UW Class Planner: Sprint 2 Report

Krishna Langille
Division of Computing & Software Systems, University of Washington Bothell
CSS 497: Capstone
May 1, 2022

Table of Contents

Introduction

In Sprint 2, the focus was on the database section of the application. This means all my artifacts and work was database related. This report will detail the artifacts of the sprint including diagrams, updated personas and requirements, and progress.

Progress

During this sprint the following parts of the project were progressed:

Azure services:

The SQL database was properly set up with a server behind it. This included creating the tables and columns inside the database that will hold the rows of data. The only issue with the services so far is the budget, but some adjustments were made to make it a bit cheaper at the cost of performance.

Diagrams:

Since this sprint was about databases, I created a logical database diagram using the relational model. This was mostly due to the fact that this was one of the key diagrams I have learned so far in the database systems course. I may revise the one I made for this project in the next sprint since I've already found out ways to improve this model. This may be disregarded though because of time constraints.

Prototype:

Some elements that were absent in the initial prototype like actually picking courses were added. This element and others will be added in the next sprint. Other than that, some of the text was altered with respect to the sponsor feedback. This will be touched on in more detail in a later sprint.

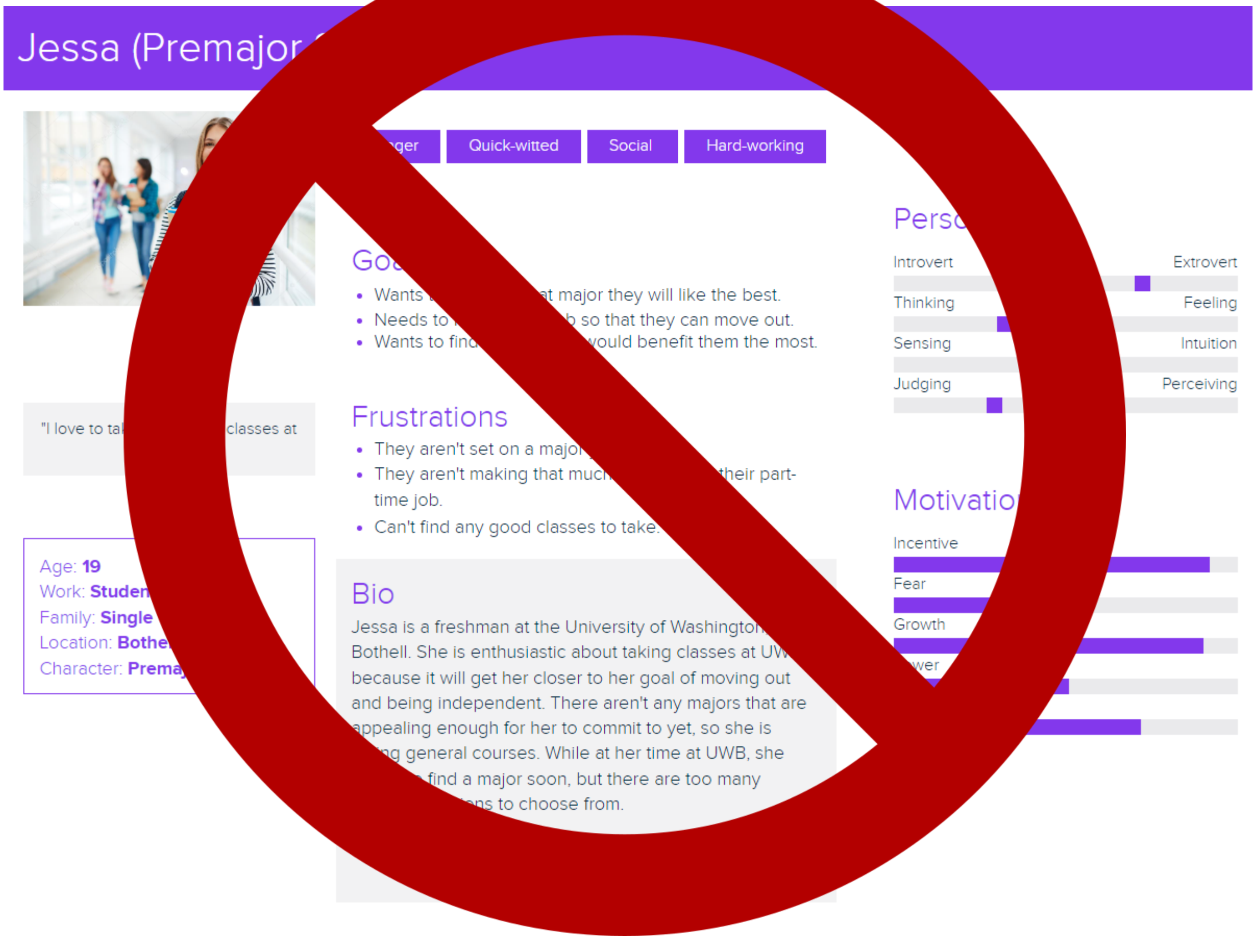
Database:

The SQL database was filled during this sprint and was properly configured to suit the applications needs. The data in the database will likely be deleted and replaced, but the initial data is here for testing purposes when starting the algorithm next sprint. There were many versions of the database during this sprint, but the version seen for this sprint is a bit different from the design. But, the use of the database and how it is used is primarily the same.

Updated Artifacts

Personas:

Premajor Student



Jessa (Premajor

ger Quick-witted Social Hard-working

Pers

Introvert Extrovert

Thinking Feeling

Sensing Intuition

Judging Perceiving

Go

- Wants t at major they will like the best.
- Needs to b so that they can move out.
- Wants to find would benefit them the most.

"I love to ta classes at

Frustrations

- They aren't set on a major
- They aren't making that much their part-time job.
- Can't find any good classes to take.

Motivatio

Incentive

Fear

Growth

wer

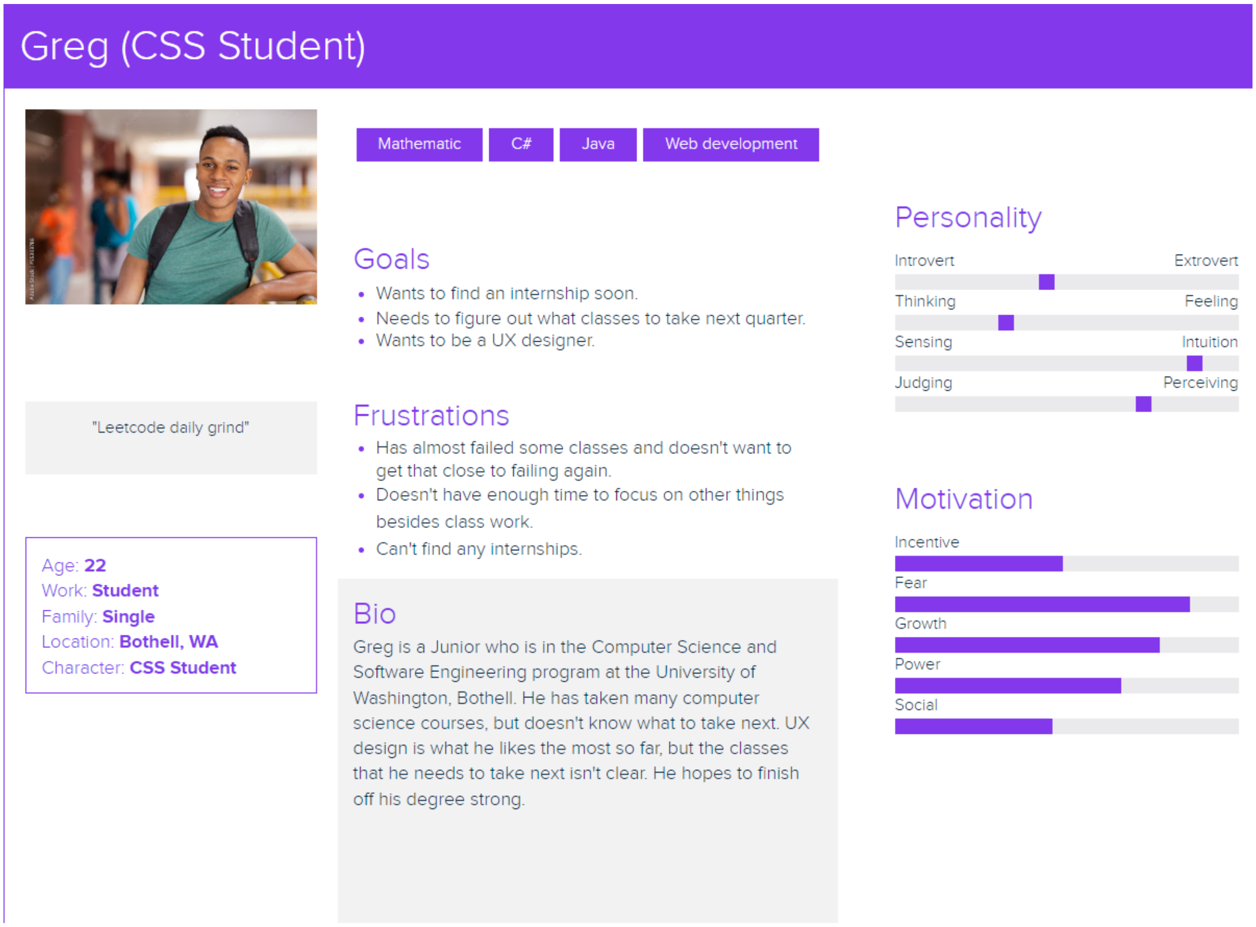
Age: **19**
Work: **Studen**
Family: **Single**
Location: **Bothe**
Character: **Prema**

Bio

Jessa is a freshman at the University of Washington Bothell. She is enthusiastic about taking classes at UW because it will get her closer to her goal of moving out and being independent. There aren't any majors that are appealing enough for her to commit to yet, so she is ng general courses. While at her time at UWB, she find a major soon, but there are too many ns to choose from.

Over time, I have realized that the premajor student doesn't make much sense in the context of the application. The application uses the major the student is in to find appropriate next courses. I thought it would be fine to include premajor students initially because they will have majors that they are planning to be in. But the issue with this is that certain courses (300 - 400 level) require that you be in the major before you take it. I was thinking of a workaround that would include either putting a note when presenting these classes, or just excluding them completely.

CSS Student



Greg (CSS Student)

Mathematic | C# | Java | Web development

"Leetcode daily grind"

Age: **22**
Work: **Student**
Family: **Single**
Location: **Bothell, WA**
Character: **CSS Student**

Goals

- Wants to find an internship soon.
- Needs to figure out what classes to take next quarter.
- Wants to be a UX designer.

Frustrations

- Has almost failed some classes and doesn't want to get that close to failing again.
- Doesn't have enough time to focus on other things besides class work.
- Can't find any internships.

Bio

Greg is a Junior who is in the Computer Science and Software Engineering program at the University of Washington, Bothell. He has taken many computer science courses, but doesn't know what to take next. UX design is what he likes the most so far, but the classes that he needs to take next isn't clear. He hopes to finish off his degree strong.

Personality

Motivation

Possible persona idea: Academic Advisor

Requirements:

Functional -

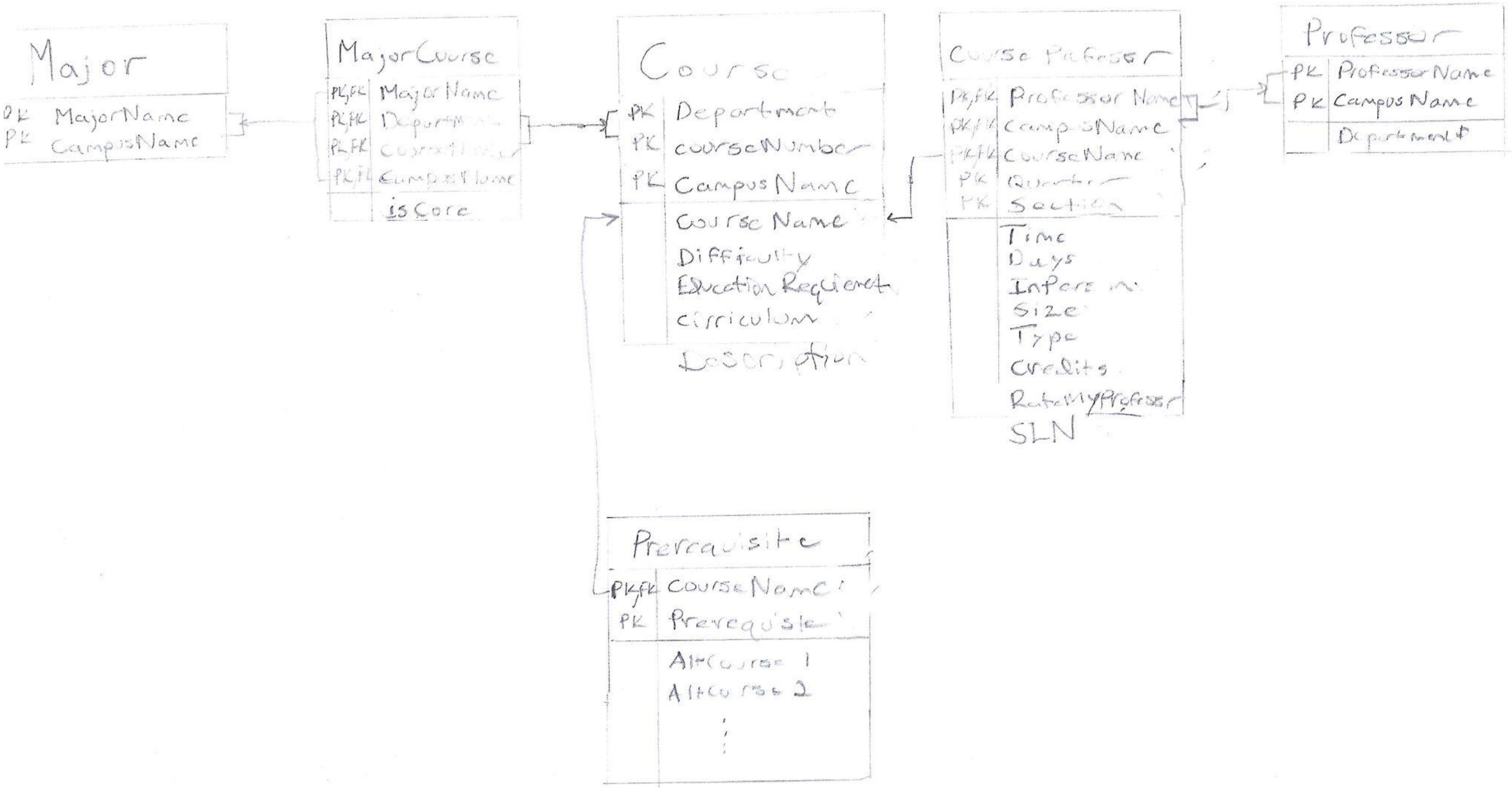
- The user must be able to query for a ranked list of classes they can take

- The user must be able to filter what aspects of the class are important to them
- The user must be able to export this data in a meaningful capacity
- The system must be able to be accessed by all UW students
- The user must be able to access the course webpage on MyPlan from the application
- The user must be able to select the courses that they have taken at UW in quarters prior
- The user must be able to select what major they are in or interested in at UW
- The system must represent the course attributes that are stored in the SQL database
- The system must be able to query the database by what the user gives in the parameters

Non-Functional -

- The system should be able to run on a Windows, Mac, and mobile device
- The user should be able to receive their data within 1 minute
- The system must be able to scale with the number of courses at all three UW locations up to 10,000 courses
- The user must be able to access and use the website with at least 2 9's of reliability
- The user must be able to understand how to use the application within 30 seconds
- The system must be able to retrieve assets through database or web crawling within 5 seconds

Diagrams



This is a logical database diagram using the relational model. There are some things I would revise, but this is the initial version. There are six tables in total with each table related to each other in some way. There should likely be more than six for this to really make sense as a database, but I was only thinking about functionality over a proper design.

To explain the tables a bit, there are essentially three main groups of entities: majors, courses, and professors. The relationship between course and professor is represented through course professor which is a table that represents classes that are offered each quarter. A better way to look at this is the different sections of classes where each section has a professor (sometimes more than one, but that is a different topic). On the other side, you have major courses which are used to show that a course is in a certain major. This can't be done in the course table since a course could be used for many majors. Calculus is a good example of this scenario. There is also a prerequisite which is used to show the prerequisites in the class. I wanted to make it so I could essentially select all rows with the value of a specific course and then look at those to see what the prerequisites are.

Sponsor Feedback

Sponsor meeting with Jake Ortiz on April 17th, 2023
Notes:

- Mentioned these points again
    - Add time constraint option
    - Add blurb about consulting with an advisor
    - Change wording to recommendations to take next quarter
    - Change wording and get rid of you
- Discussed edge cases of the application and how some were important and others weren't
    - Students wanting to use classes they took at community college
    - Premajor students?
    - Certain majors like business administration have multiple routes, how are you considering this?
- Add more filter options?
    - Time
    - Days
    - Etc.

Sponsor meeting with Jacob Lee Tea on May 1st, 2023
Notes:

- Clean up dropdowns, center them
- Add instructor option
- Make course list expandable