**Solution Architecture Specification**

|  |  |
| --- | --- |
| **Project Name** | … |
| **Customer** | … |
| **Last Saved Date** | … |
| **Revision** | … |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Organizational Position | Action | Date Reviewed |
| <approver> | < e.g. VP, IT Operations, Customer Name> | Approval |  |
| <reviewer> | <Other staff, Customer Name> | Reviewer |  |
| <reviewer> | <Other staff at Amristar> | Reviewer |  |
| <author> | Architect, Amristar | Author |  |

Change History:

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Last Revision By** | **Reason for Change** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Related Project Documents:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**Glossary**

Term Definition of term

**TABLE OF CONTENTS**

[**1.**](#_heading=h.3l18frh) **BUSINESS REQUIREMENTS AND GOALS 5**

[1.1.](#_heading=h.206ipza) Business Drivers 5

[1.2.](#_heading=h.4k668n3) Solution Requirements 5

[1.3.](#_heading=h.2zbgiuw) High Level Metrics for Success 5

[1.4.](#_heading=h.4d34og8) Architecture Boundary 5

[**2.**](#_heading=h.1egqt2p) **BUSINESS MODEL 6**

[2.1.](#_heading=h.3ygebqi) Organisational Context 6

[2.2.](#_heading=h.2dlolyb) Current Situation 6

[2.3.](#_heading=h.sqyw64) Solution Definition 6

[2.4.](#_heading=h.3cqmetx) Actors 6

[2.5.](#_heading=h.1rvwp1q) Interactions of Actors 6

[2.6.](#_heading=h.4bvk7pj) Use Cases 7

[**3.**](#_heading=h.2r0uhxc) **SOLUTION ARCHITECTURE MODEL 10**

[3.1.](#_heading=h.1664s55) Solution Components 10

[3.2.](#_heading=h.3q5sasy) Integration Specification 10

[3.3.](#_heading=h.25b2l0r) Walk-Throughs 11

[3.4.](#_heading=h.kgcv8k) Information Model 12

[**4.**](#_heading=h.34g0dwd) **DETAILED PHYSICAL ARCHITECTURE 14**

[4.1.](#_heading=h.1jlao46) Network Diagram 14

[4.2.](#_heading=h.43ky6rz) System Specification Summary 14

[4.3.](#_heading=h.2iq8gzs) Configuration Requirements 14

[**5.**](#_heading=h.xvir7l) **SOLUTION IMPACT 15**

[5.1.](#_heading=h.3hv69ve) Impact on Processes 15

[5.2.](#_heading=h.1x0gk37) Impact on People 15

[5.3.](#_heading=h.4h042r0) Technical Impact 15

[**6.**](#_heading=h.2w5ecyt) **QUALITY ATTRIBUTES 16**

[6.1.](#_heading=h.1baon6m) Reliability 16

[6.2.](#_heading=h.3vac5uf) Availability 16

[6.3.](#_heading=h.2afmg28) Serviceability 16

[6.4.](#_heading=h.pkwqa1) Best Practice Commentary 16

[6.5.](#_heading=h.39kk8xu) Architecture Commentary 16

[**7.**](#_heading=h.1opuj5n) **SUCCESS CRITERIA AND TEST PLAN 17**

[7.1.](#_heading=h.48pi1tg) Use Case Validation 17

[7.2.](#_heading=h.2nusc19) Validation of Solution metrics 18

[7.3.](#_heading=h.1302m92) Test Plan 19

[**8.**](#_heading=h.3mzq4wv) **PHASING, ROADMAP, AND IMPLEMENTATION PLAN 20**

[8.1.](#_heading=h.2250f4o) Technical Assumptions 20

[8.2.](#_heading=h.haapch) Solution Phases and Technical Milestones 20

[8.3.](#_heading=h.319y80a) Technical Work BreakDown Structure 20

[8.4.](#_heading=h.1gf8i83) Implementation Instructions 20

[**9.**](#_heading=h.40ew0vw) **APPENDIX – TEST PLANS 21**

# Business Requirements and Goals

## Business Drivers

What is the motivation for doing this? Which problem exists that needs solving, or which opportunity exists?

## Solution Requirements

High-level description of what the solution should accomplish

## High Level Metrics for Success

How will you know whether you have succeeded? Provide tangible measurable targets that should be met in order for your solution to be a success

## Architecture Boundary

What is in-scope, and what is not in-scope.

# Business Model

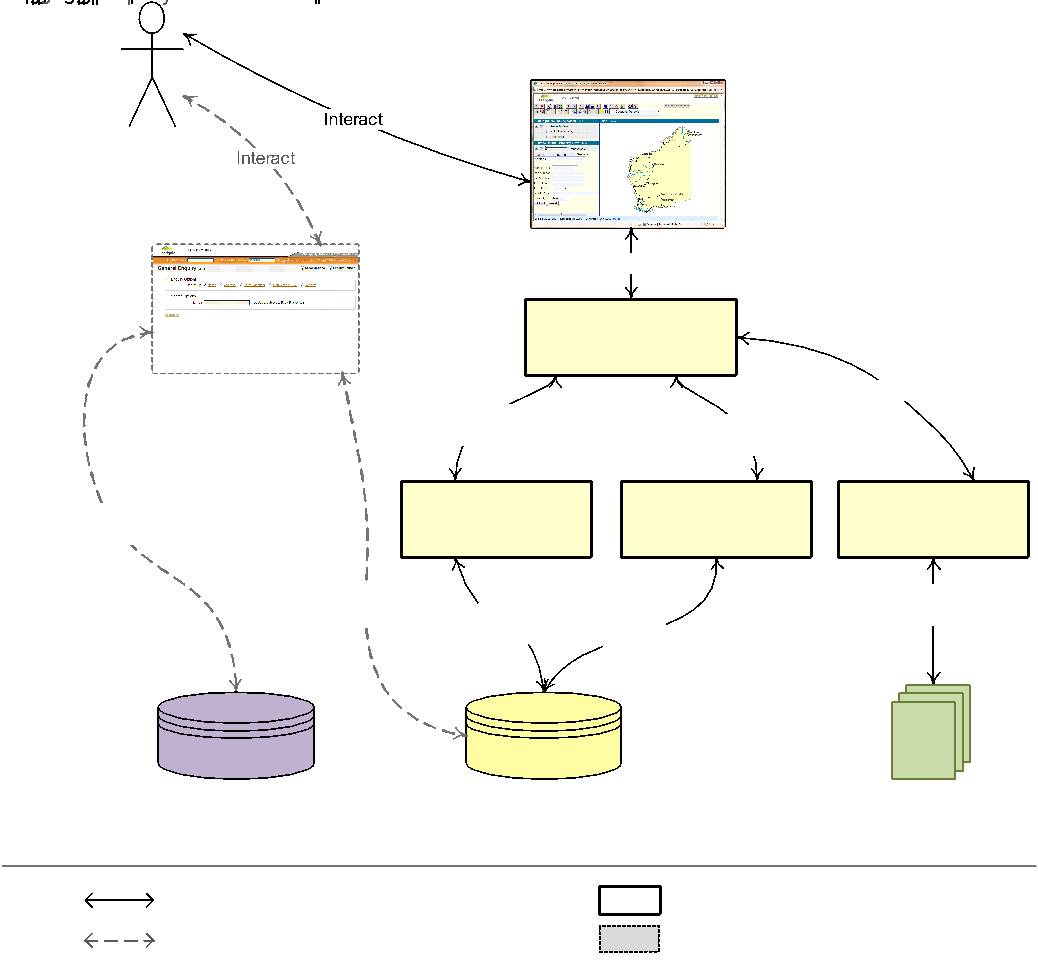
## Organisational Context

Provide a contextual diagram of the current situation, with a brief description

## Current Situation

Provide a more detailed description of the current processes, explaining what the pain points are or which opportunities exist. This section should ideally include a diagram that describes the current situation, with an accompanying description. This section should justify why something needs to be done.

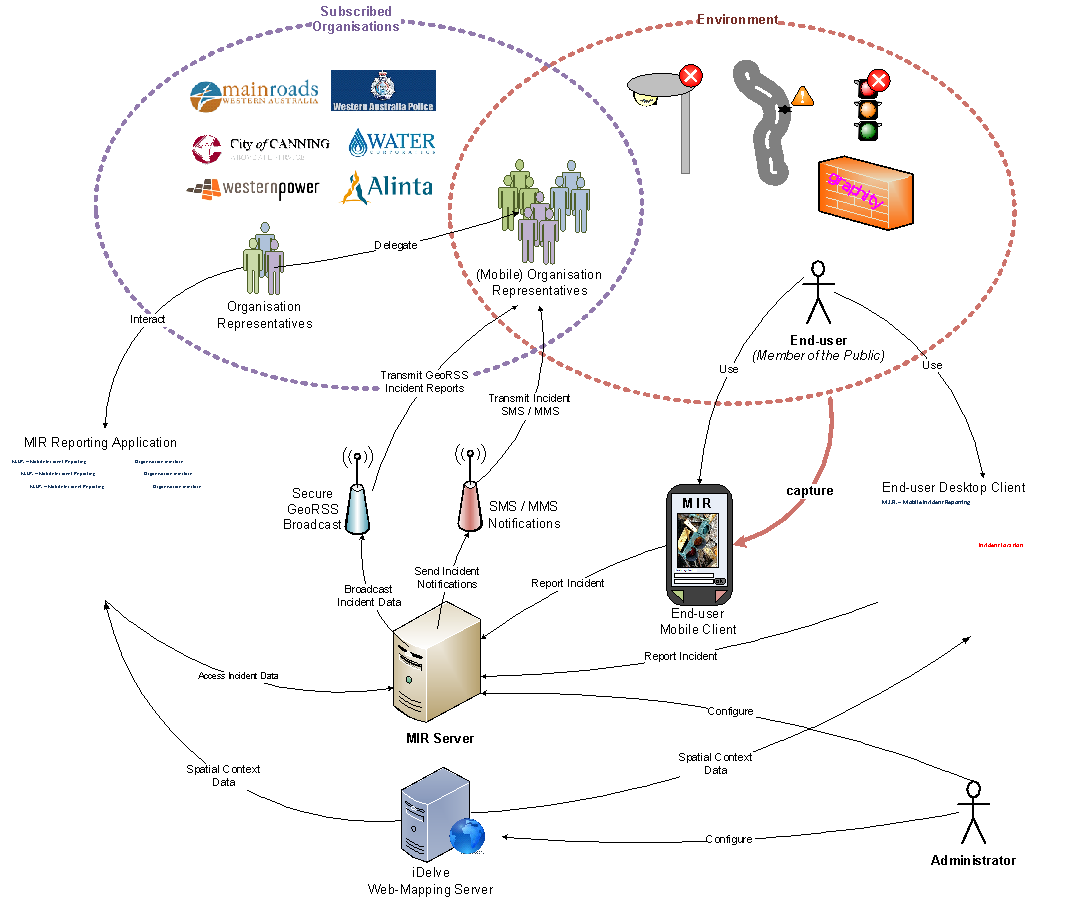
Example diagram:



## Solution Definition

Provide a high-level non-technical description of the solution, its functions and capabilities, and how it addresses pain points or takes advantage of opportunities. This section should include a diagram, and a description of the diagram. This section should explain why/how your proposed solution fixes the issues raised earlier.

Example diagram:



## Actors

Description of the actors of the new solution. These may be end-users, administrators, but could also include a scheduler (if it triggers action) and a different type of non-person actor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | **Description** | **Current Use of the System** | **Expected Use of the New Solution** |
| < \* Role - e.g. end-users, administrators> | < \* Resp-onsibility and affiliation. e.g. UNIX sysadmins> | < \* e.g. manual process via phone of fax> | < \* e.g. automated process> |

## Interactions of Actors

This should describe which business functions are accessed by which actors (some may be accessed by both) and includes a diagram. Sample diagram shown below.



## Use Cases

This section describes the use cases supported by the solution. A typical solution has about 7 use cases. If you have more than about 9, it means your solution either would have to be broken up, or you are going into a level of detail that is unnecessary for being able to describe the architecture.

Each use case should have a UML process flow diagram (cross-functional “swimlane” diagram)

Template:

|  |  |
| --- | --- |
| **Goal** | Describe the goal, but from a “functional” perspective, not from a technical implementation perspective. |
| **Actor(s)** | List actors who conduct this use case |
| **Description** | Describe the use case step-by-step, including possible options and exceptions. It must be detailed enough for readers to grasp the use case. |
| **Frequency** | Describe when and how often this use case is triggered/actioned |
| **Pre-processing** | What must have happened first before this use case is able to take place. For example, user registration, data sourcing, etc. |
| **Post-processing** | What happens after this use case takes place, e.g. send email, call Web Service, post-process a dataset, validate, etc. |
| **Included use case** | Which generic use case is included in this, e.g. “view details” or “send email” |
| **Extended use case** | Which use case does this extend? For example, “print page” an extension of “process order”? |

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Goal** | Process Incident Report | **Code** | MIR-04 |
| **Actor(s)** | Organisation Representative | | |
| **Description** | Organisation Representatives can perform several processing steps on an incident report:   * Delegate report – an incident report can be marked as delegated, to inform the user and the organisation that the incident is attended to. It can also be delegated to a different authority, if this is appropriate. * Export report(s) – organisations can export incident reports in delimited format for import into other business systems to manage incidents internally * Close resolved incident report – Organisation Representatives can mark an incident report as resolved, causing it to be closed and removed from the list of unresolved incidents * Save to dataset – when MIR is used as a data collection platform, an Organisation Representative can view inspect events before having them saved to a “master” dataset.   The Organisation Representatives must be presented with an error or warning if mandatory fields have not been filled out. | | |
| **Frequency** | As required. | | |
| **Pre-processing** | An incident report has been lodged by an End-user | | |
| **Post-processing** | Incident reports are marked as delegated, resolved, exported to a file, or saved to a master dataset | | |
| **Included use case** | MIR-04-1, MIR-04-2, MIR-04-3, MIR-04-4 | | |
| **Extended use case** | None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Goal** | Perform Sales Property Search | **Code** | MV-02 |
| **Actor(s)** | Public User / Registered User | | |
| **Description** | The User performs a search for a property or location in the sales shutter. The matching locations are returned as links. If a user clicks on a matching location, the map extent changes to this location. No sales information is presented within the search result. | | |
| **Frequency** | As required. | | |
| **Pre-processing** | The Property Finder is launched, and the sales shutter is visible. | | |
| **Post-processing** | The matching results are returned to the user as links that lead to a location. | | |
| **Included use case** | None. | | |
| **Extended use case** | None. | | |

Example Diagram:



# Solution Architecture Model

This section describes the architecture of the solution.

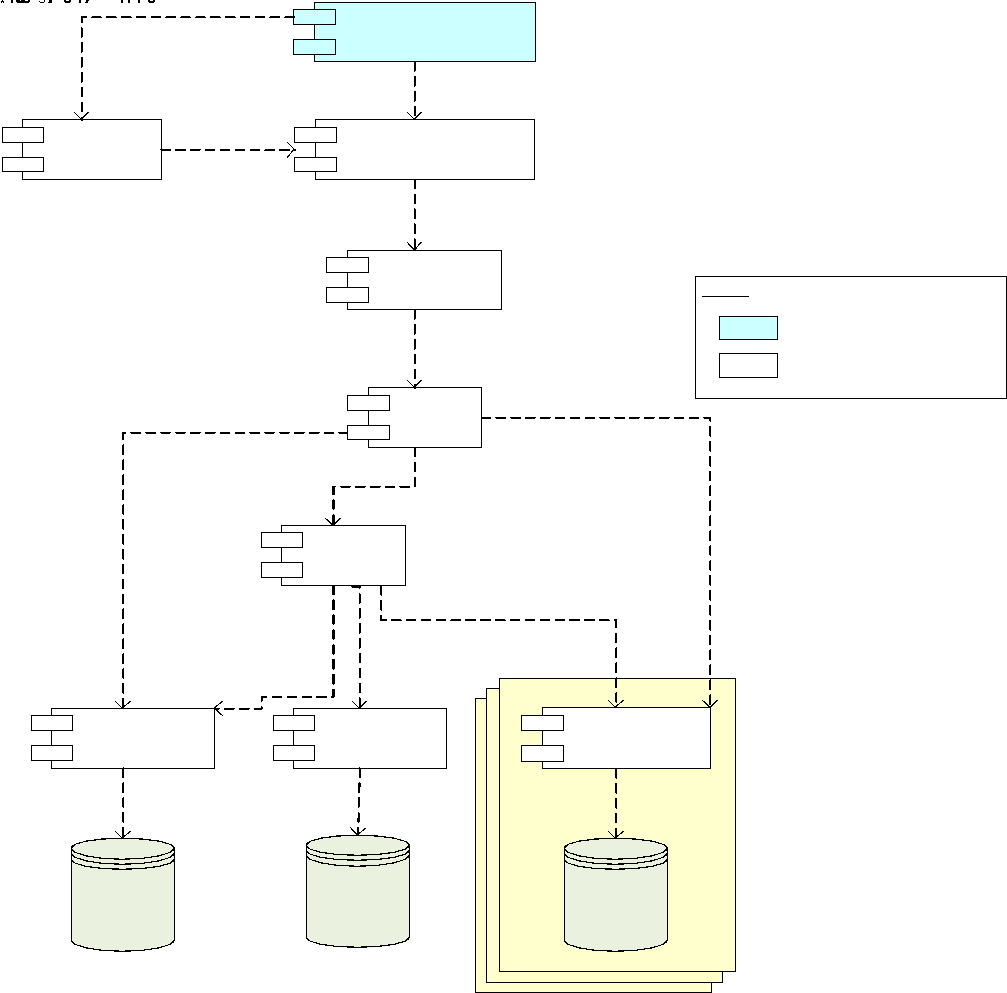
## Solution Components

Describe the components which make up the architecture. They can be internal or external. This section should not describe them from a marketing perspective, but purely from a functional perspective in the context of the solution.

## Integration Specification

This section should show an integration diagram of the solution to-be. It shows how each component depends on other components. It should have commentary describing the diagram.

Example diagram:



## Walk-Throughs

Provide a walkthrough for each use case, with “lanes” for each major component, and numbered arrows for each step. Snippets from diagrams or UI mockups can be included in this to help illustrate the link between the UI and back-end processes. The diagram should be preceded by a small description, and after the diagram there must be a detailed description for for each (numbered) step.

Example:

This activity involves the End-user viewing the status page of an incident report through a mobile or desktop browser, or an Organisation Representative viewing lodged incidents through the MIR Reporting Application.



Figure 22- View Incident Report

| **Step** | **Description** |
| --- | --- |
| 1. | The End-user accesses the Incident Viewer through a mobile or desktop browser, or the Organisation Representative accesses the MIR Reporting Application |
| 2. | The application is launched, ready for the End-user or Organisation Representative to request details of a particular incident. For the web-based status page, incident ID’s can be provided as GET variables in the URL. |
| 3. | After providing an incident number, or selecting an incident report from a list in the MIR Reporting Application, a request is sent to the MIR server for details on the incident, which are presented to the user |
| 4. | For the MIR Incident Reporting Application used by the Organisation Representative, a request is made to the iDelve server for mapping data. |
| 5. | Mapping data is obtained from the various data sources used by the iDelve server |
| 6. | Processed mapping data is sent back from the iDelve server to the MIR Incident Reporting application for display in the map |

## Information Model

This section has an ER-diagram and explanatory notes describing the entities and relationships. This section is only needed if the information model is sufficiently complex. In the case of 2 or 3 entities, it may not be needed.

Example:



# Detailed Physical Architecture

This section provides a detailed description of the physical architecture for each environment (DEV, UAT, PROD, etc.). It should include the physical servers and their specifications, and must allow the reader to see grouping of servers and geographic locations.

## Network Diagram

This section describes the locations of the various parts of the solution. It should cover the sites in scope of the solution, network connections, machines, routers, applications, DR site (where applicable). It should include port numbers, protocols, etc.   
This section should ideally include a diagram.

## System Specification Summary

This sections describes the specifications for each node in the previous section. A node may be a server, desktop, appliance, network device etc. The minimum disk space, memory and other sizing requirements should reflect the solution requirements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Node** | **Product**  **Versions**  **Options** | **Platform**  **OS**  **SP level**  **Kernel version** | **Memory**  **CPU**  **Processor Speed**  **# NIC cards** | **Minimum Disk Space** | **Quantity required** Or indicate if an existing system is being used. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Configuration Requirements

This section describes specific configuration requirements, which could include ports, firewall rules, etc.

# Solution Impact

This section should describe the impact which the solution has on processes, people and existing technology.

## Impact on Processes

This section describes the impact on processes, including newly required processes, procedures, or changes to existing policies. It could include introduction of change control mechanisms or bodies, processes for dealing with incidents. Existing processes should be cited where applicable.

## Impact on People

This section should describe impacts on roles, including changes to existing roles or new roles introduced as a result of the solution. It should describe resource requirements, including how many people are needed for each part of the solution, and the training required.

## Technical Impact

This section describes the technical impact, which can include changes to the SOE, network infrastructure, server infrastructure, etc.

# Quality Attributes

This section covers the quality attributes applicable to the solution. Different quality attributes to the ones below may be applicable, depending on the nature of the solution.

## Reliability

This should discuss things like fault tolerance, fault avoidance, (dynamic/elastic) scalability in case of spikes, expected capacity vs. maximum capacity, and references to any process changes to support it.

## Availability

This covers things such as redundancy and disaster recover, as well as anticipated downtime in the event of catastrophic failure. It should cover measures in place to protect against malicious attacks, back-up strategies, etc. If new processes are introduced to support any of this, those should be referenced.

## Serviceability

This should cover how the solution can be supported (documentation and tools provided to help troubleshoot problems), options to extend or interface with the solution, how updates/upgrades are applied, maintenance processes, etc.

## Best Practice Commentary

This should note whether any deviations from proper practice exist as a result of specific technology or customer requirements or constraints.

## Architecture Commentary

This can discuss the potential of the solution to be adapted in future to changing business requirements or evolutions in technology.

# Success Criteria and Test Plan

This section describes how the solution is validated.

## Use Case Validation

This section describes how the use cases (covered earlier) can be tested. It should match the use cases covered earlier.

Example:

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Validation** |
| MV-01 | Preview Spatial Data | When the user selects a region of WA (e.g. South West) and category of spatial data (e.g. Natural Resources) the system displays a map with the corresponding extents and spatial layers.  The Public User can update the information displayed on the map by selecting and/or unselecting public spatial layers.  The user can modify the map view using zoom and pan operations.  The user can create a bookmark for the current map view, which stores a browser bookmark to retrieve the saved map context.  The users can view a saved map context by retrieving a previously created bookmark. |
| MV-02 | Perform Simple Analysis | The custom layer can also be downloaded as a GML document.  The user can display the distance between designated points on the map.  The user can display the longitude and latitude coordinates for designated point on the map.  The user can display the details for a specific map feature, or all features contains within a define bounding-box or circular radius.  The user can retrieve map features that match spatial attribute criteria.  The user can display location information for a designated spatial coordinate.  The user can create a PDF or GIF image of the current map view. |
| MV-03 | Add Spatial Data Layers | The user can include additional spatial layers sourced from an OGC Data Service or spatial database. |
| … | … | … |

## Validation of Solution metrics

This section describes how the solution metrics can be validated. The method of validation must be specific and reproducible. Example:

Example:

|  |  |
| --- | --- |
| Maximum plugin footprint size | 120 – 180k |
| Web application footprint size (total) | 150 – 200k |
| Application load time (first load) | 8 – 10 seconds |
| Application load time (subsequent loads, cached) | 4 – 6 seconds |
| Data refresh times | 1 – 3 seconds |

The above metrics must be met 95% of the time.

## Test Plan

Successful completion of the test plan is the basis for solution acceptance. Refer to the attached test plan template in the appendix, which must be filled out for each use case. It is a more detailed version of the use case validation, and should cover metric validation.

# Phasing, Roadmap, and Implementation PLan

This section describes how implementation is broken down, and includes phasing, a detailed work breakdown structure and implementation instructions.

## Technical Assumptions

This section describes the technical dependencies, which can include a suitable lab, servers with specific operating systems, appropriate network infrastructure, etc.

## Solution Phases and Technical Milestones

This section describes the phasing at a high level, and milestones (including deployments, UAT testing, go-live, etc.). It should not cover costs/effort. It should line up with any existing WBS or project plan.

## Technical Work BreakDown Structure

This section include a WBS, being a detailed list (with grouping) of all the activities required to deliver the solution. It is more detailed than the solution phasing, and should cover all aspects of development. You may reference or include a project plan, if one exists.

## Implementation Instructions

This section should include clarifying notes to accompany the WBS, for the purpose of assisting developers in delivering the solution. It should reference appropriate documentation where necessary. It should not cover details already covered elsewhere, but should focus on specifics and “gotchas” (whether from a technical or business perspective) which are unlikely to be known to developers.

# Appendix – Test Plans

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | Logging into Virtual Jukebox application | | **Test Case No** | 1 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Open the Virtual Jukebox application | | | The system brings the user to the Login/Sign-up Page. | |  | | | |  |  | | |  |
| 2 | Enter Username and Password | | | The username field is filled and is visible to the user.  The password field is filled, but it in encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 3 | Click the ‘Submit’ button | | | The system shows a message to the user of the successful login and brings the user to the Home Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | Logging into Virtual Jukebox application  (This is for when a host has not created a Virtual Jukebox account) | | **Test Case No** | 2 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Open the Virtual Jukebox application | | | The system brings the user to the Login/Sign-up Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Sign-up’ button | | | The system brings the user to the Sign-up Page. | |  | | | |  |  | | |  |
| 3 | Enter Username, Email, and Password | | | The username field is filled and is visible to the user.  The email field is filled and is visible to the user.  The password field is filled, but it in encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 4 | Click the ‘Next’ button | | | The system brings the user to the ‘Connect Spotify’ Page. | |  | | | |  |  | | |  |
| 5 | Enter Username/Email and Password | | | The username/email field is filled and is visible to the user.  The password field is filled, but it in encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 6 | Click the ‘Connect Spotify Account’ button | | | The system shows a message to the user of the successful login and brings the user to the Home Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | Logging into Virtual Jukebox application  (The host does not remember the password for an existing Virtual Jukebox Account) | | **Test Case No** | 3 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Open the Virtual Jukebox application | | | The system brings the user to the Login/Sign-up Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Reset Password’ button | | | The system brings the user to the Reset Password Page. | |  | | | |  |  | | |  |
| 3 | Enter Email | | | The email field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 4 | Click the ‘Send Email’ button | | | The system shows a message to the user that the email has been sent and bring the user back to the Login/Sign-up page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | Logging into Virtual Jukebox application  (The system cannot find the account in the database) | | **Test Case No** | 4 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Open the Virtual Jukebox application | | | The system brings the user to the Login/Sign-up Page. | |  | | | |  |  | | |  |
| 2 | Enter Username and Password | | | The username field is filled and is visible to the user.  The password field is filled, but it in encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 3 | Click the ‘Enter’ button | | | The system shows a message to the user of the unsuccessful login . | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | Logging into Virtual Jukebox application  (The system finds the hashed password provided and the one stored are different) | | **Test Case No** | 5 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Open the Virtual Jukebox application | | | The system brings the user to the Login/Sign-up Page. | |  | | | |  |  | | |  |
| 2 | Enter Username and Password | | | The username field is filled and is visible to the user.  The password field is filled, but it in encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 3 | Click the ‘Enter’ button | | | The system shows a message to the user of the unsuccessful login. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to create a Virtual Jukebox Session | | **Test Case No** | 6 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Host Session’ button | | | The system brings the user to the Host Session Page. | |  | | | |  |  | | |  |
| 3 | Check Host Location | | | The system sets a location marker of the user’s current location on the map. | |  | | | |  |  | | |  |
| 4 | Enter Name for the session | | | The session name field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 5 | Enter Description for the session | | | The session description field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 6 | Click the ‘Next’ button | | | The system brings the user to step 2 of Host Session Page. | |  | | | |  |  | | |  |
| 7 | Select a playlist | | | The chosen playlist is highlighted. | |  | | | |  |  | | |  |
| 8 | Click the ‘Host Session’ button | | | The system shows a message to the user of the successful hosting of the session and brings the user to the Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to create a Virtual Jukebox Session  (The host’s authenticated Spotify account does not have Spotify Premium) | | **Test Case No** | 7 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Host Session’ button | | | The system brings up a pop up that says that the user cannot create a session without Spotify premium. | |  | | | |  |  | | |  |
| 3 | Click the ‘Okay’ button | | | The system brings the user back to the Home page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to create a Virtual Jukebox Session  (The host has an active session in the database) | | **Test Case No** | 8 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Host Session’ button | | | The system brings up a pop up that says that “A session is already being hosted do you want to close it and start a new one?”. | |  | | | |  |  | | |  |
| 3 | Click the ‘Start New Session’ button | | | The system closes the current session being hosted by the user and brings the user to the Host Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to create a Virtual Jukebox Session  (The host’s device does not have location services enabled) | | **Test Case No** | 9 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Host Session’ button | | | The system brings the user to the Host Session Page. | |  | | | |  |  | | |  |
| 3 | Set Host Location | | | A location marker of where the user has selected on the map is shown. | |  | | | |  |  | | |  |
| 4 | Enter Name for the session | | | The session name field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 5 | Enter Description for the session | | | The session description field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 6 | Click the ‘Next’ button | | | The system brings the user to step 2 of Host Session Page. | |  | | | |  |  | | |  |
| 7 | Select a playlist | | | The chosen playlist is highlighted. | |  | | | |  |  | | |  |
| 8 | Click the ‘Host Session’ button | | | The system shows a message to the user of the successful hosting of the session and brings the user to the Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to create a Virtual Jukebox Session  (The host’s connected Spotify account does not contain any playlists) | | **Test Case No** | 10 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Host Session’ button | | | The system brings the user to the Host Session Page. | |  | | | |  |  | | |  |
| 3 | Check Host Location | | | The system sets a location marker of the user’s current location on the map. | |  | | | |  |  | | |  |
| 4 | Enter Name for the session | | | The session name field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 5 | Enter Description for the session | | | The session description field is filled and is visible to the user. | |  | | | |  |  | | |  |
| 6 | Click the ‘Next’ button | | | The system brings the user to step 2 of Host Session Page. | |  | | | |  |  | | |  |
| 7 | Check the playlist | | | The system shows a pop up to the user that they have no active playlists and asks the user to create a playlist in Spotify. | |  | | | |  |  | | |  |
| 8 | Click the ‘Okay’ button | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Logo, company name  Description automatically generated | | **Date** | **Test Title** | | To allow guests to join a Virtual Jukebox Session  (If theVirtual Jukebox Session is public) | | **Test Case No** | 11 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Select a session from the map | | | The system brings up a pop up with the session’s title and description. | |  | | | |  |  | | |  |
| 3 | Click the ‘Join Session’ button | | | The system brings the user to the Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Logo, company name  Description automatically generated | | **Date** | **Test Title** | | To allow guests to join a Virtual Jukebox Session  (If theVirtual Jukebox Session is out of range of the guest) | | **Test Case No** | 12 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Select a session from the map | | | The system brings up a pop up with the session’s title and description. | |  | | | |  |  | | |  |
| 3 | Click the ‘Join Session’ button | | | The system brings up a pop up saying that the session is out of range. | |  | | | |  |  | | |  |
| 4 | Click the ‘Okay’ button | | | The system brings the user back to Home Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow guests to join a Virtual Jukebox Session  (If theVirtual Jukebox Session is private and the user entered the correct password) | | **Test Case No** | 13 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Select a session from the map | | | The system brings up a pop up with the session’s title, description, and password. | |  | | | |  |  | | |  |
| 3 | Enter Password | | | The password field is filled, but it is encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 4 | Click the ‘Join Session’ button | | | The system brings the user to the Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow guests to join a Virtual Jukebox Session  (If theVirtual Jukebox Session is private and the user entered the incorrect password) | | **Test Case No** | 14 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to the Home Page. | |  | | | |  |  | | |  |
| 2 | Select a session from the map | | | The system brings up a pop up with the session’s title, description, and password. | |  | | | |  |  | | |  |
| 3 | Enter Password | | | The password field is filled, but it is encrypted and not visible to the user. | |  | | | |  |  | | |  |
| 4 | Click the ‘Join Session’ button | | | The system shows a message to the user of the incorrect password and lets the user enter the password again. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to add songs from a Virtual Jukebox Session’s playlist to the session’s song queue. | | **Test Case No** | 15 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Add Song to Queue’ button | | | The system brings the user to The Virtual Jukebox Session-Add Song Page. | |  | | | |  |  | | |  |
| 3 | Select a song | | | The number of credits the user has is reduced by one and the selected song is added to the end of the song que. Then it brings the user back to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to add songs from a Virtual Jukebox Session’s playlist to the session’s song queue.  (If the selected song is already in the song queue) | | **Test Case No** | 16 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Add Song to Queue’ button | | | The system brings the user to The Virtual Jukebox Session-Add Song Page. | |  | | | |  |  | | |  |
| 3 | Select a song | | | The system shows a message to the user, that the song is already in the song queue. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to add songs from a Virtual Jukebox Session’s playlist to the session’s song queue.  (The guest or user does not have any credits) | | **Test Case No** | 17 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Add Song to Queue’ button | | | The system brings the user to The Virtual Jukebox Session-Add Song Page. | |  | | | |  |  | | |  |
| 3 | Select a song | | | The system shows a message to the user, that they do not have enough credits. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to upvote songs currently in a Virtual Jukebox Session’s song queue. | | **Test Case No** | 18 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the thumps up icon for a song | | | The playlist will reorder itself based on the upvotes and the thumps up next to the song will be highlighted. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to upvote songs currently in a Virtual Jukebox Session’s song queue.  (If there are no songs currently in the song queue) | | **Test Case No** | 19 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page and shows a message to the user, that there are no songs in the queue. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to upvote songs currently in a Virtual Jukebox Session’s song queue.  (If the user has already upvoted the selected song) | | **Test Case No** | 20 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the thumps up icon for a song | | | The system will remove the vote from the cache and the thumbs up icon will be not highlighted anymore.  The system will then rearrange the queue based on the votes. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to send messages within a Virtual Jukebox Session. | | **Test Case No** | 21 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Messages’ button | | | The system brings the user to the Messages Page and shows all messages since joining the Virtual Jukebox Session. | |  | | | |  |  | | |  |
| 3 | Click the blank message box | | | The devices keyboard is brought up if the device does not have a physical keyboard. | |  | | | |  |  | | |  |
| 4 | Enter Message | | | The message field is filled and visible to the user | |  | | | |  |  | | |  |
| 5 | Click the ‘Send’ button | | | The system will bring the user back to the Messages page with the updated messages visible. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts and guests to send messages within a Virtual Jukebox Session.  (If the system detects profanity in the typed message) | | **Test Case No** | 22 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 11 | | | The system brings the user to The Virtual Jukebox Session Page. | |  | | | |  |  | | |  |
| 2 | Click the ‘Messages’ button | | | The system brings the user to the Messages Page and shows all messages since joining the Virtual Jukebox Session. | |  | | | |  |  | | |  |
| 3 | Click the blank message box | | | The devices keyboard is brought up if the device does not have a physical keyboard. | |  | | | |  |  | | |  |
| 4 | Enter message | | | The message field is filled and visible to the user. | |  | | | |  |  | | |  |
| 5 | Click the ‘Send’ button | | | The system will show a message stating that profanity has been detected. It will then bring the user back to the Messages Page with the updated messages visible where the profanity is replaced with asterisks. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Date** | **Test Title** | | To allow hosts to close a Virtual Jukebox Session. | | **Test Case No** | 23 | | | | **Ver** | 1.0 | |
| **Tested By** | | |  | | **Platform** | | **Browser** | | | **Connection** | | |
|  | |  | | |  | | |
| **Pass/Fail** | | |  | | **Severity (if Failed)** | | | | | **1-2-3-4** | | |
| **Comments** | | |  | | | | | | | | | |
| **STEP** | **ACTION** | | | **EXPECTED RESULT** | | **ACTUAL RESULT** | | | | **Pass/Fail** | **UA** | | | **Severity** |
| 1 | Complete Test Case No 1 | | | The system brings the user to The Home Page | |  | | | |  |  | | |  |
| 2 | Click the ‘Session’ tab | | | The system will bring the user to the Virtual Jukebox Session Page they hosted. | |  | | | |  |  | | |  |
| 3 | Click the ‘End Session’ button | | | The system brings a pop up that asks the user if they are sure they want to end the session. | |  | | | |  |  | | |  |
| 4 | Click the ‘Yes’ button | | | The system closes the session and brings the user to the Home Page. | |  | | | |  |  | | |  |
|  | **END OF TEST** | | | | | | | | | | | | | |

Artifact Artefact