Software Requirements Specification

for

Class Artifacts

Version 1.1 – Awaiting approval

Prepared by Lipinski, N., Odom, Q., Tate, I., Vanderkolk, C.

CSCI491 Class Artifacts Group

November 20, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Cole Vanderkolk	11/20/2025	Removing superfluous and inaccurate information, fleshing out remaining sections	1.1
Nick Lipinski	11/20/2025	Removing superfluous and inaccurate information, fleshing out remaining sections	1.1
Quincy Odom	11/20/2025	Removing superfluous and inaccurate information, fleshing out remaining sections	1.1

1. Introduction

1.1 Project Title

Class Artifacts will be the working title of the project, subject to change

1.2 Project Roles

1.2.1 Stakeholders:

- CS Chair (Filip Jagodzinski)
- CS Exec Committee (represented by Filip Jagodzinski)
- Project maintainer (Kate Panter)

1.2.2 Clients:

- Filip Jagodzinski (primary user)
- Kate Panter (primary maintainer)
- Jesse Atkins (represented by Kate Panter)

1.2.3 Mentors:

- Kate Panter (primary)
- Blake Pedrini
- Filip Jagodzinski

1.3 Purpose

Class Artifacts is primarily meant for the CS Department Chair (Filip), and CS Exec (a department committee), in helping them to reason about class scheduling. Class Artifacts will interface with Banner to access historical class data, maintaining an accessible and well-organized record of which classes have been offered in past terms and which faculty were responsible for facilitating them. The Department Chair and CS Exec committee will be able to view statistics about class offerings, seats filled and available, and student demographics (premajor vs in-major vs in-major elective) in order to be better informed when deciding which classes to offer in future quarters.

1.4 Document Conventions

Requirement Priorities - Priorities for higher-level requirements are inherited by detailed requirements unless specified. Priority levels (High, Medium, Low) will be given if relevant.
Text Formatting - There is no specific text formatting to be aware of currently, the document should be read as is. If future formatting is added this will change.

Document language reference:

- CS – Computer Science
- WWU – Western Washington University
- CS Exec committee – Computer Science Executive Committee

1.5 Intended Audience and Reading Suggestions

Primary Audience:

- Computer Science Department Chair and Executive Committee – Key stakeholders and decision-makers who will evaluate the project's alignment with departmental needs and approve the final product.

Secondary Audiences:

- Development Team – Software developers, database administrators, and technical contributors responsible for implementing the system according to specified requirements.
- Project Stakeholders – Faculty members, administrative staff, and other parties with vested interest in the system's functionality and success.
- Quality Assurance/Testers – Individuals responsible for validating that the system meets specified requirements and performs as expected.

1.6 Product Scope

"Class Artifacts" is a software capable of analysing all previous WWU CS class enrollments on a general scale. The project will have the ability to present previous classes: Enrollment numbers, Professors, Class numbers, Class rooms, CRN's, and any other information provided within the banner that would be relevant for class projection and information capture. The Purpose of this is to allow the relevant clients and all potential users to research information about past classes, and make more informed decisions about future CS classes. The goal of the project is to create a well designed and user friendly web application that interfaces with WWU's Banner to display all relevant information about selected or searched classes. The function of this is to address concerns like enrollment numbers and which classes should potentially be altered in order to maximize efficiency in effectively teaching a course in the future.

The goal of Class Artifacts is not to create "Classfinder 2.0"; usage by the university at large or by students or general faculty is out of scope. This tool will not be used to create individual schedules for students or to examine future class offerings, it is solely designed to reference historical class data.

1.7 References

Project README - Contains relevant project information including setup instructions and development status, updated continuously throughout the project lifecycle.

User Manual - Provides end users with guidance on navigating and utilizing the system's features.

Technical Manual - Details system architecture, database schema, API specifications, and deployment procedures for developers and technical staff.

2. Overall Description

2.1 Product Perspective

Class artifacts is a new self contained product, it is taking inspiration from Classfinder with abilities such as filtering, sorting, and viewing class details across quarters. The overall system will operate for the functionality of exclusively the Computer Science department as a referential system to create and plan course schedules over the span of a school year. The product does not replace any current functional interfaces (aka Browse Classes) or previous interfaces (not a replacement for the original Classfinder) and is to be used exclusively within the department.

2.2 Product Functions

- Advanced Class Search – Users can search for classes by course number, instructor, time, days, credits, or class length.
- Filter and Sort Options – Results can be filtered or sorted by attributes like availability, meeting days, or start time.
- Web Access – Users can access the database online, and don't need to install any files locally
- Responsive and Accessible Interface – Works smoothly on desktop, following accessibility standards for all users. Features are intuitive to understand at a glance and it allows users to accomplish their tasks as efficiently as possible.

2.3 User Classes and Characteristics

Computer Science Department head and executives:
- Needs to have permanent access to the database, since they are the primary stakeholders (most important)

Developers:
- Should have database access during development, but access should be revoked once finished. Developers require the ability to inspect database information and utilize it for testing.

The Client:
- The client is not a primary user of the system, but rather a maintainer. Therefore, they need permanent access to the database.

2.4 Operating Environment

The database itself will be hosted on the Project Lab within the Computer Science building through Banner. The usage of the project will be done through the web, so any device capable of accessing the internet would be able to use it.

2.5 Design and Implementation Constraints

- Only covers the Computer Science department
- Must be a website, not an installed app
- Must be able to access data located in Banner database
    - No requirement to interface with any other applications
- Currently no specific hardware limitations
    - Potentially will be hosted on projectweb (pending approval by Kate Panter)
- Currently no specific language requirements
- Communication protocols:
    - must be accessible by web browser
    - must be able to communicate with Banner
- Security considerations:
    - Should only be accessible by department chair and exec committee
- Software to be maintained by Kate Panter
- Design format:
    - Will have a simple UI with query fields, dropdown menus, and check boxes
    - "Uncluttered"

2.6 User Documentation

- There will be a readme that gets uploaded to the github repo and will be accessible by the users as a .txt file
- There will be a User Guide .doc file that is accessible via download from the website

2.7 Assumptions and Dependencies

- Need to be able to interface with Banner
- Need to find somewhere to host the program (can't be a local install)
- Need to get database information with classes

3. External Interface Requirements

3.1 User Interfaces

- The screen will be displayed using basic HTML, CSS, and JAVA website mechanics
- The screen will display the WWU name and logo
- There will be buttons and dropdown menus very similar to classfinder 1.0
    - Class start/end times (selectable)
    - Class days (selectable)
    - College (dropdown)
    - CRN (Input)
    - Professor (Input)
    - Class # /name (input)
    - Location (dropdown)
    - Waitlisted/available (selectable)
- The screen will also have the ability to display a made schedule with the selected classes

- The website will be able to also reference past classes and display them in the same manner as all the listed above features

3.2 Hardware Interfaces

- Only exists on the cloud, no hardware needed

3.3 Software Interfaces

- Our software will need two main connections, one through the internet via any wifi capable device in order to interface through the website and connect with the database macOS, MS, or Linux, the database is (assumedly) stored on campus and only accessible through the (TBD) given to us to format the information into the website.

3.4 Communications Interfaces

- Required to connect to banner
- Runs in a web browser
- Ability to fetch and use multiple years worth of data from the database

4. System Features

4.1 Historical Class Data

4.1.1 Description and Priority

Database of all past class offerings, which quarters they were available, total enrollment in each class, and which faculty taught them. Lower priority feature: indicate enrollment of pre-major students, in-major required classes, and in-major electives

4.1.2 Stimulus/Response Sequences

Enter search terms (year, quarter, faculty, course number) → Return list of results with quick info (course name, instructor, date, seats available/filled) → User selects individual course offering → Navigates to a page with all available information and statistics about that offering

4.1.3 Functional Requirements

REQ-1:	Web hosting service to store database info
REQ-2:	UI to navigate, enter queries, and display results
REQ-3:	Security to prevent unauthorized access to data

4.2 UX/UI

4.1.1 Description and Priority

- Home page:
  - Title (Class Artifacts)

    - query fields (year, quarter, faculty, course number)
    - “Search” button
    - “Download User Guide” button
    - “Help” button
        - displays popup window with page navigation information
            - How to use search feature
            - How to find detailed results
            - How to return to homepage
- Results page:
    - Title (Class Artifacts)
    - list of results with all available information and statistics about that offering
    - “Back to search” button
    - “Detailed Results” popup window:
        - contains all available information and statistics about that offering
            - course name
            - premajor class? boolean
            - major locked? boolean
            - instructor(s)
            - date (which year/quarter this selection was offered)
            - seats available/filled
            - sections offered that quarter
            - number of times offered total
            - Which faculty have taught this class in past quarters

4.1.2 Stimulus/Response Sequences

- User can click in query fields, and type text into them
- User can click “search” button – will navigate to Results page
- User can select individual results to generate “detailed results” popup window

4.1.3 Functional Requirements

REQ-1: Home page that has query fields and search button
REQ-2: Results page that has result field, popup windows, and return button
REQ-3: Popup windows that contain data gathered from Banner database

4.3 Security

4.1.1 Description and Priority

Requires a username/password to access, entered in login fields on home page

4.1.2 Stimulus/Response Sequences

User enters username/password → presses “log-in” button → system validates, navigates to home page if valid, returns “bad username or password” if invalid

4.1.3 Functional Requirements

REQ-1: User login popup with fields for username and password and log-in button
REQ-2: Validation – check username/password against table of valid users

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Should return results quickly, but not anticipated to be accessed by more than a handful of people at any one time (don't need to worry about thousands of students trying to register simultaneously)

5.2 Safety Requirements

N/A

5.3 Security Requirements

Should prevent access to anyone not authorized to view the data. Could require username/password to log in, and block users after multiple failed login attempts.

5.4 Software Quality Attributes

Highest priority: Correctness, Maintainability, Reliability, Usability
Lowest priority: Flexibility, Portability

5.5 Business Rules

Only authorized users will have any access to the product at all. All users will have the same level of authorization.

6. Other Requirements

Interfaces with Banner database to obtain data about relevant classes

Appendix A: Glossary

N/A (for now)

Appendix B: Analysis Models

N/A (for now)

Appendix C: To Be Determined List

- Need to determine exactly where the site will be hosted (potentially projectweb)
- Need to determine exactly how to access data on Banner
- Need to determine how database will actually interface with website
- Need to determine if website is online publicly or just accessible to WWU employees
- Need to determine scope of classes that will be available

- Need to determine security, what information will be available in the database and therefor what levels of security we must ensure for the website
- Need to determine what times we will be accessing, from sound of it possibly just a past to present tool, not used for future quarters
- Need to determine level of CSS and “Prettiness” of website, we have conflicting answers
- Need to determine how to interface with database effectively and quickly