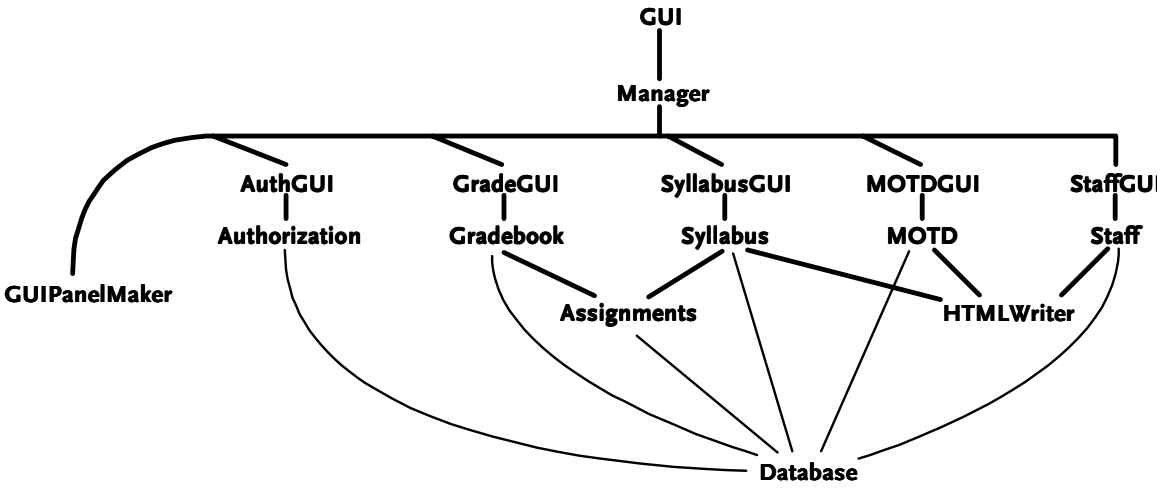
TEACHING TOOLS

Design Document

Colin Hartnett (cphartne) – 21 February 2003

1 High-Level Components



Descriptions begin with lower-level components and progress to higher-level components.

DATABASE

Stores authorization, grade, assignment, syllabus, MOTD, and staff data. It provides an interface that dominant components use to read and write relevant data.

GUI PANEL MAKER

Provides the set of widgets common to the user interfaces for each panel.

HTML WRITER

Produces and exports HTML files by reading a template file and filling its fields with parameters.

ASSIGNMENTS

Represent assignments. This includes the name of the assignment, a description of the assignment, due date, grading weight, and rubric. Given a student ID, is also able to return student's grades for said assignment and calculate the final grade based on the rubric.

AUTHORIZATION

Asks the user to specify a database file and login information. Does not allow the user to continue if login/password does not match login/password for database file.

GRADEBOOK

Provides a means of viewing and entering grades for each student and assignment. Allows the user to see only final assignment grades or all subgrades for assignment rubrics.

SYLLABUS

Facilities entry of assignment data, but not student grades (handled by gradebook). Generates syllabus web page.

MOTD

Allows input of message of the day entries and generates MOTD web page.

STAFF

Allows input of staff members. Generates a staff web page. Each staff member has an associated account to login to the program.

AUTH, GRADE, SYLLABUS, MOTD, AND STAFF GUIS

Provide the actual panel interfaces for each of these components.

MANAGER

Passes through the active subpanel GUI component based on input from top-level GUI component. Restricts access based on authorization information and permissions.

2 External Dependencies

This project is reliant on a database engine to store and retrieve grades, assignment data, authentication data, and web content.

3 Task Breakdown

PROJECT MANAGER

The project manager's primary responsibility is to make sure the project is completed. This entails scheduling meetings and otherwise facilitating communication between group members, scheduling milestones based on input from other group members, and ensuring the milestones are met. The project manager should have an intimate knowledge of the overall goals of the project but must be able to eliminate features and make tradeoffs should this become necessary to meet deadlines.

SYSTEM ARCHITECT

The system architect is responsible for creating a logically coherent design. This person must understand the function of each component and how they interact, and then create solid interfaces between components. The system architect must also support the lead programmer by explaining and clarifying the design as necessary. The system architect has the final say over both the design and external file formats.

GUI ARCHITECT

The GUI architect develops the look and feel of the user interface, then implements the GUI. This person has the final say on which widget library is to be used in the implementation of the GUI. The GUI architect is also the primary implementor of the GUI, and thus should be proficient in the chosen widget toolkit.

DOCUMENTATION MANAGER

The documentation manager is in charge of both internal and external documentation. This person keeps track of the requirements, specifications, and design over the course of the project. If and when major group meetings take place, the documentation manager should make sure each group member has ready access to all materials.

The documentation manager also works with the architect to develop the file formats, especially the webpage template format.

This person also handles the bulk of writing the user documentation and, therefore, must maintain close communication with both the system and GUI architects.

TOOLS MANAGER

The tools manager is in charge of setting up and maintaining all relevant tools, including (but not limited to) source control, makefiles, and install scripts. The tools manager may also help the lead tester develop tools to generate test data.

LEAD TESTER

The lead tester is responsible for all aspects of testing, including mining or fabricating test datasets. This person should assure that all component testing deadlines are met and should write test drivers for each high-level interface. The lead tester must also organize groups of outside testers. The lead tester is responsible for verifying whether or not the project meets the specification.

LEAD PROGRAMMER

The lead programmer is responsible for determining how the design is to be implemented and doing the bulk of the coding, excluding the GUI.

PROGRAMMERS (2)

The programmers write code for tasks assigned to them by the lead programmer.

GUI PROGRAMMER

The GUI programmer assists the GUI architect in implementing the GUI.

NOTES

The group members not directly involved with coding (the project manager, system architect, tools manager, lead tester, and documentation manager) can take on coding responsibilities if necessary. This system is intended to make it completely clear which members have the final say in each area of the project.

4 Schedule

28 FEBRUARY 2003 — DESIGN SELECTION

Determine the top-level design used as a framework for the project.

7 MARCH 2003 — TOP-LEVEL DESIGN FREEZE

Program manager signs of on top-level design. No changes should be made unless they are of dire necessity. Changes must be approved by project manager.

12 MARCH 2003 — INTERFACE PROPOSALS

Architects and programmers propose interfaces for the various components. The system architect evaluates various interfaces and attempts to piece together a coherent initial design.

17 MARCH 2003 — INTERFACE REVIEW

Architect facilities review of the interface proposals. Lead programmer assigns preliminary components to programmers. Programmers review each interface from the perspective of all components dependent on it.

21 MARCH 2003 — INTERFACE FREEZE

Project manager signs off on each interface. They can no longer be changed.

21–30 MARCH 2003 — SPRING BREAK

Important note: Very little work will be completed this week!

4 APRIL 2003 — DETAILED DESIGN PROPOSALS

Architects and programmers provide detailed designs for each component. All involved in the design process should meet and discuss designs. The system architect is responsible for fusing a coherent overall design from these proposals.

7 APRIL 2003 — DETAILED DESIGN FREEZE

Project manager signs off on detailed design. It can no longer be changed.

— IMPLEMENTATION COMMENCES

Now that the detailed design is frozen, the efforts of the group should be primarily dedicated to implementation.

— USER DOCUMENTATION WRITING BEGINS

Since the UI is frozen at this point, the documentation manager should begin writing the user documentation.

11 APRIL 2003 — FIRST-LEVEL COMPONENTS COMPLETED

First-level components are complete. The lead programmer and project manager must approve. Lead tester begins stress testing.

— BEGIN SECOND-LEVEL COMPONENT IMPLEMENTATION

14 APRIL 2003 — FIRST-LEVEL COMPONENTS FULLY TESTED

Lead tester signs off on first-level components.

16 APRIL 2003 — PARTIAL SYSTEM INTEGRATION

Components are integrated without full functionality to find bugs.

— PRELIMINARY USER DOCUMENTATION REVIEW

The documentation manager meets with the project manager and architects to assess the completeness of the user documentation, then begins revisions.

21 APRIL 2003 — ALL COMPONENTS COMPLETED

All components are fully implemented and partially tested.

— THOROUGH COMPONENT TESTING COMMENCES

— USER DOCUMENTATION REVIEW

The documentation manager presents each group member with a copy of the documentation to review. Errors, omissions, and inconsistencies are identified.

23 APRIL 2003 — COMPONENT TESTING ENDS

Lead tester approves all components.

25 APRIL 2003 — FULL SYSTEM INTEGRATION

The system is completely integrated with all specified functionality.

— COMMENCE SYSTEM TESTING

Lead tester begins thorough system tests. These tests should be extremely involved and should assert that all elements of the specification are complete and bug-free.

28 APRIL 2003 — TESTING CHECKPOINT

Lead tester should meet with programmers to assert that all bugs found in system testing have been or are being fixed.

30 APRIL 2003 — TESTING COMPLETE

At this point, the program should be bug-free. The lead tester, lead programmer, and project manger sign off on the project.

— USER DOCUMENTATION COMPLETE

The project manager and architects have approved the user documentation.

— PREPARE FOR PUBLIC DEMO

Group members should meet to determine how to best present the project.

2 MAY 2003 — PUBLIC DEMO

Present project to the world.

— DEMO REVIEW

Group should meet to identify bugs encountered in demo. If necessary, a shotgun testing strategy should be developed. Programmers should be assigned to fix bugs.

7 MAY 2003 — FINAL TESTING

Group meets to assert the bugs encountered in demo are indeed fixed. Lead tester, lead programmer, and project manager sign off on project.

— PROJECT ASSEMBLY

All project materials should be assembled together, ready to be handed in.

9 MAY 2003 — FINAL SYSTEM HANDIN

It's over, kids.

5 Assumptions About Specifications

My version of the teaching project is based primarily on Liza Molinari's specifications, as it focuses on aiding course professors and teaching assistants. Its goal is to assist in the tasks of managing grades and creating a course website, which would include a message-of-the-day page (and archive), a general information page, a syllabus/calendar page, an assignments page, a staff page that also lists office hours, and a links page. This version does not include dynamic access to grades by students, as this greatly simplifies the process since the program becomes standalone and does not need a web server to run. Thus, the program does not have any web-based user interface components. All the web pages produced by this program are output, as once they are generated, there is no need for the program to either read them or gather any information from them.