Software Requirements Specification

for

Class Artifacts

Version 1.3 – Awaiting approval

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

CSCI 491, 492, 493 Class Artifacts Group

December 9, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Cole Vanderkolk, Nick Lipinski, Quincy Odom	11/20/2025	Removing superfluous and inaccurate information, fleshing out remaining sections	1.1
Cole Vanderkolk, Nick Lipinski, Quincy Odom	11/29/2025	Adding Gantt Chart for timeline (Cole), adding wireframe (Quincy), making corrections and implementing feedback to SRS (Nick)	1.2
Nick Lipinski	12/9/2025	Update all sections based on feedback from Kate and Filip, revise Banner interface sections to indicate we will pull once per quarter and store the data locally for regular queries	1.3

1. Introduction

1.1 Project Title

1.1.1 Class Artifacts will be the working title of the project, subject to change.
1.1.2 The working acronym for the project will be CAR (Class Artifacts), subject to change.

1.2 Project Roles

1.2.1 Stakeholders:
- CS Chair
- CS Exec Committee
- Project maintainer

1.3 Purpose

1.3.1 Class Artifacts is primarily meant for the CS Department Chair (Filip), and CS Exec (a department committee, represented by Filip), in helping them to inform them about class scheduling using historical data. 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 in order to be better informed when deciding which classes to offer in future quarters.

1.4 Document Conventions

1.4.1 Requirement Priorities - Priorities for higher-level requirements are inherited by detailed requirements unless specified. Priority levels (High, Medium, Low) will be given if relevant.
1.4.2 Document language reference, bold text for acronym explanations:
- CAR – **C**lass **Ar**tifacts
- CS – **C**omputer **S**cience
- DB – **D**ata**b**ase
- WWU – **W**estern **W**ashington **U**niversity
- CS Exec committee – **C**omputer **S**cience **Exec**utive Committee
    - The CS Exec committee helps the CS Chair determine which class offerings will be made available in future quarters.

1.5 Intended Audience and Reading Suggestions

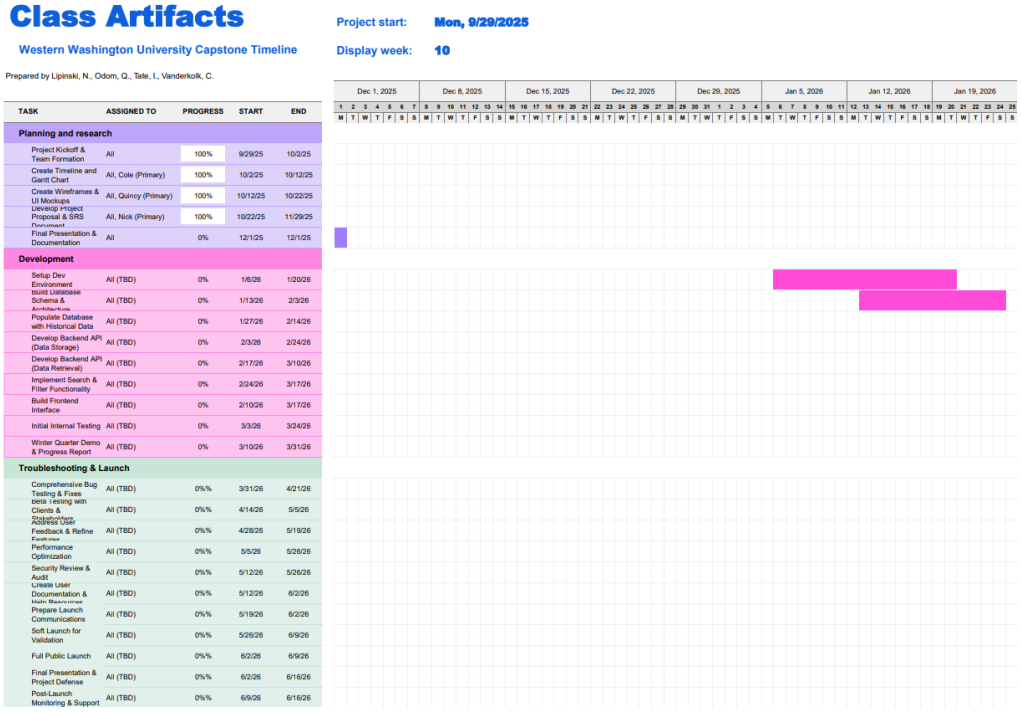
1.5.1 Primary Audience:
- Project maintainers - These users will be responsible for the continued maintenance of Class Artifacts.

1.5.2 Secondary Audiences: Development Team –

- Software developers, database administrators, and technical contributors responsible for implementing the system according to specified requirements.
- Project Stakeholders – any stakeholders with vested interest in the system's functionality and development.
- Quality Assurance/Testers – Individuals responsible for validating that the system meets specified requirements and performs as expected.

1.5.3 Reading suggestion:

- Deliverables Timeline (next 8 weeks)



Class Artifacts

Western Washington University Capstone Timeline

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

Project start: Mon, 9/29/2025

Display week: 10

Dec 1, 2025 | Dec 8, 2025 | Dec 15, 2025 | Dec 22, 2025 | Dec 29, 2025 | Jan 5, 2026 | Jan 12, 2026 | Jan 19, 2026

TASK	ASSIGNED TO	PROGRESS	START	END
Planning and research				
Project Kickoff & Team Formation	All	100%	9/29/25	10/2/25
Create Timeline and Gantt Chart	All, Cole (Primary)	100%	10/2/25	10/12/25
Create Wireframes & UI Mockups	All, Quincy (Primary)	100%	10/12/25	10/22/25
Develop Project Proposal & SRS Document	All, Nick (Primary)	100%	10/22/25	11/29/25
Final Presentation & Documentation	All	0%	12/1/25	12/1/25
Development				
Setup Dev Environment	All (TBD)	0%	1/6/26	1/20/26
Build Database Schema & Architecture	All (TBD)	0%	1/13/26	2/3/26
Populate Database with Historical Data	All (TBD)	0%	1/27/26	2/14/26
Develop Backend API (Data Storage)	All (TBD)	0%	2/3/26	2/24/26
Develop Backend API (Data Retrieval)	All (TBD)	0%	2/17/26	3/10/26
Implement Search & Filter Functionality	All (TBD)	0%	2/24/26	3/17/26
Build Frontend Interface	All (TBD)	0%	2/10/26	3/17/26
Initial Internal Testing	All (TBD)	0%	3/3/26	3/24/26
Winter Quarter Demo & Progress Report	All (TBD)	0%	3/10/26	3/31/26
Troubleshooting & Launch				
Comprehensive Bug Testing & Fixes	All (TBD)	0%%	3/31/26	4/21/26
Beta Testing with Clients & Stakeholder	All (TBD)	0%%	4/14/26	5/5/26
Address User Feedback & Refine Features	All (TBD)	0%%	4/28/26	5/19/26
Performance Optimization	All (TBD)	0%%	5/5/26	5/26/26
Security Review & Audit	All (TBD)	0%%	5/12/26	5/26/26
Create User Documentation & Help Resources	All (TBD)	0%%	5/12/26	6/2/26
Prepare Launch Communications	All (TBD)	0%%	5/19/26	6/2/26
Soft Launch for Validation	All (TBD)	0%%	5/26/26	6/9/26
Full Public Launch	All (TBD)	0%%	6/2/26	6/9/26
Final Presentation & Project Defense	All (TBD)	0%%	6/2/26	6/16/26
Post-Launch Monitoring & Support	All (TBD)	0%%	6/9/26	6/16/26

- Full deliverables timeline available here:
  - https://wwu2-my.sharepoint.com/:x:/g/personal/vande9_wwu_edu/EdnPIueLONhOjf0yygfL2UkBiJ9IM3XLbPFTn5I7Xgy9wA?rtime=5cjlpHYw3kg&nav=MTVfezAwMDAwMDAwLTAwMDEtMDAwMC0wMDAwLTAwMDAwMDAwMDAwMH0

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, which are found within Banner. 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 intended users of Class Artifacts are the CS Chair and Exec Committee, not the university at large. Usage 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.

2. Overall Description

2.1 Product Perspective

Class artifacts is a new self contained product, it has 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 (ie Browse Classes) and is to be used exclusively within the Computer Science department.

2.2 Product Functions

- Advanced Class Search – Users can search for classes by course name, course number, instructor, time, days, credits, or class length. Users can search for all classes offered during specific quarters and/or specific years.
- Filter and Sort Options – Results can be filtered or sorted by attributes like availability, meeting days, start time, enrollment, total seats available, pre-major vs in-major required vs in-major elective status. Classes that exist but are not offered that quarter can also be displayed.
- Users can view all extant class data for any course that exists in the Banner database.
- Web Access – Users can access the database online, and don’t need to install any files locally
- Responsive and Intuitive Interface – Works smoothly on desktop. 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 chair and executive committee:

- Needs to have permanent access to the database, since they are the primary users.

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.

System Maintainer:

- The system maintainer's primary role in system usage is to control user access to the system along with developing and approving patches if the system requires updating.

2.4 Operating Environment

The database itself will be hosted on the Project Web within the Computer Science department. Data will be copied from Banner on a quarterly basis. Queries sent through the program will return results from the local database and will not need to access 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.
The database will be accessed via an interface that submits real-time direct database queries, and cache data locally.

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
- Currently no specific language requirements
- Communication protocols:
    - must be accessible by web browsers (Chrome, Firefox, Edge)
    - must be able to communicate with Banner
- Security considerations:
    - Should only be accessible by department chair and exec committee
- Software to be maintained by System Maintainer
- Design format: Will have a simple UI with query fields, dropdown menus, and check boxes

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 .pdf file that is accessible via download from the website
- The user guide will contain:
    - Instructions on how to access CAR
    - Information on CAR's capabilities and intended use
    - How to contact the current project maintainer(s) to report bugs

2.7 Assumptions and Dependencies

- Interface with Banner
    - Be able to pull relevant info once per quarter
    - Have access to these fields:

        - Course number
        - Course Description
        - Class Details
        - Prerequisites
        - Detailed prerequisites
        - Restrictions
        - Location/Time/Instructor
        - Enrollment/Waitlist
        - Corequisites
        - Credits
        - Instructor
        - CRN
        - Meeting Times
        - Term (year and quarter)
        - Delivery
        - Seats available/filled
        - Number of sections
        - Attribute
        - Status
        - Cross listed courses
        - Catalog
- Program hosted on projectweb (can't be a local install)
    - Database containing info cloned from Banner will reside on projectweb
        - Program will access cloned DB rather than query Banner for day to day use, and access Banner one time each quarter to update the local DB
- 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
    - Class start/end times (selectable)
    - Class days (selectable)
    - CRN (Input)
    - Professor (Input)
    - Class # /name (input)
    - Location (dropdown)
    - Waitlisted/available (selectable)
- The website will be able to reference all past classes in the CS department and display them in the manner as listed above features

3.2 Software Interfaces

- Our software will need to run on any major operating system (macOS, MS, or Linux)
- It will need to run on any major browser (Chrome, Edge, Firefox)
- Will utilize a Power BI dashboard
- It will need an interface to access Banner and copy historical class data from Banner to the local database once per quarter
- It will need to interface with the local database (of data cloned from Banner) stored in projectweb on an as-needed basis to provide functionality for normal operations and return results from queries

3.3 Communications Interfaces

- Required to connect to banner and have permission to pull data once per quarter
- Connect to a local database with cloned data on projectweb, accessible on an as-needed basis
- Runs in a web browser
- Ability to fetch and use multiple years worth of data from the database
- Will use a Power BI dashboard

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

1. Enter search terms (year, quarter, faculty, course number)
2. Return list of results with quick info (course name, instructor, date, seats available/filled)
3. User selects individual course offering
4. 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: Uses WWU login to prevent unauthorized access to data

4.2 UX/UI

4.2.1 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
    - A visual indicator for each item that shows which quarter a class is part of (Fall, Winter, Spring, and Summer)
    - "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

4.2.2 Stimulus/Response Sequences

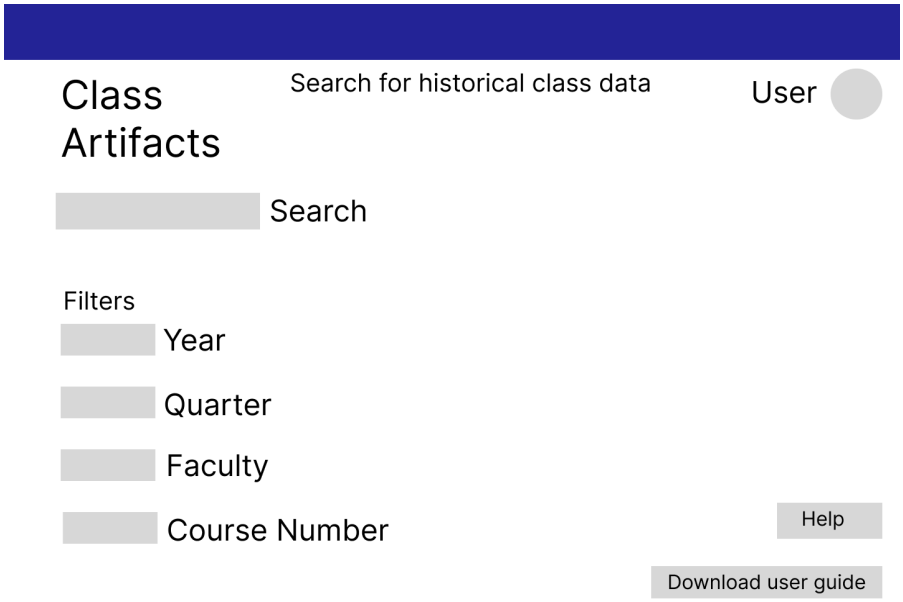
- User can click in query fields, and type text into them
- User can click "search" button – will navigate to Results page
- User can select individual results to generate "detailed results" popup window

4.2.3 Functional Requirements

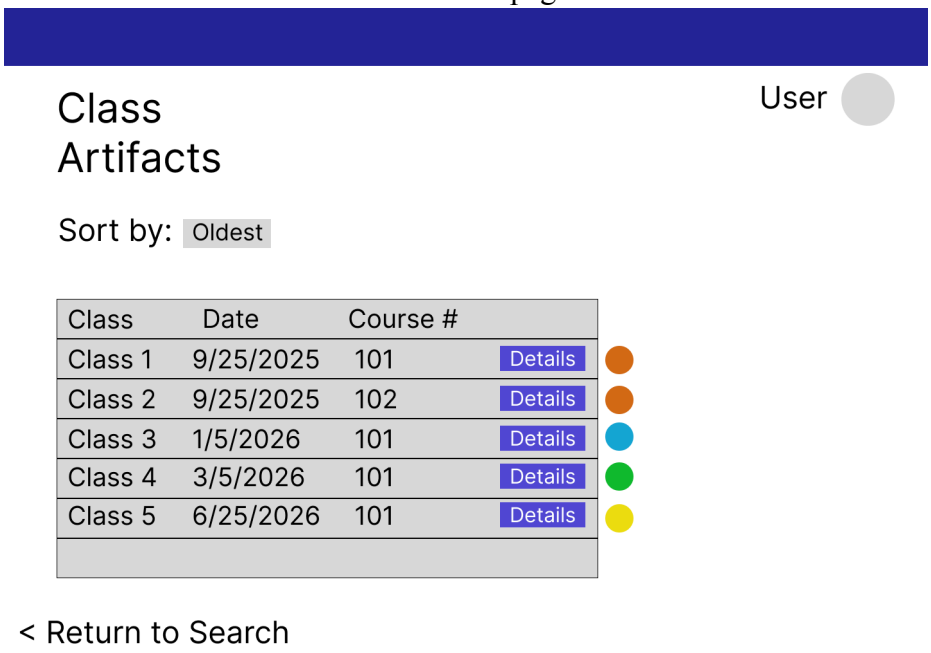
REQ-1: Home page that has query fields and search button
REQ-2: Results page that has result field, popup windows, and return button
REQ-3: Popup windows that contain data gathered from projectweb database

4.2.4 Wireframes

Class Artifacts home page wireframe



Class Artifacts results page wireframe



5. Other Nonfunctional Requirements

5.1 Performance Requirements

Return results quickly, but not anticipated to be accessed by more than a handful of people at any one time. Queries should return results in five seconds or less under normal operating conditions.

5.2 Security Requirements

The project will use WWU's SSO (Single Sign-On) login credentials to prevent access to anyone not authorized to view the data. Require username/password to log in, and block users after multiple failed login attempts. Database will be securely stored on project web in the CS department.

5.3 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

Utilize a Power BI dashboard that is connected to our database
Interfaces with Banner database to obtain data about relevant classes
Interfaces with projectweb to store cloned database and provide responses to queries

Appendix A: To Be Determined List

- Need confirmation on exactly where the site will be hosted (projectweb)
- Need to determine exactly how to access data on Banner
- Need to determine exactly how Power BI dashboard will interface with database and Banner
- Need to determine if website is online publicly or just accessible to WWU employees
- Need to determine how to interface with WWU login to provide security, what information will be available in the database and therefore what levels of security we must ensure for the websites
- 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