Project Apple

A Complete Application to Registration to Graduation System

Timothy Lytkine

The George Washington University

Table of Contents

Table of Figures

Abstract

The report provides an overview and technical analysis of the design of a university database management system (DBMS) called Project Apple built using mySQL and PHP on the SEASCF. Project Apple is the subsequent phase of the project analyzed in the STARS Advising System report (Lytkine, 2018). The Apple System consists of three modules from the Phase 1 project. These three modules include the advising system referred to in the STARS Advising System Report which will be referred to as ADS, an application system which will be referred to as APPS and a registration system which will be referred to as REGS. The integration of these modules forms a complete application to registration to graduation system. Using the presentation layer, data layer and logic layer of the three-tier architecture of the system an analysis of the integration process is performed. The design decisions made during integration including changes to the design of the underlying database are analyzed to understand how the Apple System works. Furthermore, use cases demonstrate the way that the integration of the modules comes together to form the complete system. Finally, to conclude a reflection of the process involved in the design of the system is included. The advantages, disadvantages and further possible improvements to the system are discussed. In addition, what would have been done differently in the design of the ADS module in Phase 1 if the schema and features of the other modules were known of is included. The appendix contains a code snippet that was essential to the integration of the modules. The source code can be found at the following GitHub link: https://github.com/GWU-CSCI-2541W/Team-A2.

Keywords: database management system

Project Apple Summary

Project Apple is the result of the integration of three modules developed by three different teams in Phase 1 of the project which is referred to in the STARS Advising System report. One person from each of these teams was assigned to form a new team to work on the integration of these modules. These modules consist of ADS (Advising System) implemented by Timothy Lytkine, APPS (Application System) implemented by Dietrich Reidenbaugh and REGS (Registration System) implemented by William Miller. The purpose of the integration of these modules is to accommodate the workflow demanded by the client which involves the underlying elements of the application, registration and graduation systems and the way that the workflow transitions between them. The result of the integration of the three systems will be referred to as the Apple System. The Apple System stores information about the Computer Science program at the George Washington University. Each user of the system has access to a certain set of features which query the underlying database based upon their role. These roles include system administrator, grad assistant and advisor (GS), chair of admissions committee which can also be a faculty advisor or faculty reviewer, faculty instructor, student applicant, current graduate student and alumni. The Three-Tier Architecture Model provides a deeper understanding of both the frontend and backend design of the Apple System and how they differ based upon the roles present in the system.

Data Layer of 3-Tier Architecture The data layer of the three-tier architecture of the apple system is where the information required for the functions of each user to operate is stored. This information is queried from the database and then used by the logic tier in order to form the application layer.

List of Changes Made to the Database Schema of the ADS Module.

1. The names of the facultyid and gwid attributes were condensed into a single id attribute in the personalinfo table.
2. The crn attribute was changed to the coursenum attribute which follows the dept attribute when viewing available classes, ex: CSCI 6221.
3. The login table was split up into a users and roles table. The users table now uses email and password for login rather than username and password. The roles table contains the complete list of roles present in ADS, REGS and APPS and the additional roles demanded by the client.
4. All of the personal information of the students, faculty and alumni is now stored in a unified personal information (personalinfo) table.
5. The applications table was renamed to the graduationapplication table. Rather than using the now non-existent student table, the graduationapplication table now stores the cleared bit which indicates if a student has been cleared to graduate rather than storing it in the previous students table.
6. The courses table was expanded to include the attributes from the REGS module which include dept, section, year, semester and professorid.
7. The advises table was changed so it stores the degree name of the student being advised.

8. The course_prereq table is now the prereqs table and it stores the prerequisite course ids in prereqid within a single attribute rather than in three separate ones.
9. The course_status and alumni_course_status tables are now stored within the transcript table which includes additional information from the registration module such as dept, professorid, year, semester, grade and title.
10. The degrees table was changed to the degreerequirements table and similarly to the prereqs table, rather than having three attributes for each core course, core courses are now simply stored within one attribute.

Rationale Behind Changes. The changes made to the ADS module specified above were necessary to the integration of it with the REGS and APPS modules. The common attributes between the three modules were condensed into a new set of tables that can be seen in in the section below. For further insight into how the schema differs between the Apple System and the ADS module, the STARS Advising System report (Lytkine, 2018) can be referenced. Most of the changes were made so that information across previously existing tables can be easily queried between the ADS, REGS and APPS modules. The id attribute was unified so that the personal info of users can be edited and viewed by querying one table regardless of the user type. Storing personalinfo in one table and distinguishing users by their id and role within the same table optimized queries since any user can be now identified by just their id and role. The crn attribute previously stored a string of the department and course num and the coursenum attribute stored the courseid. This was changed by separating crn into dept and coursenum and using the courseid attribute as the primary key of the courses table. This was necessary in order for the course registration and transcript queries in the registration system to work.

Database Design. The following is a list of the tables present in the final database design. The names of each table are italicized, the primary keys are underlined and the remaining attributes are followed by the primary key.

- *users:* email, password, id
- *roles:* id, role
- *personalinfo:* id, firstname, lastname, dob, address, ssn
- *admissionsapplication:* id, reviewerusername, status, finaldecision, semester, year
- *review:* applicationid, decision, defcourse, comments, reasons
- *recommendation:* recommendationid, applicationid, writername, writeremail, affiliation, rating, genericrating, crediblerating
- *documentstatus:* applicationid, applicationsubmitted, transcriptreceived, letterofrecreceived, personalinfosubmitted
- *academicinfo:* applicationid, degreeapplyingfor, gretotal, greverbal, greanalytical, grequantitive, gredate, greadvscore, gresubj, greadvdate, toeflscore, toefldate, bachgpa, bachmajor, bachyear, bachuni, masgpa, masmajor, masyear, masuni, areaofint, experience
- *courses:* courseid, dept, coursenum, section, title, credithours, day, time, year, semester, professorid
- *transcripts:* studentid, dept, coursenum, year, semester, professorid, grade, title
- *prereqs:* courseid, prereqid
- *advises:* studentid, facultyid, hold, degreename
- *graduationapplication:* studentid, courseid, year, cleared
- *degreerequirements:* degreename, courseid
- *gradecalc:* grade, qualitypoints

Logic Layer of the Three-Tier Architecture.

The logic layer of the apple system is what connects the database schema listed in the data layer in order to allow the queries of the presentation layer to properly function. The different queries of the system described above are are executed similarly to the way described in the STARS Advising System (ADS) Report (Lytkine, 2018). All of the tables in the schema described in the data layer are queried using SQL based upon the information required for a particular function. The data required for all three modules is stored within the same database. Upon login, a users role is queried from the database. This role is then used in a PHP function which redirects the corresponding user to a main menu containing the features they are given access listed in the presentation layer section above. A demonstration of the code used to distinguish lists of functions between roles can be found in Figure 1 in Appendix A. In order to support the integration of the ADS, APPS and REGS modules, the code used to connect to the database is stored in a PHP file that is referenced at the header of each page that needs to perform queries. In addition, another header is used which returns the email of the user logged in, a link to return to the menu and a link to logout. This allows the user to easily go back to the main menu. Similarly to Phase 1, each PHP page uses links and informations entered by a user to query the database using the SELECT query or manipulate the database using the SQL queries UPDATE, DELETE and INSERT. In terms of changes to the ADS module, many of these queries had to be modified in the source code in order to support the new database schema specified in the data layer. In addition, rather than having data immediately available for each user to see such as in Form 1 in the ADS system, the Apple System now contains a unified menu page which allowed for seamless integration of the ADS, REGS and APPS modules.

Additional Features. In addition to all of the queries previously existing in the APPS, ADS and REGS modules, additional queries were added to the system to satisfy the workflow demanded by the client. Some of the queries requested were already implemented in the prior systems which made integration simpler. These included the GS being able to update an applicant or students academic or personal information, view a list of all advisees for a faculty advisor (also applicable to a faculty advisor) and change the advisor for a student. In addition, the transcript functionality accessible by students and faculty was already present in both the ADS and REGS modules. The features added to the APPS module are as follows. The grad secretary or a faculty reviewer can now search for an applicant by using their last name or student number. The lists of graduate applicants, admitted students and statistics on the applicants admitted or rejected which include test scores and grades can now be viewed by the GS by using the semester, year or degree program. Finally, online recommendation letters are now built into the system. The features added to ADS module are as follows. Lists of graduating students, alumni and their email addresses and current students can now be viewed by the GS by searching using the semester, year or degree program. Furthermore, a feature added that is common to both the ADS and REGS modules is online advising forms to lift a registration hold. All of the SQL queries and PHP functions connect with the data layer in to form the presentation layer of the Apple System.

Presentation Layer of the Three-Tier Architecture.

Figure 1. Table with roles present in the Apple System and their corresponding functions and characteristics.

Role	Characteristics / Functions
System Administrator	-has access to all functions in system -only user that can manually add other users -update user information -deactivate users -change user roles -update admission document status -view admissions applications and reviews -update final admissions decisions -view student transcripts -enter student grades -update advisor assignments -update student holds -approve/deny graduation applications -update degrees -update available classes -update courses offered
Grad Assistant and Grad Advisor (GS)	-update personal information -view student information -update admissions document status -view admission applications and reviews -update final admissions decisions -view student transcripts -enter student grades -update advisor assignments -view student holds -approve or deny graduation applications
Chair of Admissions Committee	-update personal information -view admissions applications -review and update final admissions decisions
Faculty Reviewer	-update personal information -review admissions applications
Faculty Advisor	-update personal information -view advisee information / application status -view advisee transcripts -update advisee holds
Faculty Instructor	-can update their personal information -view class rosters -enter student grades -view student transcripts
Graduate Student Applicant	-can update personal information -apply for admission -view admissions application -view application status
Current Graduate Student	-update personal information -view class schedule -register for classes -view degree requirements -apply to graduate
Alumni	-update personal information -view transcript

Use Cases of Roles Specified in Figure 1. The following outlines the process by which the user interface allows the application to registration to graduation system to function. First, a graduate student applicant must navigate to the Project Apple website and apply for admission to a graduate degree program. In this case, the applicant applies to the Masters of Science Computer Science program. They can do this by clicking the create applicant account button on the homepage and entering a username and password. Upon login, they can update their personal information and then complete an admissions application. In the admissions application, they will fill out certain information such as the semester they are applying for, the year, the degree, test scores, previous universities, supplemental information and provide information for their recommenders. After entering all of the required information, they will be notified that their application is submitted. They can then check their application status which will be incomplete until the admissions committee reviews their application and the grad secretary and the chair of admissions committee admits the applicant. Once the admissions committee which consist of faculty members reviews the application they can make a recommendation which can be either admit (admit with aid or admit without aid) or reject. If the chair of the admissions committee who is also a faculty reviewer logins in, they can then view an admission applications and the reviews submitted for the application. They can then see the reviews submitted and admit the applicant. Once the applicant is admitted, when they login, they can check their application status. The statuses include “application received and decision pending”, “application materials missing” or “admission decision: accepted” or “admission decision: rejected.” If the applicant is admitted, they can choose to accept admission and matriculate. Once they accept their admission,

they become a current graduate student so their role in the system is changed accordingly. Upon login, they can now view their degree requirements which they can use to register for classes. Before they can register for classes, a student that has just matriculated must have the advising hold lifted off of their account. This can be done by them submitting an advising form that lists the course that they will take which a faculty advisor must approve upon login to lift the hold. Once they register for classes, they can view their class schedule. Each of the professors who teach the classes that the students are registered for will then have to submit grades. Once they submit grades for the courses the student has taken, the student can check their transcript, see their total GPA and make sure that they have taken all of the courses required for their degree. Once a current graduate student has met all of the requirements for graduation, they can then apply to graduate by filling out a Form 1. This form requires the student to list all of the courses they have taken and the degree they are applying for. If the student has met all of the requirements for the degree, their application is cleared and ready for approval. Once this happens, the grad secretary can review their application for graduation and approve their application for graduation. They can then login as an alumni where they can check their transcript, view the degree they graduated with and in what year and update their personal information.

Conclusion

Overall, the Apple System successfully integrates the ADS, APPS and REGS modules , satisfies the workflow and implements the additional queries requested by the client. The system has many advantages and some slight disadvantages. In terms of advantages, there is error checking across the advising, registration and application modules due to extensive error testing in the system which prevents users from entering invalid input. In addition, due to the login system from the APPS module being used, the passwords are secured by using hashing. Furthermore, the system supports a variety of different holds including a first time registration hold, an advising hold and other holds that a faculty advisor can choose to create. Finally, there is support for multiple degrees in the system. The disadvantages of the system are simply that some queries may not be fully optimized and there are differences in how the modules were implemented despite the user interface looking uniform. If there was prior knowledge of the design of the other modules, then the ADS system would have been implemented with a database schema closer to the one that is implemented in the Apple System. This would have eliminated the need to modify certain queries. In addition, a menu page that gives access to certain features by role rather than just listing all the information at once as seen in Figure 2 in Appendix A would have been implemented. Nevertheless, the Apple System meets the workflow specifications demanded by the client and fulfills its duties as serving as a complete application to registration to graduation system for a university.

Reference

Lytkine, T., Reidenbaugh, D., & Miller, W. (n.d.). Team-A2 Github Library. Retrieved May 1, 2018, from https://github.com/GWU-CSCI-2541W/Team-A2

(n.d.) CS 2541 Team Project Banner++: CS Student Admissions, Records and Registration System Phase 1. Retrieved May 1, 2018, from http://www2.seas.gwu.edu/~bhagiweb/cs2541/lectures/phase1.html

(n.d). CS Student Database Retrieved May 1, 2018, from http://www2.seas.gwu.edu/~bhagiweb/cs2541/lectures/phase2.html

(n.d). Phase 2 Submission Instructions Retrieved May 1, 2018, from http://www2.seas.gwu.edu/~bhagiweb/cs2541/lectures/phase2-instructions.html

Lytkine, T. (2018). Washington DC, United States: STARS Advising System (ADS)

Appendix A

Figure 2. Example of how features are restricted by roles on the menu page.

```
<html lang="en-US">
<head>
    <meta charset="UTF-8">
    <title>Menu</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
<?php
$allowed_user_types = array("USER");
include 'header.php';
?>
<h1>Menu</h1>

<?php if (in_array("USER", $_SESSION["roles"]) && !in_array("APPLICANT", $_SESSION["roles"]) && !in_array("ALUMNI", $_SESSION["roles"])) : ?>
<a href='view-info.php'>Update Personal Information</a> <br />
<?php endif; ?>

<?php if (in_array("ADMIN", $_SESSION["roles"])) : ?>
<a href='add-users.php'>Add Users</a> <br />
<a href='update-user-info.php'>Update User Information</a> <br />
<a href='deactivate-user.php'>Deactivate Users</a> <br />
<a href='change-user-roles.php'>Change User Roles</a> <br />

<a href='document-status.php'>Update Admissions Document Status</a> <br />
<a href='view-admissions-applications.php'>View Admissions Applications and Reviews</a> <br />
<a href='final-decisions.php'>Update Final Admissions Decisions</a> <br />

<a href='search-student-transcripts.php'>View Student Transcripts</a> <br />
<a href='search-student-grades.php'>Enter Student Grades</a> <br />
<a href='update_advisor_assignments.php'>Update Advisor Assignments</a> <br />
<a href='update_student_holds.php'>Update Student Holds</a> <br />
<a href='approve_deny_graduation_applications.php'>Approve/Deny Graduation Applications</a> <br />

<a href='update_degrees.php'>Update Degrees</a> <br />
<a href='update-classes.php'>Update Available Classes</a> <br />
<a href='add-remove-classes.php'>Update Courses Offered</a><br />
<?php endif; ?>

<?php if (in_array("GS", $_SESSION["roles"])) : ?>
<a href=''>View Student Information</a> <br />

<a href='document-status.php'>Update Admissions Document Status</a> <br />
<a href='view-admissions-applications.php'>View Admissions Applications and Reviews</a> <br />
<a href='final-decisions.php'>Update Final Admissions Decisions</a> <br />

<a href='search-student-transcripts.php'>View Student Transcripts</a> <br />
<a href='search-student-grades.php'>Enter Student Grades</a> <br />
<a href='update_advisor_assignments.php'>Update Advisor Assignments</a> <br />
<a href='list_of_advisees.php'>List of Advisees</a> <br />
<a href='graduating_students.php'>List of Graduating Students</a><br />
<a href='view_student_holds.php'>View Student Holds</a> <br />
<a href='approve_deny_graduation_applications.php'>Approve/Deny Graduation Applications</a> <br />
<?php endif; ?>
```