Contents

[Feature 1: General Landing page 2](#_Toc43999964)

[**User Story: 1** 2](#_Toc43999965)

[**User Story: 2** 3](#_Toc43999966)

[Feature 2: Un-registered talent use case 4](#_Toc43999967)

[**User Story: 1** 4](#_Toc43999968)

[**User Story: 2** 5](#_Toc43999969)

[**User Story: 3** 6](#_Toc43999970)

[Feature 3: Registered talent use case 7](#_Toc43999971)

[**User Story: 1** 7](#_Toc43999972)

[**User Story: 2** 8](#_Toc43999973)

[**User Story: 3** 9](#_Toc43999974)

[**User Story: 4** 10](#_Toc43999975)

[**User Story: 5** 11](#_Toc43999976)

[Feature 4: Un-Registered Company/Vendor. 12](#_Toc43999977)

[**User Story: 1** 12](#_Toc43999978)

[**User Story: 2** 13](#_Toc43999979)

[Feature 5: Staff Use cases 14](#_Toc43999980)

[**User Story: 1** 14](#_Toc43999981)

[**User Story: 2** 15](#_Toc43999982)

[**User Story: 3** 16](#_Toc43999983)

[**User Story: 4** 17](#_Toc43999984)

[**User Story: 5** 18](#_Toc43999985)

[**User Story: 6** 19](#_Toc43999986)

[Feature 6: Admin Use cases 20](#_Toc43999987)

[**User Story: 1** 20](#_Toc43999988)

[**User Story: 2** 21](#_Toc43999989)

[**User Story: 3** 22](#_Toc43999990)

[Feature 7: prerequisite features 23](#_Toc43999991)

[**User Story: 1** 23](#_Toc43999992)

[**User Story: 2** 24](#_Toc43999993)

# Feature 1: General Landing page

### **User Story: 1**

**Title: General Landing page and static content for all users.**

**Priority: High**

**Points: 5 Pts**

**As a product owner**

**I want to** create a landing page which is responsive. It should contain sections like

HOME, SERVICES, ABOUT US, Link to DATABASE (Talent), Clients & Testimonials, Upcoming Events and Login / Sign-up Button.

PS: In home page, few talents needs to be shown category wise and full talent database can be kept in a separate page in link with homepage.

so that it serves as a common public page for all sort of users as soon as they enter the[**www.pronationevents.com**](http://www.pronationevents.com)

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Talent Database.**

**Priority: High**

**Points: 8 Pts**

**As a product owner**

**I want to** create a responsive webpage with pagination for all registered talents.

When the logged-in user is staff or admin they should be able to filter by many criteria like Gender, talent, country, language etc.

When the logged-in user is fellow talent he/she should be able to only follow other talents.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 2: Un-registered talent use case

### **User Story: 1**

**Title: Registration.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**

**I want to** create a registration page for new talents with First name, last name, email, confirm email, phone, password and confirm password.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Follow up - Registration.**

**Priority: High**

**Points: 5 Pts**

**As a product owner**

Once the user submits the above details an email should be triggered with an activation link to user, upon clicking it should activate and collect more details like Personal, appearance, skills, experience, contact and photo details.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 3**

**Title: Restrict authentication.**

**Priority: High**

**Points: 1 Pts**

**As a product owner**

The registered talent will not be able to login until he/she receives approval in his/her registered mail ID.

Also, they will receive appropriate error while trying to login before he/she gets approval mail.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 3: Registered talent use case

### **User Story: 1**

**Title: Registration Approval.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**

**I want to** send an approval mail when the staff approves a talent registration. Now the registered talent must be able to login successfully.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Talent Dashboard.**

**Priority: High**

**Points: 5 Pts**

**As a product owner**

**I want to** create a dashboard for the registered talent.

The dashboard page can contain different sections like Jobs applied with status, response to events, profile, number of followers and following.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 3**

**Title: E-mail Template with action buttons.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**

**I want to** create an email template which can be send for any jobs or events.

The talent should be able to apply or ignore the requested job within the cutoff date.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 4**

**Title: Follow-Feature.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**.

The talent should also be able to follow other talents when he is logged in to the application.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 5**

**Title: Custom Gallery for Talent.**

**Priority: High**

**Points: 8 Pts**

**As a product owner**

**I want to** create functionality to build custom gallery for the Talent/Entertainer by allowing them to upload pictures and videos. [Photos/Videos should be embedded or save to external entities like S3 buckets, so that they don’t take storage in our application]

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 4: Un-Registered Company/Vendor.

### **User Story: 1**

**Title: Company Registration.**

**Priority: High**

**Points: 1 Pts**

**As a product owner**

**I want to** create a registration page for new company/vendor with Company name, contact name, email, confirm email, phone, password and confirm password.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Company Brief-Registration.**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

Once the user submits the above details an email should be triggered with an activation link to user, upon clicking it should activate and collect more details like business type, company info, activities and photo details.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 5: Staff Use cases

### **User Story: 1**

**Title: Staff Login.**

**Priority: High**

**Points: 1 Pts**

**As a product owner**

**I want to** create a backend credentials for staff in the database with encrypted password and hand it over to the client.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Display Registrations for approval.**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** display all the pending registration approvals of talents, company/vendors in the tabular form with Approve and Reject checkboxes.

The format of display should also support the feature to display the full of profile by clicking the name as a model.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 3**

**Title: Display response of job or event.**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** display all the response from talents, company/vendors for job specification or event requests.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 4**

**Title: Filter talents.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**

**I want to** select talents using customized filters and send email for any new jobs /events.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 5**

**Title: Capture Response**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** create a feature which can extract all profiles based on the response from talents and company/vendors.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 6**

**Title: Post Events.**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** create a page where events can be created and posted.

Also, it should trigger an email to all the talents regarding the created event.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 6: Admin Use cases

### **User Story: 1**

**Title: Create Customized Dashboard.**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** create a customized dashboard with widgets containing the data of consolidated number of talents, number of entertainers, number of suppliers, their statuses, number events posted and upcoming events. (Customized requirements).

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: Incorporate all Staff features**

**Priority: High**

**Points: 5 Pts**

**As a product owner**

**I want to** enable all features of above user story of staff.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 3**

**Title: Send requests to suppliers.**

**Priority: High**

**Points: 2 Pts**

**As a product owner**

**I want to** send request to all our registered company/vendors.

The requests should be sent to their registered mail IDs

TBD: Should their response to be tracked from mail or application

P.S At the moment the talents response for job specification and registration are handled through application only.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

# Feature 7: prerequisite features

### **User Story: 1**

**Title: Create DB Design for entire application**

**Priority: High**

**Points: 5 Pts**

**As a product owner**

**I want to** finalize the Database design as MYSQL for the application and document it.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.

### **User Story: 2**

**Title: AWS Account**

**Priority: High**

**Points: 3 Pts**

**As a product owner**

**I want to** create a new AWS account for demo purpose where the database and application can be hosted and released every week.

**Acceptance Criteria**

The above feature includes Database design, persistence retrieval.

Integration of service to front-end.

These Function should be unit tested.

Demo with AWS EC2.

Acceptance test from client and close the user story.