CI102

LAB 6

Quiz

Note: You should take Quiz 2 at the start of your lab period.

Objectives

1. Define an initial architecture for your project.
2. Create architectural diagrams to represent your system from at least two perspectives – an overview of the entire system, and one more particular perspective.

Designers and developers need to have a common understanding of the system that they are building. Architectural design provides that understanding at a high level. This is commonly done with an Overview diagram, and various architectural perspectives. The architectural design gives everyone the "big picture" view of the system. This lab will help you create and document an initial architecture for your system.

Procedure

Step 1 – Create a System Overview Diagram

An Overview Diagram helps people understand the following:

System boundary – The boundary tells where the system begins (or ends). Things inside the boundary are part of the system; things outside the boundary are not part of the system; anything that crosses the boundary is an interface.

System components – Try to identify the major parts of your system from a technical perspective. Typical major components are:

- Screens or Web pages that your system displays
- Databases that you design and whose data is part of your system
- Files (e.g., data that is part of the system but not stored in a database)
- Code that you write or incorporate into your system

1.A - Use Figure 6-1 to start to define your system.

- Fill in the system name
- Identify things that cross the system boundary: does your system interact with people? This is a user interface. Does your system interact with another system? This is a data interface. Does your system exchange messages with another system? This is a programming interface. Draw these in using arrows to represent things coming into or out of your system. Label each arrow.

1.B - Use Figure 6-2 to identify some of the components that exist within your system boundary

- If you identified a user interface above, create a component to represent the code and screens of the user interface.
- If there is data flowing into or out of your application, list a file or database where your system will store the data (if it gets stored and not used immediately)
- List one or more component to represent the software in your system. If you can name major functional parts of the software, that would be a reasonable starting point. If not, show one block to represent the code.

1.C - Use Figure 6-3 to combine the information in 6-1 and 6-2 into a single system overview diagram

Step 2 – Create a Screen Hierarchy Diagram
A screen hierarchy diagram provides a single picture showing all the screens in the system and the basic navigation paths connecting the screens. Typically, the "home" or "landing" screen that is displayed when the system starts is shown at the top (or on the left). Pages that can be reached from that page are shown below the home page, and so on. Note that not all possible navigation paths need be shown. For example, if every screen has a Help button, all those links are often not shown in this diagram.

Be sure to provide a caption for each of your diagrams.

What to Turn In

In order to obtain full credit for this lab, ***each team*** must turn in:

1. Figure 6-1 - System Boundary
2. Figure 6-2 - List of major system components
3. Figure 6-3 - System Overview Diagram
4. Figure 6-4 - Screen hierarchy diagram for your system
5. Project Gantt Chart - Update and upload

Figure 6-1 – System Boundary

Figure 6-1 represents the system boundary for BuddyU. There are three interfaces: A user interface for students wanting to schedule classes with their peers; a program interface for authentication of students.

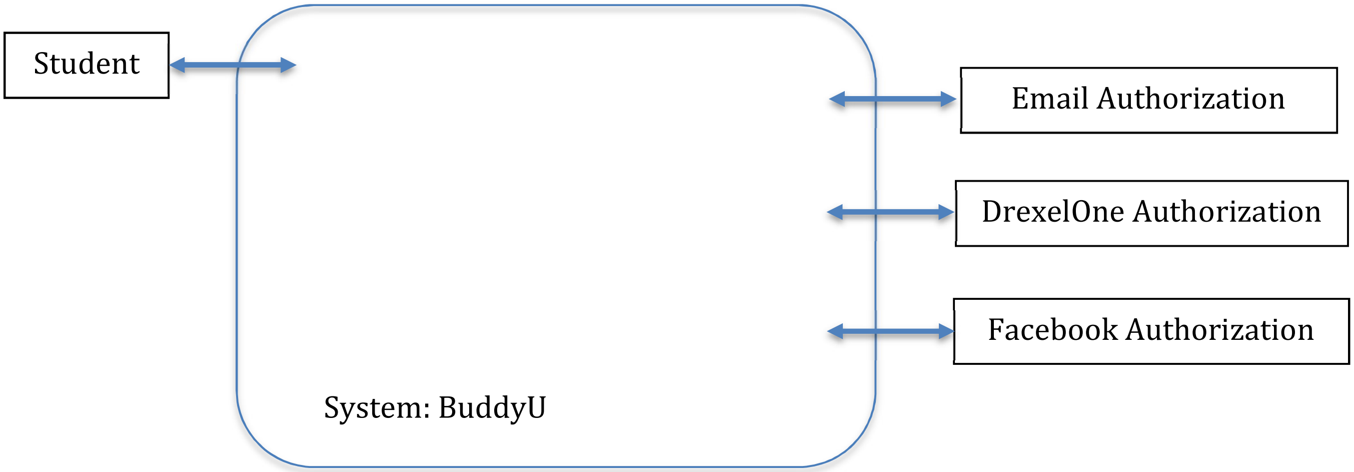


Figure 6-2 – Possible system components

Figure 6-2 provides an initial list of system components. Component 1 would include screens and the code that supports the screens. Components 1-5 are program code with each component addressing one of the major functional aspects of the system. Component 6 is the database that stores all the data needed for BuddyU.

1. BuddyU App-Mobile (Android) app for students
2. Event Manager – App Interface for Administrators
3. Access Manager – User Authentication Module
4. User Manager-history and preferences
5. Server code manager- Server code for mobile app
6. Database (MongoDB)-stores user information (including user profile and how long the user is in the app), stores number of downloads, APIs of Drexel TMS, CORQ, and Schedulr

Figure 6-3 – System Overview Diagram

Figure 6-3 shows the major components of the system, the system interfaces, and the main connections among components.

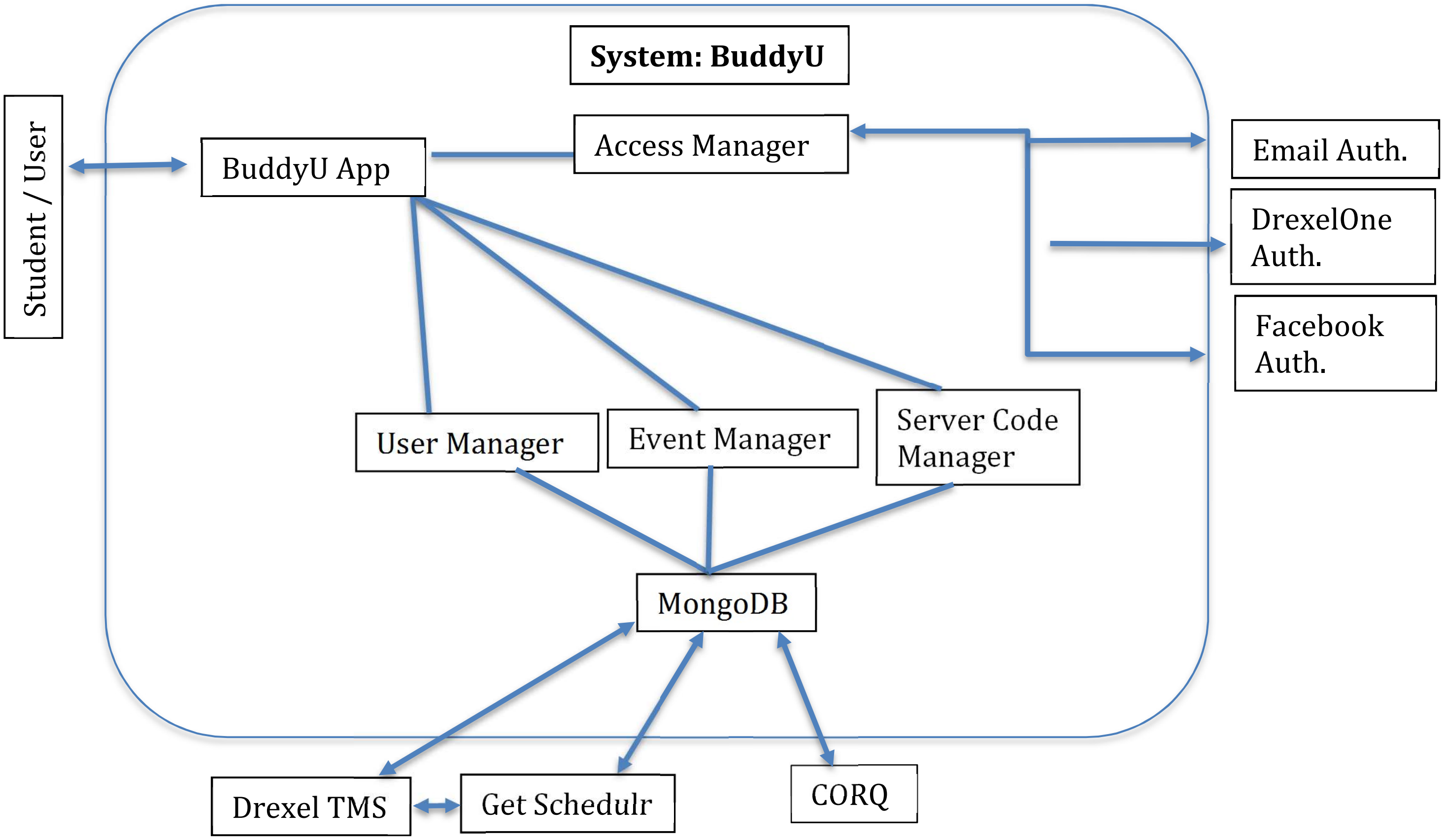
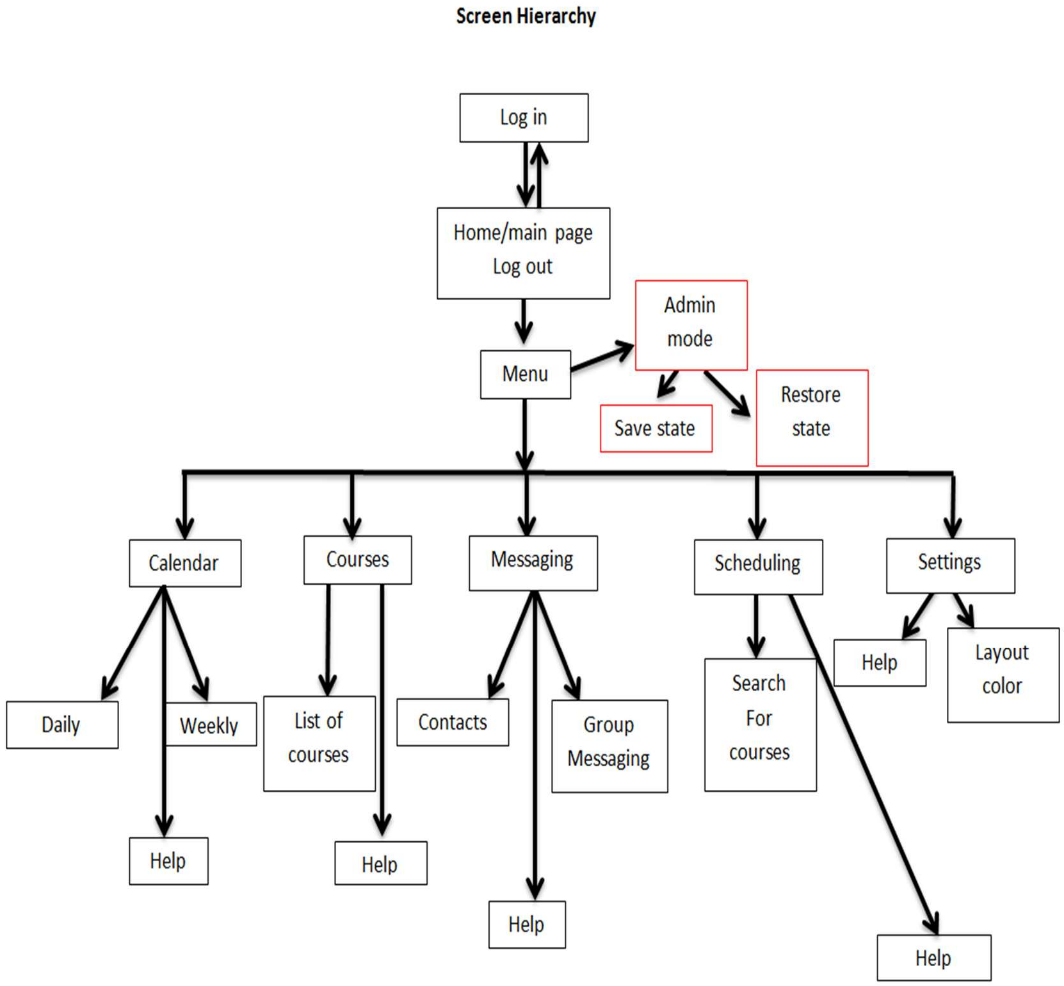


Figure 6-4 – Screen Hierarchy Diagram

Figure 6-4 shows the screen hierarchy for the BuddyU interface.


Screen Hierarchy
Log in
Home/main page
Log out
Menu
Admin mode
Save state
Restore state
Calendar
Courses
Messaging
Scheduling
Settings
Daily
Weekly
Help
List of courses
Help
Contacts
Group Messaging
Help
Search For courses
Help
Help
Layout color