Class Mover

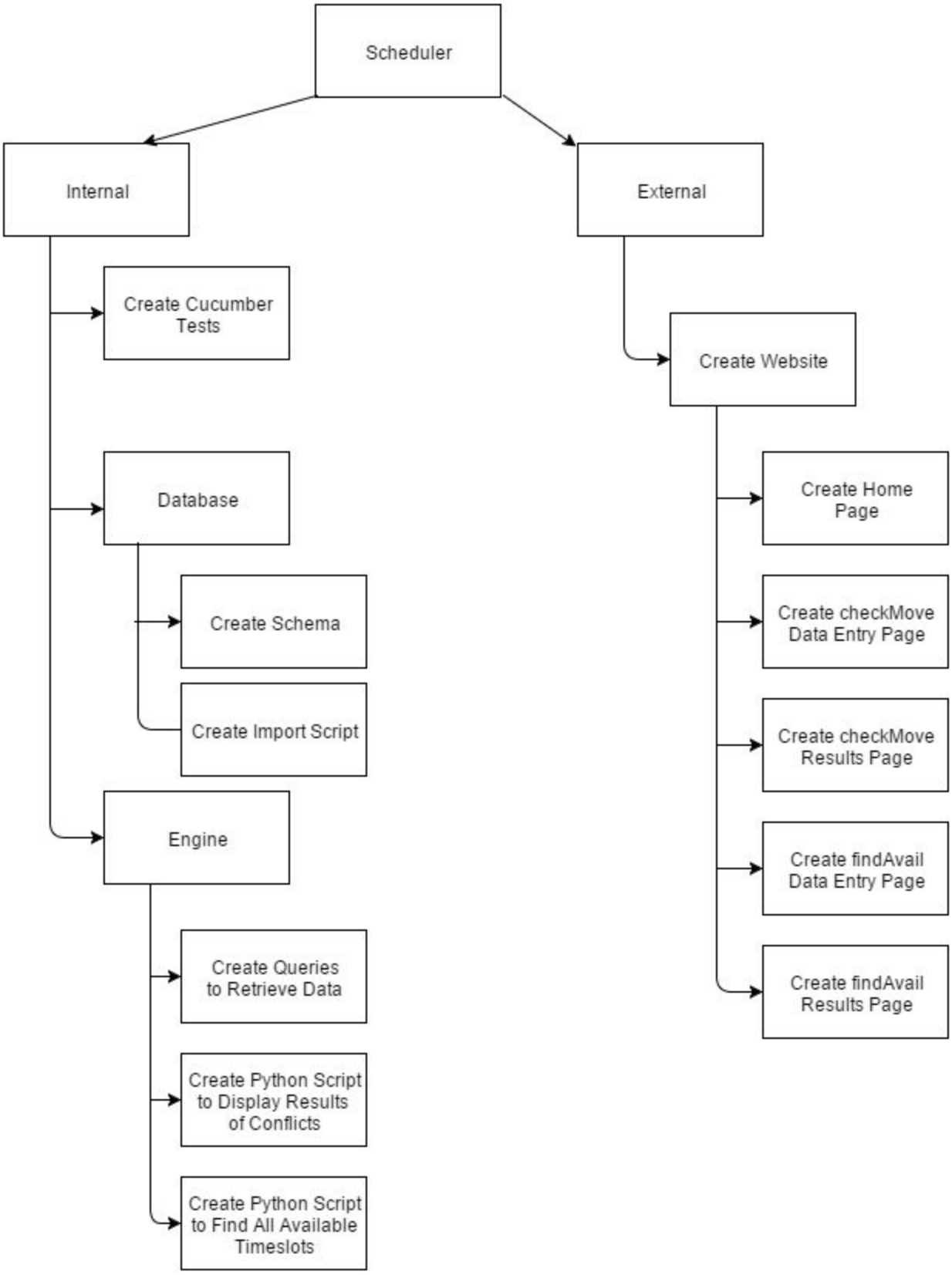
Executive Summary

This document contains the process that we used to develop the project and the documentation used to aid in the creation of the application. This application will be used to aid in the movement of classes from one time slot to another. It will display all conflicts that may occur if a class is moved to a particular day and time or display all possible days and times that a class can be moved to without many conflicts.

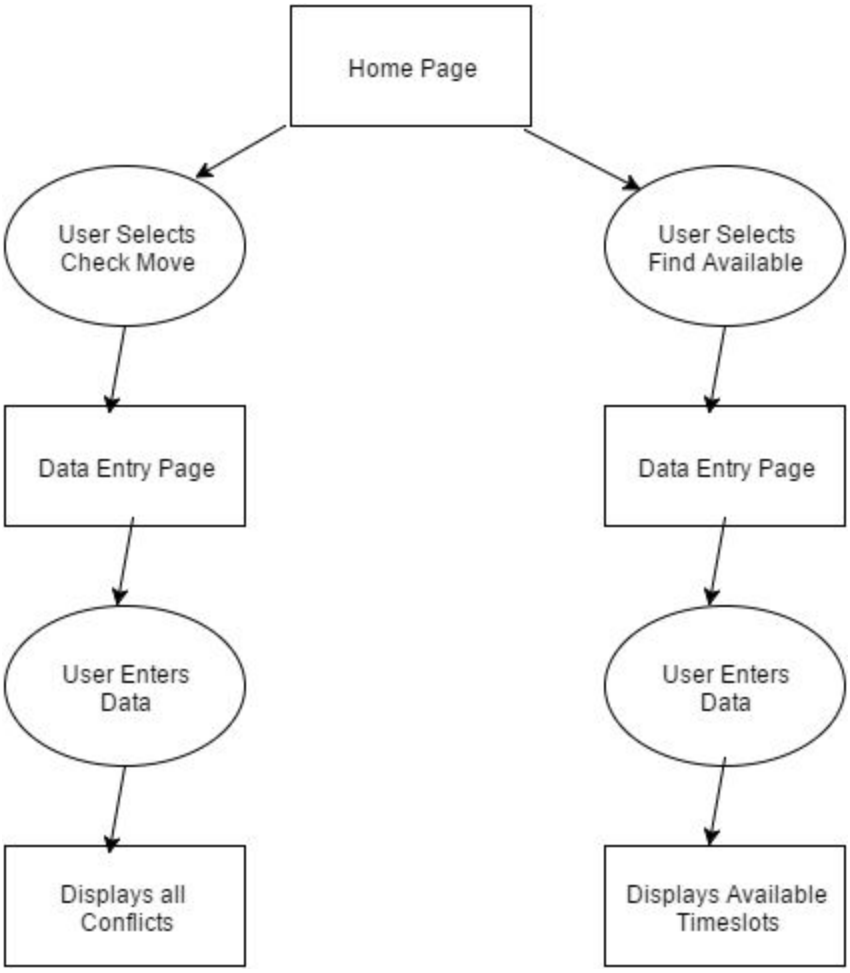
Project Scope

The application will display all conflicts that moving a course to a certain time would have. This includes displaying the id, name, email address, and conflicting course of conflicting students and the professor name and conflicting course of a conflicting professor. The application will not move the courses within the database.

Work Breakdown Structure



Detailed Design by Activity



Testing

Cucumber tests will be created for various cases that can occur while using the application. These tests will be run throughout the development cycle of the project to ensure that the application is running as expected whenever functionality is added.

Weekly Progress

Each week we set a goal to be achieved before our weekly check in with the client, to ensure that we made progress each week on the project.

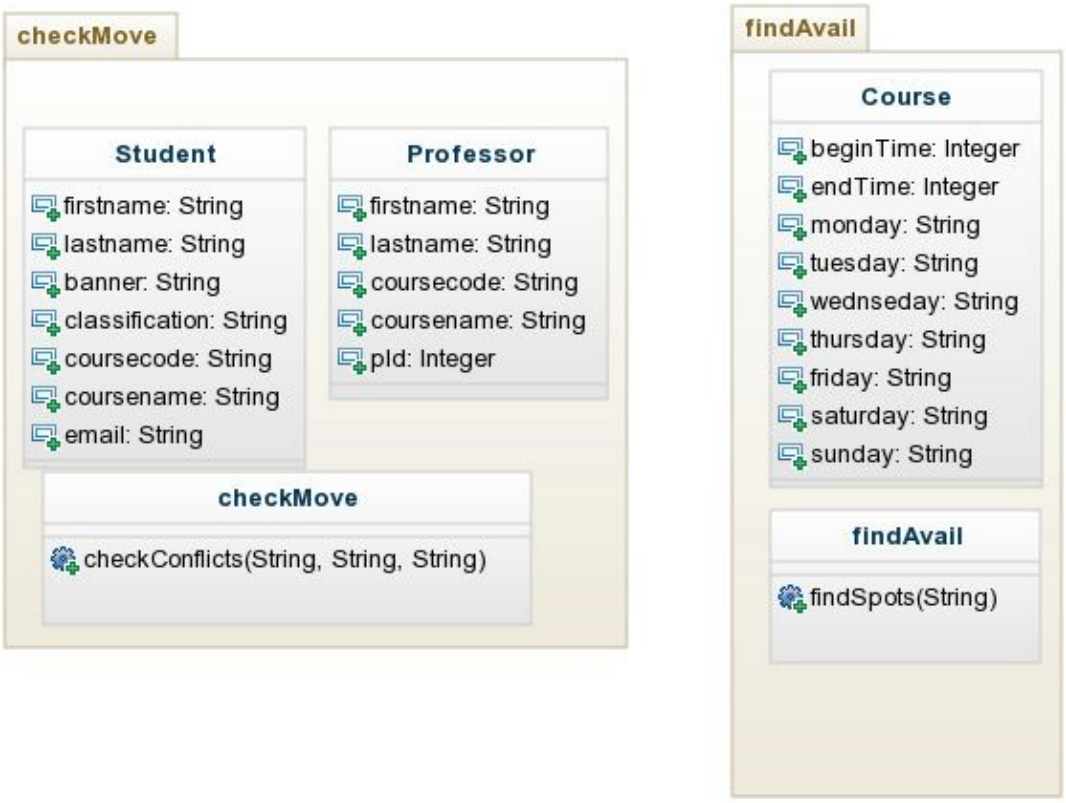
Requirements Check

In our weekly check-ins with the client, we ensured that we were meeting her expectations of what she wanted in the final product and that she would be able to use the tool.

User Testing

For user testing, each week presented our current work on the project to our client to let her evaluate the progress and give us feedback on any changes necessary for her optimal use

UML



Installation Instructions/Help Manual

The application requires python 2.7, MySQL, and the python mysql.connector. The application is intended for use on linux operating systems. The web based version of the app (using the files matching *Page.py) can be run in any web server implementation, but apache is recommended.

- Python 2.7 should come stock with any installation of linux.
- MySQL is likewise handled by your distro's package manager, see the documentation for your package manager to determine how to install packages.
- mysql.connector is a module that must be added to your python install. Pip is recommended for adding python packages
- Apache is also handled by your package manager.

Once the requisite programs are installed, mysql must be set up. A username and password combination must be set up for the app to use, and a database must be created for the tables. Once set up, these names should be added to the relevant fields in config.ini. Technically, the host can also be set, but it should be left as localhost. Following this, the .csv files should be placed in a folder within the same directory as import.py. (config.ini should also be in the same directory) Running import.py will then import the data from all .csv files in the folder named csvs.

To set up the apache versions, install apache2 using your package manager. Then, to set apache to run python, run the following commands.

- sudo a2dismod mpm_event
- sudo a2enmod mpm_prefork cgi
- sudo nano /etc/apache2/sites-enabled/000-default.conf
    - add “Options +ExecCGI” to the www directory listing for apache’s webroot
    - add “AddHandler cgi-script .py” within the listing for port 80
- sudo service apache2 restart

Now place the files findAvailPage.py and checkMovePage.py along with a copy of config.ini into apache’s webroot. These scripts can now be viewed from a web browser.

Usage:

There are two main applications. checkMove checks and reports the result of attempting to move a class from one time to another. The command line interface expects class name in the form <program><class number>.<section number>, ex. CS120.1 . The second argument is the dates to add the class, ex. MWF. The final argument is the time, which is defined in military time ex 1300-1350 for 1:00 - 1:50. The output lists the number of conflicts, and information about them, and includes the professor and students, along with their classification and conflicting class. The web interface checkMovePage operates similarly, but with input boxes. findAvail checks for all available times that a class can be moved within two parameters of acceptable losses. The first parameter is the same as for checkMove, and the next two are the number of seniors we are willing to have conflicts with, and the number of other students we are willing to have conflicts with. Any conflict with a professor is unacceptable. Once again, the web interface provides input boxes for the same functionality.

Deployment Plan

The application will be kept on a server owned by the development team for a week after the project due date to be used by the client. After this time period, the server will be shut down and the application will need to be redeployed to a new server for the client to use, which will be handled the owner of the application.