

# Entertainment

## Requirements Document

### 1. Introduction

#### 1.1 Overview

'Entertain Me' is a cross-platform mobile application that connects freelance entertainers, artists, and socialites to businesses and hosts that are looking to hire entertainers to their events. It provides entertainers a platform to showcase and market their talents to a wide audience.

#### 1.2 Scope of the Product

'Entertain Me' is a cross-platform mobile application that helps people to find, communicate, negotiate and book entertainers seamlessly. The application upon completion will be available in both Android and IOS App stores for the users to download.

#### 1.3 Business Case for the Product

The key business value of this application will be its ability to connect the people trying to host events and the entertainers. Now, most people might be getting to know about entertainers through word of mouth and negotiations might be happening through phone calls, mail or in-person for bigger events. 'Entertain me' will act as an alternative to this archaic process by providing a platform that makes hiring entertainers easy.

### 2. General Description

'Entertain Me' is a mobile app that allows anyone, anywhere to hire LIVE entertainment. The App removes friction from the LIVE entertainment booking process by seamlessly facilitating discovery, communication, scheduling, agreement, and payment.

#### 2.1 Product Perspective

'Entertain Me' is a one-stop talent booking management platform. By streamlining the entire talent booking process, 'Entertain Me' helps ease the booking process from start to finish. This application also helps entertainers to connect with each other and expand their network. This

project is proposed by an entertainer and will be delivered by the Department of Computer Science, University at Buffalo.

## 2.2 Product Functions

### Host Account

- Log in/Sign up.
- Search for entertainers based on their talent, budget, name and location of the event.
- Propose bookings, negotiate and book the entertainers.
- Create private or public events.
- View public events created by other users.
- Rate the booked entertainer
- s and provide reviews.
- View/Cancel their bookings.
- Upgrade from user to entertainer.

### Entertainer Account

- Search for entertainers based on their talent, budget, name and location of the event.
- Propose bookings, negotiate and book the entertainers.
- Create private or public events.
- View public events created by other users.
- Rate the booked entertainers and provide reviews.
- View/ Cancel their bookings.
- Exist as freelancer or link themselves to a business account.
- Negotiate and accept bookings from users if they are a freelancer.

### Business Account

- Log in/Sign up.
- Search for entertainers based on their talent, budget, name and location of the event.
- Propose bookings, negotiate and book the entertainers.
- Create private or public events.
- View public events created by other users.
- Rate the booked entertainers and provide reviews.
- View/ Cancel their bookings.
- The business account has all the functionality that a user has.
- Negotiate and accept bookings from users for all entertainers linked to them.

## 2.3 User Characteristics

The target demographic of this application is mostly people between the age of 20 - 40. They can be from any background and should have used a smartphone before. The users who haven't used a smartphone before may experience some difficulty in using the application, but with little guidance, they should be able to use it.

## 2.4 General Constraints

The application should work on both Android as well as IOS mobiles.

## 2.5 Assumptions and Dependencies

We assume that the users will have email addresses in order to sign up.

The AWS free tier has the following capacities.

- 750 hours of Amazon RDS Single-AZ db.t2.micro Instance usage running MySQL, MariaDB, PostgreSQL, Oracle BYOL or SQL Server (running SQL Server Express Edition) – enough hours to run a DB Instance continuously each month
- 20 GB of General Purpose (SSD) DB Storage
- 20 GB of backup storage for your automated database backups and any user-initiated DB Snapshots

We assume that the AWS tier should be sufficient to handle user traffic as the number of concurrent users is estimated to be around 5.

# 3. Specific Requirements

## 3.1 User Requirements

### 3.1.1 Host Requirements

1. Hosts should be able to sign up with his/her username, password, email id.
2. Hosts should receive the OTP via the email address that he/she provided while signing up. Host should provide the same OTP in the mobile application to verify the email id.
3. Hosts should be able to view "Manage account" in which he/she can edit his/her profile. The host can update the name and password.

4. Any user should be able to create an entertainer profile by filling out talents, talent category, years of experience, location, price range, sample of past performances which should be verified by Admin.
5. Hosts should be able to view his/her past and upcoming bookings under "Manage account" and cancel any upcoming bookings.
6. Hosts should be able to login with username and password.
7. Hosts should be able to search for entertainers with Name.
8. Hosts should be able to sort the search results alphabetically(default) and based on highest to lowest rating, years active and the number of events completed.
9. Hosts should be able to view available public events.
10. The host should be able to view the entertainer profile and connect with them by filling out the pre-requirement form which includes the below fields by default. The entertainer can also add extra fields based on his requirement.
  - a. event\_time
  - b. event\_name
  - c. event\_location
  - d. event\_date
  - e. numb\_of\_attendee
  - f. Monetary compensation
  - g. Comments
11. If the form is declined by the entertainer, the host will be notified via email and push notification and they can use the chat to negotiate.
12. Once the form is accepted by the entertainer, the host can book the entertainer by paying the minimum deposit - 10% upfront deposit. Payment gateway is integrated into the application and allows the host to pay the entertainer using the app.
13. Upon arriving at the location, the host and entertainer will verify arrival to guarantee a minimum payment. Verification can be in the form of a randomly generated multi-digit code, unique to each event, or a bar-code. Further verification will occur in the form of a randomly generated multi-digit code, unique to each event, or a bar-code at the end of the event to guarantee full payment.
14. Hosts can rate entertainers on the basis of multiple factors such as punctuality, professionalism, attitude.
15. Host should be able to provide general complaints and customer feedback.

### 3.1.2 Entertainer Requirements:

16. Entertainers should receive the OTP via the email address that he/she provided while going for the event. Host should provide the same OTP in the mobile application to verify the email id.
17. Entertainers should be notified by mail and notification from the mobile application wherever his/her account is verified and activated by Admin.

18. Entertainers should be able to view “Manage account” in which he/she can edit the following details.
  - a. age
  - b. profile\_pic
  - c. description
  - d. events\_completed
  - e. Location
  - f. last\_performance
  - g. type\_of\_entertainer
  - h. Earnings
  - i. Sample videos
19. Entertainers should be able to view his/her past and upcoming bookings under “Manage account”.
20. Entertainers should be notified if his/her bookings are cancelled by the host.
21. The Entertainer should be notified by mail whenever his/her is promoted to administrative positions.
22. Entertainers should be able to view and add fields to the pre-requirement form. The default fields in the form are
  - a. Date
  - b. Location
  - c. number of attendees of the event
  - d. Compensation
23. Entertainers should be notified via email by the mobile app whenever their pre-requirement form is filled out by a host. Entertainers can view the form and accept/decline it.
24. Entertainers should be notified by mail whenever the host makes payment and books the entertainer.
25. Entertainers can use the chat feature in the application to negotiate and discuss with the host.
26. Upon arriving at the location, the host and entertainer will verify arrival to guarantee a minimum payment. Verification can be in the form of a randomly generated multi-digit code, unique to each event, or a bar-code. Further verification will occur in the form of a randomly generated multi-digit code, unique to each event, or a bar-code at the end of the event to guarantee full payment.

### 3.1.2 Business Requirements:

27. Business Should be able to view Entertainers linked to it.
28. Business should be able to have Entertainers.

### 3.1.3 Admin Requirements:

#### 3.1.3.1 Main Admin Requirements:

29. Admin should be able to confirm/deny entertainer's profiles to finalize their account activation.
30. Admin should be able to view all booked events with full view of public event information and limited view of private event information(location information is hidden).
31. Admin should be able to promote accounts to administrative positions.
32. Admin should be able to view/respond to complaints and customer feedback.
33. Admin should be able to add or modify talent categories.
34. Admin should be able to view/deactivate accounts after a period of inactivity and cancel events (to prevent potentially illegal events/activities from taking place).

#### 3.1.3.2 Requirements of Accounts promoted to Admin Status:

35. He/She should be able confirm/deny entertainer profiles to finalize their account activation.
36. He/She should be able to respond to customer feedback.

## 3.2 System Requirements

1. SetUp React Native and Node.js which is the tech stack of the Project.
2. Data needs to be secured. Hence login credentials should be handed over to the client.
3. Admin should be given access to block or delete user profiles.
4. Aws should be scalable and should be able to handle variation in user traffic and it should be always up.

## 3.3 Interface Requirements

<https://www.figma.com/proto/m8f2bfylZTTNOPxlZ9t7Cn/Entertainment-Wireframe?node-id=9%3A109&scaling=scale-down>

The above link is the wireframe of the application. It provides the look and feel of the working prototype of the application.

## 4. Appendices

Client's responses to questionnaire regarding requirements:

[https://docs.google.com/document/d/12zFGIKZNCWk7zFtAbsSbitj5qxo\\_QtwnaAM0N8I2Es/e/dit?usp=sharing](https://docs.google.com/document/d/12zFGIKZNCWk7zFtAbsSbitj5qxo_QtwnaAM0N8I2Es/e/dit?usp=sharing)

Brief description of the required product given by the Client during the first meeting:

[https://docs.google.com/presentation/d/1BiqJiURfUrJtcpsT8U6-XsK5HkkW3xhONT1YaALx9SQ/edit?usp=sharing\\_eip&ts=5e3b9727](https://docs.google.com/presentation/d/1BiqJiURfUrJtcpsT8U6-XsK5HkkW3xhONT1YaALx9SQ/edit?usp=sharing_eip&ts=5e3b9727)

## 5. Glossary

**React Native:** React Native is an open-source mobile application framework created by Facebook. It is used to develop cross-platform applications that work on both Android and IOS devices by enabling developers to use React along with native platform capabilities.

**Node.js:** Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser.

**IOS:** Operating System for Apple iPhones.

**Android:** Operating system for Android mobile phones.

## 6. References

1. Figma Prototype/wireframe tool - <https://www.figma.com/>