

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**

**Table of Contents**

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 Objectives

< Define the primary goals of the software being developed and set clear expectations  
for what the system aims to achieve.>

1.2. Project Scope  
< Define the boundaries of the project, including what is included and excluded. >

1.3. Project Overview  
< Describe the project's goals, the steps to achieve these goals, and the expected  
outcomes. Outline the project schedule, necessary resources, and status.>

2. Project Description

This project is about editing / creating a venue scraper which will scrape multiple homepages for venue websites based in Raleigh / Durham / Cary.

2.1. Project Features / Functions  
< List at least 3 main features and functionalities to be developed.>  
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  
< Simple, concise description of a feature or requirement from the perspective of an end  
user. User stories are often used in Agile development methodologies to capture user  
needs and facilitate communication between stakeholders. List at least 3 user story  
statements to briefly describe the type of user, what they want and why. >  
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  
< Describes a specific interaction between a user (or "actor") and the application to  
achieve a particular goal. It outlines the steps involved in this interaction and can help to  
clarify functional requirements. Use cases are especially useful in understanding how  
users will interact with the application and are typically presented in a structured format.  
List at least 3 user cases to describe a set of interactions between a system and one or  
more actors. >  
2.3.1. Use Case 1 (either Use Case Diagram or Use Case Specification)  
2.3.2. Use Case 2 (either Use Case Diagram or Use Case Specification)  
2.3.3. Use Case 3 (either Use Case Diagram or Use Case Specification)

--------------------------

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.

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.

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  
< Outlines the factors that are assumed to be true for the project and the external elements  
that the project depends on. These assumptions and dependencies help manage risks by  
identifying potential challenges or factors that could impact the project’s success.>

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  
< Specify what collaboration and documentation tools / platforms that will be used in this  
project development and how your group will use these tools / platforms.>

4. Project Management  
< Specify what project management methods and tools will be used in this project  
development and how your group will use these methods and tools. >

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  
< Outline the high-level needs of the business or organization that the application must  
fulfill. List at least 3 business requirements with MOSCOW code.>  
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  
< Describe what the end users need or expect from the application. These requirements  
focus on the functionality and usability aspects of the software from the user's perspective.  
They outline what the application should enable users to do in order to achieve their goals.  
List at least 3 user requirements with MOSCOW code.>

| 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.>  
  
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 |