

**Venue Scraper**

**Software Requirements Specification  
Version 1**

**Team Number:** 4 **Project Manager:** CJ Coronado **Team Members:** Katelyn Doyle, Raegan Durdin, Fatima Zaid, James Dove, Sha He

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**Table of Contents (version 1)**

1. Introduction  
 1.1. Project Objectives  
 1.2. Project Scope  
 1.3. Project Overview  
2. Project Description  
 2.1. Project Features / Functions  
 2.2. User Stories  
 2.3. Use Cases  
 2.4. Project Assumptions and Dependencies  
3. Project Collaboration and Documentation  
4. Project Management  
5. Requirements Specification  
 5.1. Business Requirements  
 5.2. User Requirements  
 5.3. Functional Requirements  
 5.4. Non-Functional Requirements

**1. Introduction**

**1.1. Project Overview**

This project aims to develop a web-based software solution for **Shirley Road Records**, a Durham-based music business, to maximize projected ticket sales for concerts. The solution will automate the data collection process from multiple concert venues, perform ticket sales analysis using regression models, and provide users with a dashboard to manage, view, and analyze the collected data.

Currently, Shirley Road Records manually collects data from concerts they book and other publicly available sources. The company performs regression analysis in R to rank artists based on ticket sales potential. To streamline this process, the project will expand an existing Python web scraper and build an interface that allows the user to manage and analyze data more efficiently.

**1.2. Project Objectives**

1. **Automate Data Collection**: Develop a robust web scraper that collects concert data (e.g., band/artist name, venue, ticket price, and day of the week) from multiple venue websites.
2. **Data Management**: Create a system that organizes, stores, and processes scraped data for use in analysis. This could involve exporting the data to a database or Excel sheet.
3. **Sales Prediction and Analysis**: Integrate R language regression analysis into the system to predict which concerts are likely to have high ticket sales based on historical and newly scraped data.
4. **User Interface**: Design a user-friendly interface that allows users to:
   1. View, manage, and edit collected concert data.
   2. Run analysis tasks to predict ticket sales for different bands and venues.
   3. Display the analysis results in an intuitive manner.
5. **Seamless Integration**: Ensure smooth interaction between the front-end (UI), backend (web scraper and data processor), and analysis tools.
6. **Maximize Ticket Sales**: Provide actionable insights for concert booking decisions by identifying bands and venues that perform well in terms of ticket sales.

**1.3. Project Scope**

The scope of this project includes:

1. **Development of Multi-Venue Web Scrapers**:
   1. Extend the existing web scraper to collect data from multiple concert venues.
   2. Ensure flexibility to easily add or modify scraping for new venues as needed.
   3. Data fields to be collected include band/artist, venue, ticket price, and date.
2. **Backend Data Processing**:
   1. Data normalization and formatting for easy insertion into an Excel sheet or database.
   2. Implement a system to process and prepare scraped data for analysis.
3. **R Integration for Analysis**:
   1. Use R for regression analysis on the data to predict ticket sales potential.
   2. Create a workflow where users can trigger R scripts to run analysis on selected data.
4. **User Interface Development**:
   1. Build a front-end dashboard where users can view, manage, and analyze data.
   2. Display sales projections and analysis results in a clear, intuitive format (e.g., charts, graphs, ranked lists).
5. **System Testing and Validation**:
   1. Validate that the web scraper collects accurate and complete data.
   2. Ensure the system integrates correctly with the R analysis tools.
   3. Perform performance testing to ensure the UI and data analysis processes are smooth and user-friendly.

**Excluded Features**

1. **User Login and Access Management**: No need to develop complex user management functionality.
2. **Advanced Recommendation System**: Only basic filtering and sorting of data are supported, without complex AI-based recommendation algorithms.
3. **Mobile Platform Support**: The project will focus on web development, without mobile platform support.
4. **Internationalization Support**: The project will only support English and will not include multi-language versions.

**Project Constraints and Assumptions**

1. **Time Constraints**: The project will be completed within four sprint cycles.
2. **Technical Requirements**:
   1. Ensure that the scraper tool can bypass anti-scraping mechanisms to reliably collect data.
   2. The database must support efficient data querying and filtering for sales prediction analysis.
3. **Dependencies**:
   1. The project depends on third-party websites (venue websites) for data. Changes in these sources may affect project progress.

This software will streamline decision-making processes, allowing Shirley Road Records to optimize concert bookings and maximize ticket sales through a data-driven approach.

**2. Project Description**

**2.1. Project Features / Functions**

1. User Login and Access Management
2. Venue Search and Filtering Functionality
3. Data Scraping and Automatic Updates
4. Database Management
5. User Interface
6. Backend Development and API Design

**2.2. User Stories**

**2.2.1.** “As a booking agent, I want an easily accessible list of venues and their performance history so that I can book adequate matches between bands and venues.”

**2.2.2.** “As a venue owner involved in the booking process, I want an easily accessible database of ticket performance history so that I can make educated comparisons based on surrounding and past shows in certain area.”   
  
**2.2.3.** “As a venue owner involved in ticket sales, I want a website that allows for easy ticket creation, band promotion, and ticket sales comparisons so that I can make efficient sales and promotion for upcoming shows.”

**2.3. Use Case**

**2.3.1.** Use Case 1: View List of Scraped Venues

* Actor: User (Visitor)
* Goal: View a list of venues already scraped by the system.
* Steps:
* User visits the web scraper application’s homepage.
* System displays a list of venues that have already been scraped.
* User can filter the list by venue type, location, or event date.
* System updates the venue list based on the filters selected by the User.
* User clicks on a specific venue to view more detailed information (e.g., address, upcoming events).
* System presents the detailed venue information to the User.

**2.3.2.** Use Case 2: Search for a Venue

* Actor: User (Visitor)
* Goal: Search for a specific venue from the list of scraped venues.
* Steps:
* User accesses the search bar on the web scraper application.
* User inputs the name of a venue or keywords (e.g., "concert hall" or "outdoor space").
* System searches the scraped venue database for matching results.
* System displays the list of matching venues to the User.
* User clicks on a venue to view detailed information.
* System presents detailed information (e.g., capacity, events, contact info) about the venue.

**2.3.3.** Use Case 3: System stays up-to date

* Actor: User (Application)  
  Goal: Update system with most up to date information
* Steps:
* Option a:
* Customer logs into the system.
* Customer views home page.
* Scraper runs and pulls in the most up to date information.
* Option b:
* User refreshes the page.
* Scraper runs and pulls in the most up to date information.
* Option c:
* After a certain amount of time has passed, Scraper runs and pulls in the most up to date information.

**2.4. Project Assumptions and Dependencies**

Assumptions:

* Team developers will be available for all portions of the developmental process
* All data provided will be accurate and substantial.
* All sprint deadlines will be met.

External Dependencies:

* Venues providing data dependent for populating data base.
* Band providing data dependent for populating data base.
* Program/language functionality dependent for code development.

**3. Project Collaboration and Documentation**

* Microsoft Teams
  + Task tracking: team lead sends message about tasks, members clarify with him
  + Virtual meetings: meet weekly to discuss our progress, organization, and goals
* Google Drive
  + File storage: hold assignments, documentation, notes, and research for each week’s tasks
* Blackboard Drive
  + Assignment storage: information, videos, and training from instructors to reference and clarify to facilitate our work
  + Groups: communicate with our group members about the project
* Text messaging
  + Quick communication: updates from team lead, quick updates about scheduling and submission progress
* GitHub / Git
  + Collaborative coding: we can edit the code in separate branches and merge them
  + Coding version history: we can always track our previous code to reference for any changes and updates

**4. Project Management**

We will be using the Agile Methodology. Each part of the project will be broken up into different sprints that focus on a different part of the project. An example of this could be Sprint 1, focusing on the basic creation of the website. Sprint 2 on error handling, and Step 3 can be for applying to a broader use. We will hold a weekly meeting to go over our goals and what has been accomplished in the previous week, as well as constant updates on what everyone is currently working on. This way the project can stay organized, and the group can stay focused. A tool that will be used is called Jira, it is used to help plan, organize, and help everyone stay on top of the project.

**5. Requirements Specification  
5.1. Business Requirements**  
  
High Level Business Needs:  
1. Event Venue Discovery. The application should gather and display information about specific venues helping users find venues that are suitable for their needs.

2. Accurate Information. This information must be up to date with information about each venue, availability, pricing, etc.

3. Filter Down Information for Decision Making. Users should be able to use filters to help narrow down their options to help when choosing a venue for their specific needs.

Business Requirements:  
1. Must- Accurately collect details and display them.

2. Should- Have a working filtering feature for easy searching.

3. Could- Search through venues in a defined area.

4. Will not- Be able to sign up for a venue, just links to where they should sign up for a specific venue.

**5.2. User Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| UR1 | Users must be able to create an account and log in securely using their email and password. | M |
| UR2 | Users must be able to save venues and store them in a storage library. | C |
| UR3 | Users must be able to search venues and filter based on desired criteria. | S |
| UR4 | Users must be able to pull and compare performer’s  Ticket sale Data at specific venues. | M |
| UR5 | Users must be able to develop promotional material and create tickets. | W |

**5.3. Functional Requirements**

Detail the specific behaviors, functions, and capabilities that the application must provide.

These requirements are derived from user and business requirements and outline how the

application will fulfill those needs. List at least 3 functional requirements with MOSCOW code.

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **MOSCOW** |
| **FR1** | The application should store data for lots of details like band, artist, venue, price, day of week | M |
| **FR2** | The application should maximize projected concerts ticket sales based on regression calculations (Project Goal) | M |
| **FR3** | The application should provide authentication mechanisms to ensure secure access for the users handling this data | M |
| **FR4** | The application should show each piece of data for the user to easily view | M |
| **FR5** | The application should have permissions so some users can view the tools, and others can edit | S |
| **FR6** | The application could have audit tracking to be able to monitor any updates in the database or to the systems that can affect calculations | C |

**5.4. Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Description | MOSCOW |
| NFR1 | The tool must be able to scrape multiple websites simultaneously. | M |
| NFR2 | The tool must be able to handle a growing userbase should the stakeholder decide to transition the tool to a website. | M |
| NFR3 | The interface should be user-friendly and intuitive to navigate. | S |
| NFR4 | The program could have robust security for future use as a website, to prevent the stakeholder’s data from being acquired by competitors. | C |