A screenshot of a social media post

Description automatically generated

Integrate and Configure

Mike’s Comments: Add bullet point list of requirements and specific details of each step

Specification: Software that can handle over 100 users for class registration, and be relatively simple to use

Finding premade software: There are software packages available for use that can be reconfigured to a specific use case, which makes the task easier than building everything from the ground up.

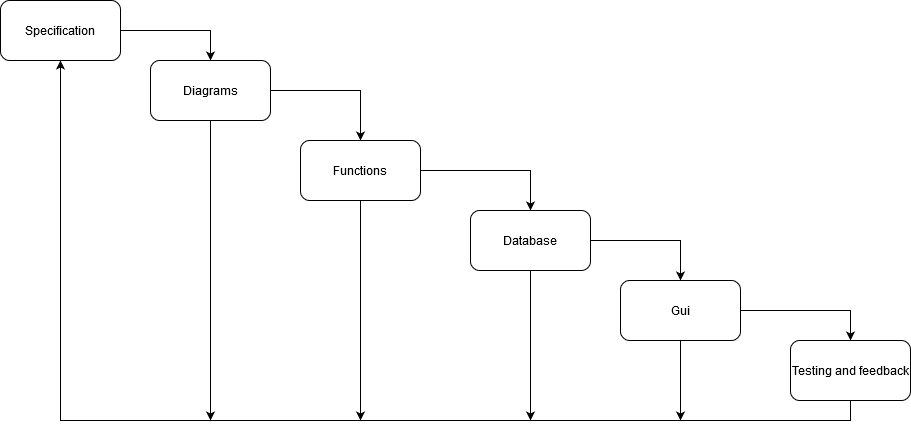
Moodle: A simple prebuilt web-based front end for education focused applications

Knack: An almost turnkey system for anything requiring data tracking, i.e. a database

Integration: Moodle already has many APIs with plenty of documentation allowing for it to be integrated with knack or other services in the future

Fill in the gaps: Fixing bugs and taking care of things not covered natively by the front and back end services

System validation: Testing and making sure the system is working as intended



Waterfall

Mike’s Comments: Add titles of what each diagram represents. What type of model is this representing?

Diagrams: How individual tasks are going to be accomplished and how the system is going to work as a whole

Functions: Making the base part of the program, accomplishing the main goal of the project

Database: Storing data efficiently and in a scalable manner, makes sense for a program meant to handle over 100 users

GUI: Gives the user an easier way of interacting with the program

Testing: Figuring out what is wrong with the program and fixing it, along with implementing suggested changes for improving its functionality

A screenshot of a cell phone

Description automatically generated

Incremental

Mike’s Comments: What is the basic product look like and what changes will be made with each incremental design?

The basic product would be a command line program with case statement menus and a SQL database housing all the stored data.

Basic functionality: Creating the database, searching, and printing courses

Database: Filling out the database with the necessary information such as people at the school, courses, and schedules for everyone

Initial version: The initial version would most likely only contain functions for 1 or 2 classes of users, and the full database.

Adding functions: More advanced functions such as checking for schedule conflicts and expanding the menu will be taken care of later in time

Intermediate versions: Full functionality but with some bugs in the code that will prevent the program from working

Troubleshooting: Looking for logical errors or edge case problems where a student is registered for too many courses for the database to handle

Testing for Scale: Create a lot of database entries to simulate the number of users the program is meant to handle

Final version: Ideally something better than lconnect if we have the time