

### Introduction

We began this project with the goal of creating an online language supplement to assist students enrolled in the University of Washington's Technical Japanese Program (TJP). Not only is our system aimed at helping students to review class concepts, but it also aims to facilitate the administration of class assignments and the communication between students. Major design concepts include a "flashcard" like approach for review sessions, quiz sessions with immediate feedback, and a mechanism for recording and viewing user study progress.

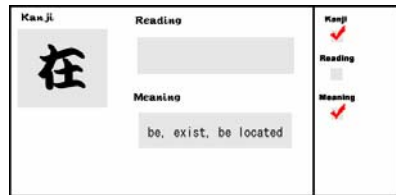


### User Needs Assessment

#### Paper Prototypes

We Based on our discussions with the head TJP instructor, prior experience in the TJP program, and an analysis of existing TJP materials, we created paper prototypes for kanji and vocabulary review/quiz interfaces.

Figure 1: Kanji Paper Prototype



The approach we took to create the Kanji review prototype was based off an established method of study: flashcards. Our virtual Kanji flashcard contained a Kanji, its readings, as well as its English equivalent. Checkboxes were used to hide and display information.

Our interviews were aimed at uncovering the current study habits of TJP students, and to collect feedback on the paper prototypes. In addition, we implemented a survey in order to identify demand for other learning ideas we gleaned from TJP documentation.

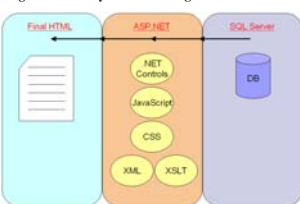
#### Findings

- Students are most concerned with their class grade. However, the system focus should be on helping TJP students learn Japanese overall.
- A basic template format should be used for Reviewing and Quizzing modules.
- Detailed functionality should be included as an option for ambitious students.
- Grammar & Intonation modules could be added to the Learning Component.

### System Design

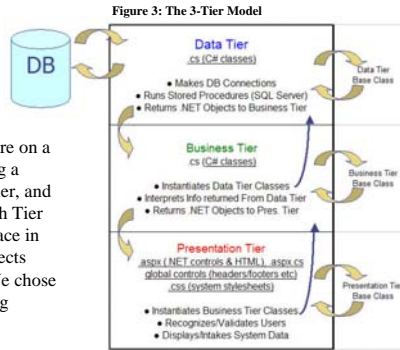
#### Technologies in the System

Figure 2: TJO System Technologies



We developed TJO using Microsoft's ASP.NET environment. We used Microsoft SQL Server for our backend database. We adopted a 3-Tier model for our system framework. This model was designed with the goal of making dynamic websites more scalable (for adding new modules) and flexible (for modifying and improving existing modules). Other Technologies we worked with included JavaScript, CSS, XML, and XSLT.

### Programming Architecture: 3-Tier Model

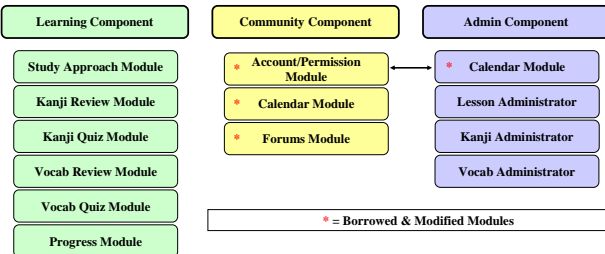


We based the TJO programming architecture on a 3-Tier model, containing a Data Tier, a Business Tier, and a Presentation Tier. Each Tier represented a logical space in which class-defined objects could be instantiated. We chose C# for our programming language.

### System Modules

Our modules were categorized under three main system components: 1) Learning, 2) Community, and 3) Admin. Some modules were borrowed from existing code and modified to work properly in our system. Other modules were created completely by the TJO team, and then integrated into the system.

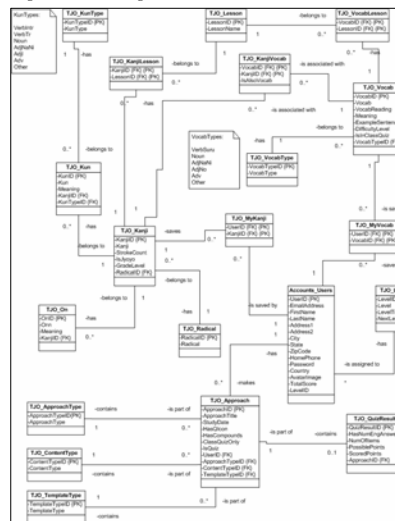
Figure 4: TJO System Modules



### Database Design

We first identified potential entities for the database by sketching out our ideas on paper. As a result we came up with the "Kanji", "Vocabulary", & "Lesson" tables, along with relationship-supporting tables. In addition, we created the "MyKanji" and "MyVocab" tables that would associate users to specific Kanji and Vocabulary. This allowed for users to "save" their own Kanji and Vocab. Using a similar approach, we finally created the tables for storing user review sessions and quiz results. As necessary, subsequent supporting tables were assigned in order to store additional information that we determined would grow throughout the life of our system.

Figure 5: TJO ER Diagram



### System Representative Modules

Figure 6: Kanji Review Module



#### Kanji Review

This module was designed to help students review basic Kanji information in conjunction with compounds associated with each Kanji. The user is able to display information after they guess the correct answer.

#### Kanji Data Entry

Figure 7: Kanji Data Entry Interface

The Admin module was designed for instructors so they could input Lesson, Kanji, and Vocabulary information into the system. Here is the interface for Kanji data entry. After searching the database for a specific Kanji, the data fields could be either newly filled in or updated.



### User Impact Assessment

#### Methods

For our impact assessment, we conducted usability tasks while asking informal interview questions. We repeated this process on seven students and two instructors. Participants were given one task for each module and were asked to write down problems with each module plus corresponding solutions for these problems.

#### Common Problems Found

- Lack of on-screen explanation/directions
- Poor link/message visibility
- Poor Kanji Module visual cues
- Inconsistency in My Account & Kanji Quiz layouts
- Approach page has too many options at once
- Need option to begin a quiz based on review just completed
- Only Instructors should be able to create user accounts
- Need to consider copyright issues with TJP sensitive materials

### Next Steps

- Use Impact Assessment feedback to enhance system modules
- Enhance system to meet security needs of TJP
- Continue programming incomplete modules
- Research a deployment and maintenance plan
- Iteratively receive feedback from users