse music. 3. Download Music. 4. Play/Pause/Skip/Seek music. 5. Buy music or purchase subscription. 6. Buy concert ticket, download concert, play concert.	A valid user account.	User should be able to access all features without any interruptions.		
Post Condition: User data has been saved into the system.				

Project name: Trident Music Test Case: FR_3			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: Artist Side Features			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test artist side features. Test Description: Verify if all artist side features work as expected.				
Precondition: Must be logged in.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Browse music and Play/Pause/Skip/Seek. 3. Download, Play Concerts. 4. Release tracks and	A valid artist account.	Artist should be able to access all features without any interruptions and unexpected closing of the app.		

an album. 5. See statistics of released music.				
Post Condition: Artist usage data has been saved into the system.				

Project name: Trident Music Test Case: FR_4			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: Record-Label Features.			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test record-label features. Test Description: Verify if all record-label features work as expected.				
Precondition: Must be logged in using Record-Label account.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Add an artist to the label. 3. Upload music, release concert and put tickets up for sale	A valid record-label account.	Record-Label should be able to access all features without any interruptions.		
Post Condition: Record-Label usage data has been saved into the system.				

Project name: Trident Music Test Case: FR_5			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: Player Features			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test player features. Test Description: Verify if all player features work as expected.				
Precondition: A valid subscription and payment method.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Buy DRM free music. 3. Buy VR concert	A valid user or artist account with a valid subscription or payment method	All player features should function as expected without any streaming glitch or		

tickets. 4. Create playlist. 5. Follow another user or artist. 6. Add music to queue, favorites and playlists. 7. Share music link on social media. 8. Link social media account to Trident. 9. See listening history of a follower user.	added.	force close.		
Post Condition: User activity has been saved into the system.				

Project name: Trident Music Test Case: FR_6			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: System Features			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test system features. Test Description: Verify if all system features work as expected.				
Precondition: A valid subscription and payment method.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Check if system suggests similar tracks to previous listening activity. 3. Enable push notification option from settings. 4. Play a previously listened music. 5. Play music with poor internet connectivity. 6. Limit the number of devices signed in with current	A valid user or artist account with a valid subscription or payment method added.	User should be able to play music with poor internet connection with reduced quality also push notifications should work.		

account.				
Post Condition: User activity and changed settings has been saved into the system.				

Project name: Trident Music Test Case: FR_7			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: Billing and Payments			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test billing and payments methods. Test Description: Verify if all billing and payments functionality works as expected.				
Precondition: Must be logged in first.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Add a payment method.	A valid user account and a credit with minimum balance.	Desired payment method should be added without any failures.		
Post Condition: Payment method has been saved into the system.				

Project name: Trident Music Test Case: FR_8	Test Designed by: Same as previous Test Design date: Same as previous

Test Priority: Medium Module Name: Sign out Functionality			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test sign out functionality. Test Description: Verify if signed in user can sign out without issues.				
Precondition: Must be logged in first.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1. Open app. 2. Go to settings and select sign out from all devices. 3. Sign out from current session.	A valid user, artist or record-label account.	User should be signed out from all previously signed in devices and from current running session.		
Post Condition: Sign out activity has been logged.				

Project name: Trident Music Test Case: NFR_1			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: High Module Name: Availability			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test server up time. Test Description: Verify if server and system are up and running 99.8% of the time.				
Precondition: System must be up and running.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1.Visit Website plant 2. Fill website address (URL)	System/Server Down checker (Website Planet)	System will be up and running 99.8 percent of the time.		
Post Condition: System uptime data has been saved into the system.				

Project name: Trident Music Test Case: NFR_2	Test Designed by: Same as previous Test Design date: Same as previous
Test Priority: High Module Name: Integrity	Test Executed by: Same as previous Test Execution date: Same as previous

Test Title: To test email verification, credit card verification and DRM protected music integrity. Test Description: Verify if emails, credit cards are authentic and DRM protected music can't be decrypted.				
Precondition: System and server must be up and running.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1.Sign up. 2. Before login User need verify email. 3.Go to bincode.com 4.Check credit card authenticity. 5.Test DRM protected music decryption level.	Email and credit card checker (bincode.com/creditcard-cheker)	Fake email and credit cards won't be accepted by system. DRM protected music decryption won't be easy.		
Post Condition: Email/Credit card data has been saved into the system.				

Project name: Trident Music Test Case: NFR_3	Test Designed by: Same as previous Test Design date: Same as previous
Test Priority: Medium Module Name: Maintainability	Test Executed by: Same as previous Test Execution date: Same as previous

Test Title: To test maintenance time is between 40 min to 1 hour. Test Description: Verify if maintenance time is between 40 min to 1 hour or not.				
Precondition: System/Server must be down.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1.Start Maintenance 2.Check maintenance time	Maintenance time.	Maintenance time should be between 40 min to 1 hour.		
Post Condition: Maintenance data has been saved into the system.				

Project name: Trident Music Test Case: NFR_4	Test Designed by: Same as previous Test Design date: Same as previous

Test Priority: High Module Name: Robustness			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test recover state of music player and download after accidental shut down of application. Test Description: Verify if music player and download state are recovered after accidental shut down of application.				
Precondition: System/server must be up and running.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1.Open app. 2.Play a music 3.Also download a song in Background. 3. Turn off device.	State of music player and download.	State of music player and download should be recovered after re-opening of app.		
Post Condition: State recovery data has been saved into the system.				

Project name: Trident Music Test Case: NFR_5			Test Designed by: Same as previous Test Design date: Same as previous	
Test Priority: Medium Module Name: Usability			Test Executed by: Same as previous Test Execution date: Same as previous	
Test Title: To test user friendliness of system. Test Description: Verify if system is user friendly or not.				
Precondition: System/server must be up and running.				
Test Steps	Test Data	Expected Result	Actual Result	Status
1.Open app. 2.Play a music 3.Download music. 4. Browse music. 5. Check Transitions.	User rating.	User rating must 4.5 or above out of 5.		
Post Condition: User rating data has been saved into the system.				

Project name: Trident Music Test Case: NFR_6	Test Designed by: Same as previous Test Design date: Same as previous
Test Priority: Medium Module Name: Interoperability	Test Executed by: Same as previous Test Execution date: Same as previous
Test Title: To test data exchange function between card and the system. Test Description: Verify if data exchange happing between card and system.	
Precondition: System/server must be up and running.	

Test Steps	Test Data	Expected Result	Actual Result	Status
1.Open app. 2.Go to payment section 3.Give credit/Visa card data. 4. check transaction.	Exchange data between card and system.	Data should be exchange between system and card.		
Post Condition: Exchange data has been saved into the system.				

8. ITEM PASS/FAIL CRITERIA

The test process will be completed when the beta version is completed. When the sales administration staff is satisfied that the sales graph is satisfactory after consecutive 5 months of sales data.

9. TEST DELIVERABLES

- Acceptance test plan
- System/Integration test plan
- Unit test plans/turnover documentation
- Screen prototypes
- Report mock-ups
- Defect/Incident reports and summaries
- Test logs and turnover reports
- Fixed bug report

10. STAFFING AND TRAINING NEEDS

11. RESPONSIBILITIES

Role	Minimum Resources Recommended (number of full-time roles allocated)	Specific Responsibilities or Comments
Test Manager		Provides management oversight. Responsibilities include: • planning and logistics • agree mission • identify motivators • acquire appropriate resources • present management reporting • advocate the interests of test • evaluate effectiveness of test effort
Test Analyst		Identifies and defines the specific tests to be conducted. Responsibilities include: • identify test ideas • define test details • determine test results • document change requests • evaluate product quality
Test Designer		Defines the technical approach to the implementation of the test effort. Responsibilities include: • define test approace • define test automation

		architecture • verify test techniques • define testability elements • structure test implementation
Tester		Implements and executes the tests. Responsibilities include: • implement tests and test suites • execute test suites • log results • analyze and recover from test failures • document incidents
Test System Administrator		Ensure test environment and assets are managed and maintained. Responsibilities include: • administer test management system • install and support access to, and recovery of, test environment configurations and test labs
Database Administrator, Database Manager		Ensures test data (database) environment and assets are managed andmaintained. Responsibilities include: • support the administration of test data and test beds (database)
Designer		Identifies and defines the operations, attributes, and associations of the test classes. Responsibilities include:

		• defines the test classes required to support testability requirements as defined by the test team
Implementer		Implements and unit tests the test classes and test packages. Responsibilities include: • creates the test components required to support testability requirements as defined by the designer.

12. TESTING SCHEDULE

Date	Test Item
November 2 – November 7	Login and signup functionality
November 8 – November 12	Consumer side features
November 13 – November 15	Artist side features
November 16 – November 20	Record-label features
November 21 – November 24	Player features
November 25 – November 28	System features
November 29 – November 30	Billing and payment methods
December 2 – December 7	Sign out functionality
December 8 – December 12	Server up time
December 13 – December 15	Email verification, credit card verification and DRM protected music integrity
December 16 – December 20	Maintenance time is between 40 min to 1 hour
December 21 – December 24	Recover state of music player and download after accidental shut down of application
December 25 – December 28	User friendliness of system
December 29 – December 31	Data exchange function between card and the system

Note: All the test items are divided into 4 parts and each of the member has done their part. From November 2 – November 20, test has been done by [redacted]. From November 21 – November 30, test has been done by [redacted]. From December 2 – December

20, test has been done by [redacted]. From December 21 – December 31, test has been done by [redacted].

13. PLANNING RISKS AND CONTINGENCIES

1. Prior to the music app launch, it's easy to focus all of our resources on development alone. While it's essential to build a high performing app, marketing efforts are just as important in order to ensure success once your app has hit the app stores.

2. Consumers are more likely to download an app if they can see its value. Strong ratings and reviews provide potential users with insight before they've even downloaded the app. additionally; App store algorithms consider reviews and ratings as a part of their ranking system. In order to rank high, your app needs to have positive feedback from its users.

3. Marketing efforts should never be an afterthought. We should start when we decide to begin building our app. This is the most important element which cannot be ignored.

14. APROVALS

Project Sponsor	[redacted]
Development and Testing team	[redacted]
Project manager	[redacted]
Test manager	[redacted]
Development team Manager	[redacted]

The End