Team Undefined
Documentation (Coding) Standard
April 27, 2016

Daniel Baggott
Matthew Canton
Daniel Silveri
Darryl Papke

User's Manual

There are two user types that can use Tashi Talk, students and teachers. The following manual will start with students and then teachers.

1. **1.Students**
    a. **Registering - Before using Tashi Talk students are required to register. This can be done by clicking the "Sign Up" button located on the homepage of the website. Once on the sign-up page enter a first name longer than two characters. A last name longer than two characters. A valid email. And a password longer than 8 characters that must contain numbers and letters. Aftering entering the data the student can click the button following the fields to create their login.**
    b. **Logging in- Logging into Tashi Talk can be done via the login page. A student can reach the login page by clicking the "Sign In" button on the home page. Once on the login page then enter the email and the password you signed up with when creating your user. If entered correctly they will log you into the student dashboard (main page). If entered incorrectly an error message will be displayed and you may attempt to log in again.**
    c. **Contact/send feedback to Tashi Talk - Through the feedback page feedback and messages can be sent to Tashi Talk. The feedback page can be accessed via the front page of Tashi Talk by clicking the "Contact/Feedback" button located at the top of the page. When logged in the feedback page can be accessed through the feedback button available through the drop down menu in the toolbar at the top of the page. Sending feedback requires a message though a name and Email are optional. Clicking send message will send your message to Tashi Talk and a notification will be displayed.**
    d. **Log out - When logged in the top toolbar will contain a drop down menu. Click the drop down menu to expose more options. The option at the bottom of the list is to Log Out. Logging out will require that you log in again before you can use features exposed in Tashi Talk.**
    e. **Chat with Partners - You can chat with others on the site with live video chat by accessing the video chat page. The video chat page can be accessed via the "Chat Partners" button in the top toolbar. If on the home logged in page choosing the option "Chat with Partner" will send you to the chat page. When on the chat page click the enter room button to enter the live video chat.**
    f. **Respond to video prompts or assignments - Currently any default and assigned video prompts can be accessed via the home logged in page.**

Clicking the "Prompts" button in the toolbar or clicking the "Practice with Prompts" button in the student's home page will take them to the prompts page from which they can see available default and assigned prompts in a list. Assigned prompts are available on the prompts page and on the student's main page as a list.

2. **Teachers**
    a. **Teacher's have a special dashboard removed from the user dashboard. Obtaining the required id numbers to be a teacher user requires contacting Tashi Talk and becoming approved and being issued the appropriate id numbers.**
    b. **Teachers can create a class - Teachers can create a class by clicking "Start a Class" icon on the Teacher's dashboard or by clicking the "Create a Class" button in the toolbar. Enter the class ID and class name of the new class then submit the information.**
    c. **Create assignment - Teachers can create an assignment on the create an assignment page. Teachers can access the create an assignment page by clicking on "Create an Assignment" page from the main page or the "Create an Assignment" button in the toolbar. On the create an assignment page the teacher should enter their teacher ID, the class ID, the prompt title, the prompt description, and choose a prompt image to upload via the "Choose File button". After entering the information clicking "Submit" will create the assignment.**
    d. **Grade Assignment - Assignment grades can be chosen via the grade assignment page. The grade assignment page can be accessed from the teacher dashboard by clicking the "Grade Assignments" icon or by clicking the "Grade Assignments" button in the toolbar. On the grade assignments page a teacher can choose a grade for the drop down list and submit it via the "Submit" button.**
    e. **Go back to dashboard - The teacher can go back to the teacher dashboard by clicking the "Back to Dashboard" button on the top toolbar.**

System Documentation Presentation

4/26/2016

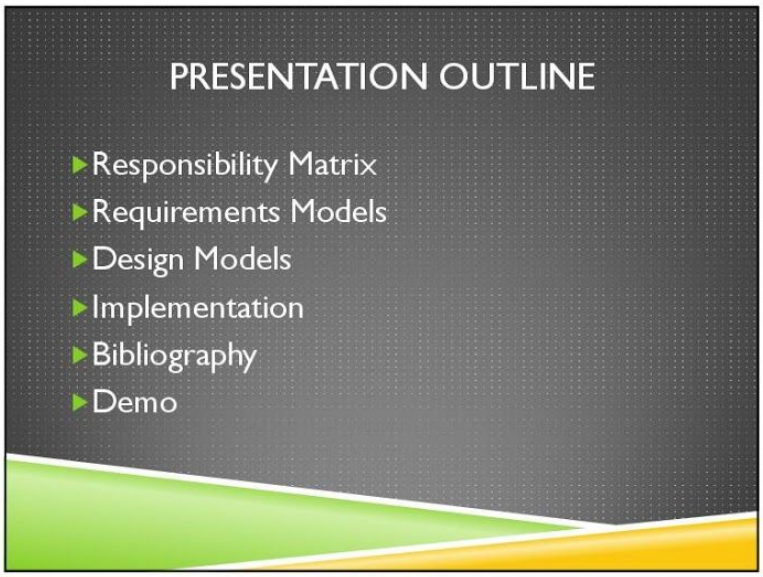
1

PROBLEM DESCRIPTION & OBJECTIVES

- Web based software that allows Japanese English students and English Japanese students to learn to communicate with each other.
- Quizzing/assessment component which records student responses to teacher-made prompts

SUCCESS CRITERIA

- The website is both easy-to-use and visually pleasing.
- Teachers can upload assignments for their students to complete.
- Students are able to upload submission in response to teacher prompt.
- Teachers are able to grade student's submissions.



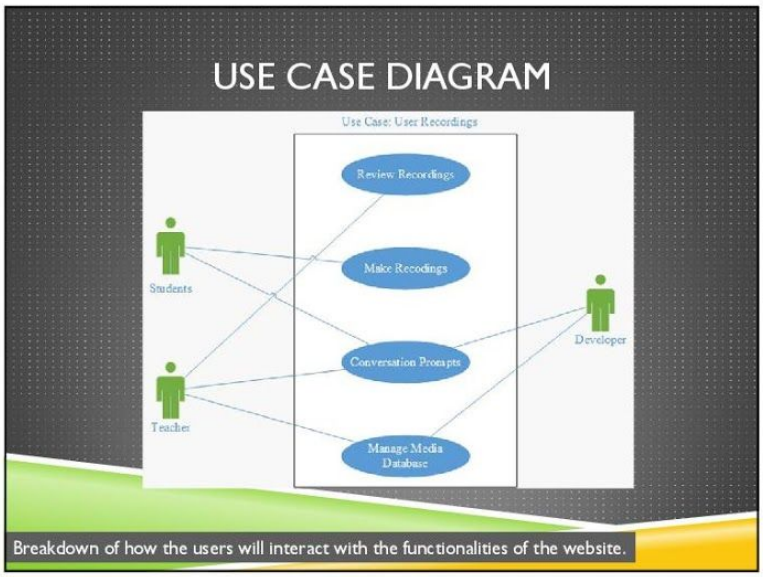
TEAM MEMBER RESPONSIBILITIES

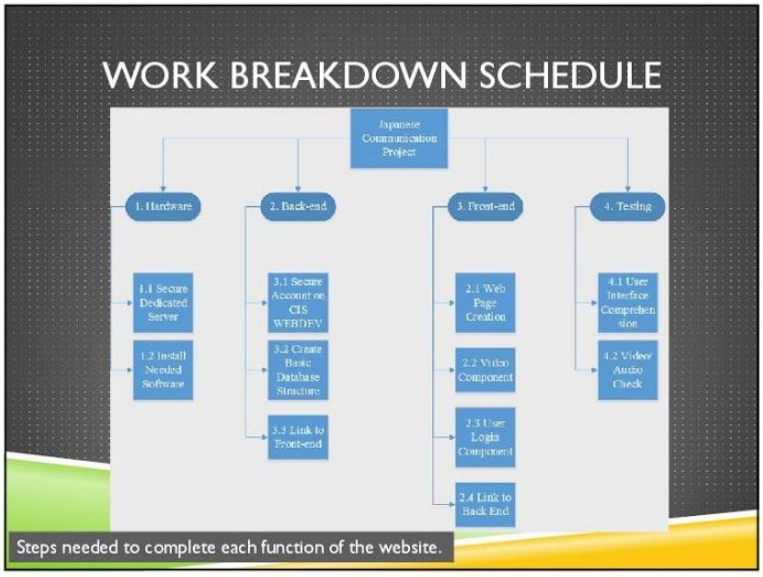
Task	Daniel Baggott	Daniel Silveri	Darryl Papke	Matthew Canton
Project Description	x			x
Project Objectives				x
Success Criteria				x
Resource Estimation				x
Cost Estimation				x
Project Schedule	x			
Responsibility Matrix			x	
Risk Planning			x	
GANTT/PERT Chart	x			
Management/Technical Constraints			x	
Project Monitoring	x			
Use Case Diagram		x		
Activity Diagram				x
Requirements Class Models			x	
Data Dictionary			x	
Non-Functional Requirements		x		
Tools Used		x		
Bibliography		x		
Product Testing		x		
Quality Assurance		x		
Software Design			x	
Project Management	x			
Customer Relations	x			
Implementation	x	x	x	x

REQUIREMENTS MODELS

GANTT CHART

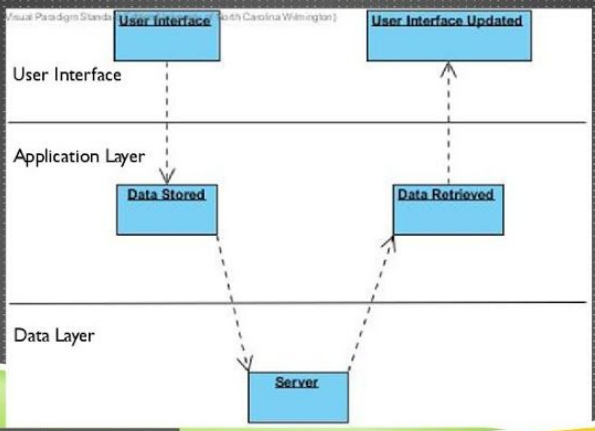
Schedule for assignments and date to be completed.


USE CASE DIAGRAM
Use Case: User Recordings
Review Recordings
Make Recodings
Conversation Prompts
Manage Media Database
Students
Teacher
Developer
Breakdown of how the users will interact with the functionalities of the website.


WORK BREAKDOWN SCHEDULE
Japanese Communication Project
1. Hardware
2. Back-end
3. Front-end
4. Testing
1.1 Secure Dedicated Server
1.2 Install Needed Software
3.1 Secure Account on CIS WEBDEV
3.2 Create Basic Database Structure
3.3 Link to Front-end
2.1 Web Page Creation
2.2 Video Component
2.3 User Login Component
2.4 Link to Back End
4.1 User Interface Comprehension
4.2 Video/ Audio Check
Steps needed to complete each function of the website.

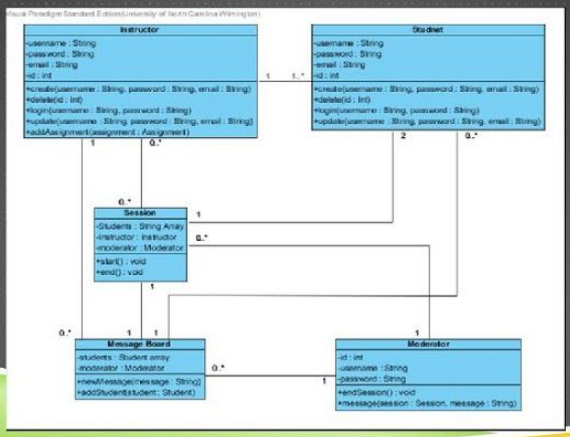
DESIGN MODELS

ARCHITECTURAL MODEL



Three-tier architectural model.

DESIGN CLASS DIAGRAM



Description of classes that will need to be implemented.

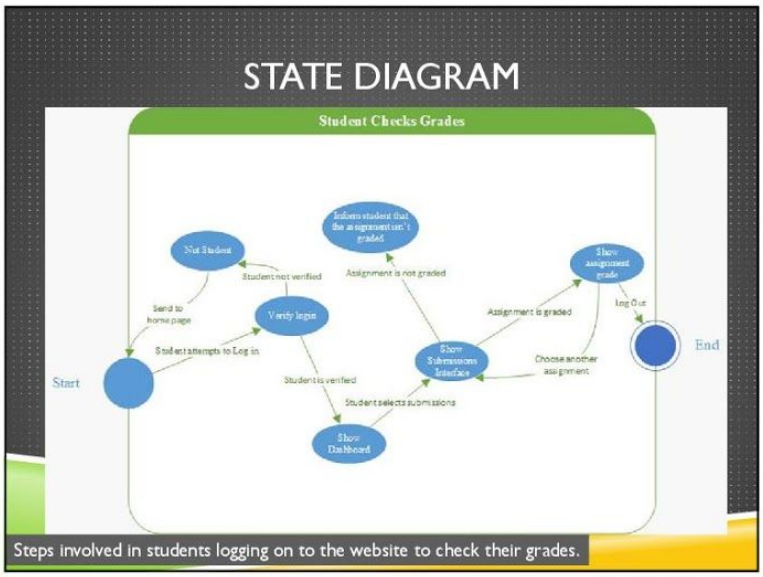
PRE AND POST CONDITIONS

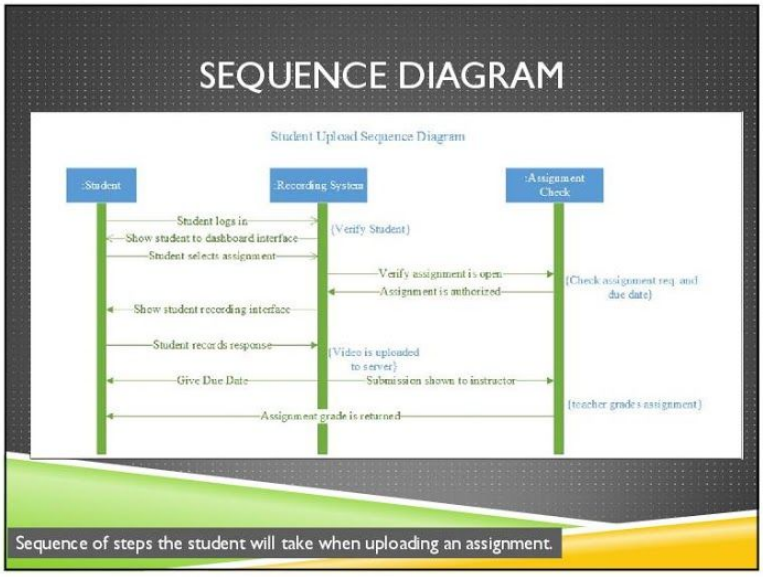
Pre Conditions

- Student must login to the website to access class assignments.
- A class must exist for a teacher.
- Teacher creates assignments for a given class.

Post Conditions

- Student completes and then submits assignments.
- Teacher can create assignments for classes.
- Students are graded on assignments that are for their class.


STATE DIAGRAM
Student Checks Grades
Not Student
Inform student that the assignment isn't graded.
Show assignment grade
Student not verified
Assignment is not graded
Send to home page
Verify login
Assignment is graded
Log Out
Student attempts to Log in
Show Submissions Interface
Choose another assignment
End
Start
Student is verified
Student selects submissions
Show Dashboard
Steps involved in students logging on to the website to check their grades.

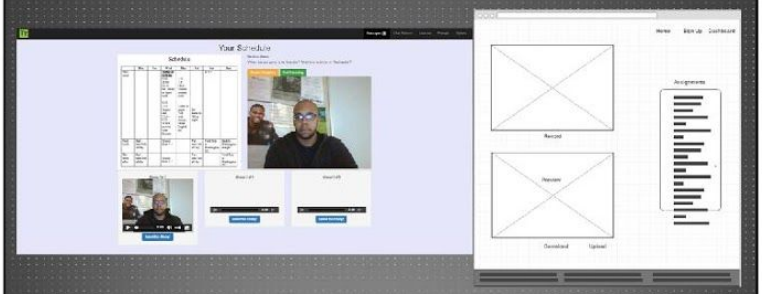

SEQUENCE DIAGRAM
Student Upload Sequence Diagram
:Student
:Recording System
:Assignment Check
Student logs in
(Verify Student)
Show student to dashboard interface
Student selects assignment
Verify assignment is open
Assignment is authorized
(Check assignment req. and due date)
Show student recording interface
Student records response
(Video is uploaded to server)
Give Due Date
Submission shown to instructor
(teacher grades assignment)
Assignment grade is returned
Sequence of steps the student will take when uploading an assignment.

IMPLEMENTATION

USER INTERFACE

User interface final design compared to initial design.

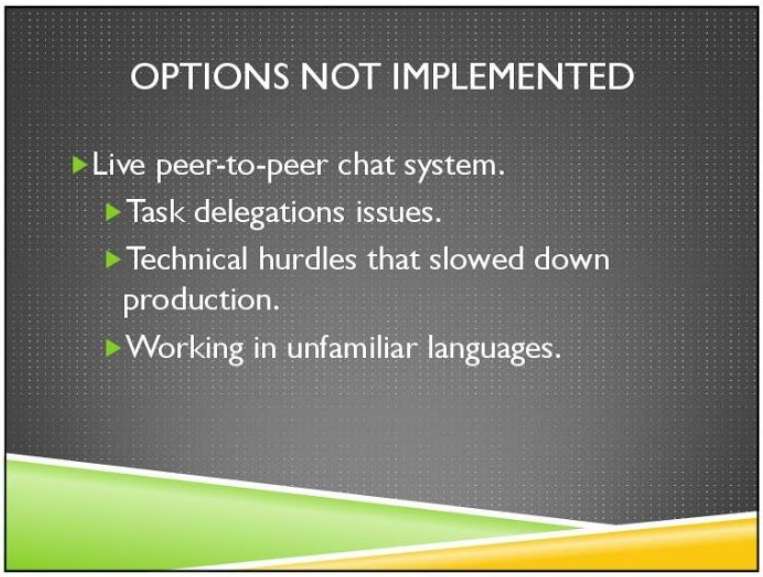
USER INTERFACE

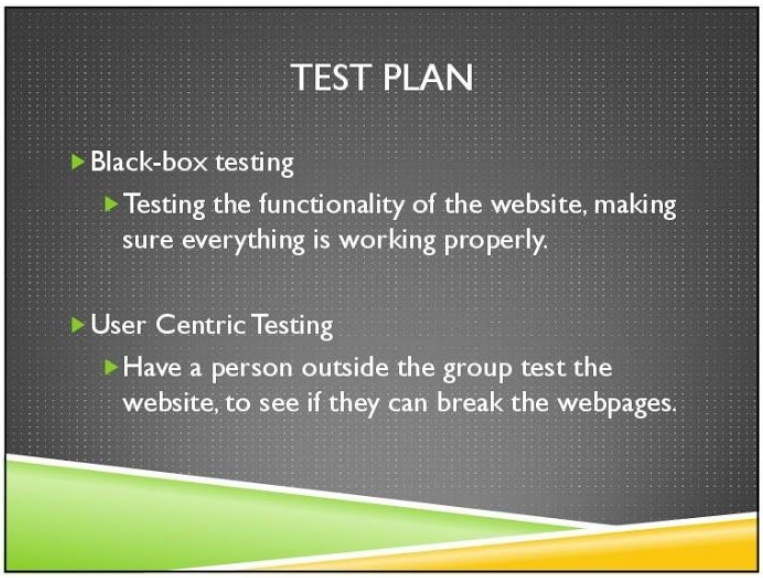


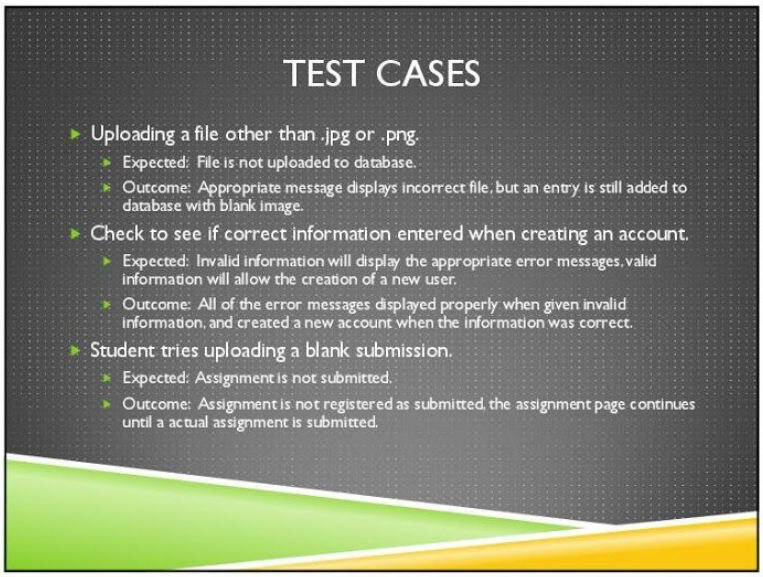
User interface final design compared to initial design.

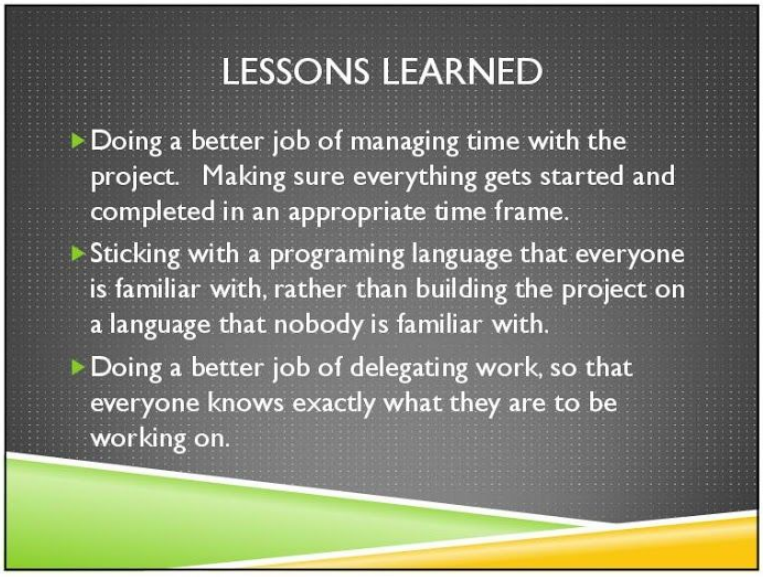
OPTIONS IMPLEMENTED

- Ability to create new accounts.
- Teachers can upload assignments for students.
- Students can complete assignments and upload solution.
- Teachers can grade submissions.


OPTIONS NOT IMPLEMENTED
Live peer-to-peer chat system.
Task delegations issues.
Technical hurdles that slowed down production.
Working in unfamiliar languages.


TEST PLAN
Black-box testing
Testing the functionality of the website, making sure everything is working properly.
User Centric Testing
Have a person outside the group test the website, to see if they can break the webpages.


TEST CASES
Uploading a file other than .jpg or .png.
Expected: File is not uploaded to database.
Outcome: Appropriate message displays incorrect file, but an entry is still added to database with blank image.
Check to see if correct information entered when creating an account.
Expected: Invalid information will display the appropriate error messages, valid information will allow the creation of a new user.
Outcome: All of the error messages displayed properly when given invalid information, and created a new account when the information was correct.
Student tries uploading a blank submission.
Expected: Assignment is not submitted.
Outcome: Assignment is not registered as submitted, the assignment page continues until a actual assignment is submitted.


LESSONS LEARNED
Doing a better job of managing time with the project. Making sure everything gets started and completed in an appropriate time frame.
Sticking with a programing language that everyone is familiar with, rather than building the project on a language that nobody is familiar with.
Doing a better job of delegating work, so that everyone knows exactly what they are to be working on.

BIBLIOGRAPHY

- [1] W. Rankin, "Increasing the Communicative Competence of Foreign Language Students Through the FL Chatroom," *Foreign Language Annals*, vol. 30, no. 4, pp. 542–546, 1997.
- [2] W. S. Lam, "LLT Vol8Num3: SECOND LANGUAGE SOCIALIZATION IN A BILINGUAL CHAT ROOM: GLOBAL AND LOCAL CONSIDERATIONS," *LLTVol8Num3: SECOND LANGUAGE SOCIALIZATION IN A BILINGUAL CHAT ROOM: GLOBAL AND LOCAL CONSIDERATIONS*, Sep-2004. [Online]. Available at: http://llt.msu.edu/vol8num3/lam/. [Accessed: 22-Mar-2016].
- [3] R. Madachy, "COCOMO II - Constructive Cost Model", *Csse.usc.edu*, 2016. [Online]. Available: http://csse.usc.edu/tools/COCOMOII.php. [Accessed: 17- Feb- 2016].
- [4] Ganttproject.biz, "GanttProject: free desktop project management app", 2016. [Online]. Available: https://www.ganttproject.biz/. [Accessed: 17- Feb- 2016].
- [5] I. Cakir, "The Use of Video as an Audio-Visual Material in Foreign Language Teaching Classroom," *The Turkish Online Journal of Educational Technology*, vol. 5, no. 4, Oct. 2006.
- [6] C. Canning-Wilson, "Practical Aspects of Using Video in the Foreign Language Classroom," *The Internet TESL Journal*, vol. 6, no. 11, Nov. 2000.

DEMO

Technical Documentation

Installation Documentation

Tashi talk can run on any webserver that supports the basic web LAMP (Linux, Apachi, MySql, PHP) stack. Simply drag the files into the directory served by Apache (currently typically var/www/html). The paths are relative and can be located in any subdirectory if desired. To setup the server creation of a database is required. The database info should be entered in the following files.
/userSystem/userBase.php
/TeacherDashboard/mysqli_connect.php
/userDashboard/mysqli_connect.php

Video, video display, folder creation, setting required permissions is all handled with standard web languages as mentioned above requiring no additional configuration.

The database should be initialized with the following database creation commands which can be run via a tool like MyPHPAdmin which is optional.

SQL DATABASE SETUP CODE:

```
-- phpMyAdmin SQL Dump
-- version 4.0.10deb1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Apr 26, 2016 at 08:13 PM
-- Server version: 5.5.46-0ubuntu0.14.04.2
-- PHP Version: 5.5.9-1ubuntu4.14

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `goga_database`
--
CREATE DATABASE IF NOT EXISTS `goga_database` DEFAULT CHARACTER SET
latin1 COLLATE latin1_swedish_ci;
USE `goga_database`;

-- --------------------------------------------------------

--
-- Table structure for table `Assignments`
--
-- Creation: Apr 25, 2016 at 11:23 PM
--

DROP TABLE IF EXISTS `Assignments`;
CREATE TABLE IF NOT EXISTS `Assignments` (
  `assign_id` mediumint(9) NOT NULL AUTO_INCREMENT,
  `teacher_id` mediumint(9) NOT NULL,
  `class_id` mediumint(9) NOT NULL,
  `prompt_loc` varchar(2000) NOT NULL,
  `prompt_name` varchar(50) NOT NULL,
  `prompt_text` varchar(400) NOT NULL,
  PRIMARY KEY (`assign_id`),
  KEY `teacher_id` (`teacher_id`),
  KEY `class_id` (`class_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;

--
-- RELATIONS FOR TABLE `Assignments`:
--   `teacher_id`
```

```
--       `Teachers` -> `teacher_id`
--   `class_id`
--       `Classes` -> `class_id`
--


--
-- Dumping data for table `Assignments`
--

INSERT INTO `Assignments` (`assign_id`, `teacher_id`, `class_id`, `prompt_loc`,
`prompt_name`, `prompt_text`) VALUES
(10, 100, 100, './../uploadedMaterial/teachers/JPN_101/pick.jpg', 'Say Hello', 'dsafasdf adf
asd asd fasd f'),
(11, 100, 100, './../uploadedMaterial/teachers/JPN_101/Pasted image at 2016_04_25 10_29
PM.png', 'Your Schedule', 'Where are you going to be Saturday?\r\nWhat time is dinner
on Wednesday?'),
(14, 100, 100, './../uploadedMaterial/teachers/JPN_101/paper.jpg', 'Popular Locations', 'Say
popular locations'),
(15, 100, 100, './../uploadedMaterial/teachers/JPN_101/konichiwa2.jpg', 'Greetings', 'say
hello in japanese');

-- --------------------------------------------------


--
-- Table structure for table `Classes`
--
-- Creation: Apr 23, 2016 at 09:21 PM
--

DROP TABLE IF EXISTS `Classes`;
CREATE TABLE IF NOT EXISTS `Classes` (
  `class_id` mediumint(9) NOT NULL AUTO_INCREMENT,
  `class_name` varchar(50) NOT NULL,
  PRIMARY KEY (`class_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=101 ;

--
-- Dumping data for table `Classes`
--
```

```
INSERT INTO `Classes` (`class_id`, `class_name`) VALUES
(100, 'JPN_101');

-- --------------------------------------------------------

--
-- Table structure for table `studentgrade`
--
-- Creation: Apr 26, 2016 at 05:49 PM
--

DROP TABLE IF EXISTS `studentgrade`;
CREATE TABLE IF NOT EXISTS `studentgrade` (
  `sub_id` mediumint(9) NOT NULL DEFAULT '0',
  `subattempt_loc` varchar(2000) DEFAULT NULL,
  `userNumber` mediumint(9) NOT NULL,
  `class_id` mediumint(9) NOT NULL,
  `grade` varchar(5) NOT NULL,
  `first_name` varchar(255) NOT NULL,
  `last_name` varchar(255) NOT NULL,
  `class_name` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `studentgrade`
--

INSERT INTO `studentgrade` (`sub_id`, `subattempt_loc`, `userNumber`, `class_id`,
`grade`, `first_name`, `last_name`, `class_name`) VALUES
(5, './../submissions/students/JPN_101/11/11/video.webm', 6, 100, 'A', 'Student', 'Student',
'JPN_101');

-- --------------------------------------------------------

--
-- Table structure for table `Submissions`
--
-- Creation: Apr 26, 2016 at 07:30 AM
```

```
--

DROP TABLE IF EXISTS `Submissions`;
CREATE TABLE IF NOT EXISTS `Submissions` (
  `sub_id` mediumint(9) NOT NULL AUTO_INCREMENT,
  `userNumber` mediumint(9) NOT NULL,
  `class_id` mediumint(9) NOT NULL,
  `teacher_id` mediumint(9) NOT NULL,
  `subattempt_loc` varchar(2000) DEFAULT NULL,
  `grade` varchar(5) NOT NULL,
  PRIMARY KEY (`sub_id`),
  KEY `userNumber` (`userNumber`),
  KEY `class_id` (`class_id`),
  KEY `teacher_id` (`teacher_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=65 ;

--
-- RELATIONS FOR TABLE `Submissions`:
--   `userNumber`
--       `users` -> `userNumber`
--   `class_id`
--       `Classes` -> `class_id`
--   `teacher_id`
--       `Teachers` -> `teacher_id`
--


--
-- Dumping data for table `Submissions`
--

INSERT INTO `Submissions` (`sub_id`, `userNumber`, `class_id`, `teacher_id`,
`subattempt_loc`, `grade`) VALUES
(5, 6, 100, 100, './../submissions/students/JPN_101/11/11/video.webm', 'A'),
(63, 5, 100, 100, './../submissions/students/JPN_101/10/10/video.webm', ''),
(64, 10, 100, 100, './../submissions/students/JPN_101/15/15/video.webm', '');

-- --------------------------------------------------------

--
```

```
-- Table structure for table `Takes`
--
-- Creation: Apr 23, 2016 at 09:21 PM
--

DROP TABLE IF EXISTS `Takes`;
CREATE TABLE IF NOT EXISTS `Takes` (
  `class_id` mediumint(9) NOT NULL,
  `userNumber` mediumint(9) NOT NULL,
  PRIMARY KEY (`class_id`,`userNumber`),
  KEY `userNumber` (`userNumber`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `Takes`:
--   `userNumber`
--       `users` -> `userNumber`
--   `class_id`
--       `Classes` -> `class_id`
--


--
-- Dumping data for table `Takes`
--

INSERT INTO `Takes` (`class_id`, `userNumber`) VALUES
(100, 2);

-- --------------------------------------------------------


--
-- Table structure for table `Teachers`
--
-- Creation: Apr 23, 2016 at 09:21 PM
--

DROP TABLE IF EXISTS `Teachers`;
CREATE TABLE IF NOT EXISTS `Teachers` (
  `teacher_id` mediumint(9) NOT NULL AUTO_INCREMENT,
```

```
  `teacher_fname` varchar(30) NOT NULL,
  `teacher_lname` varchar(30) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`teacher_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=101 ;

--
-- Dumping data for table `Teachers`
--

INSERT INTO `Teachers` (`teacher_id`, `teacher_fname`, `teacher_lname`, `password`)
VALUES
(100, 'Matthew', 'Canton', 'pass');

-- --------------------------------------------------------

--
-- Table structure for table `Teaches`
--
-- Creation: Apr 23, 2016 at 09:21 PM
--

DROP TABLE IF EXISTS `Teaches`;
CREATE TABLE IF NOT EXISTS `Teaches` (
  `class_id` mediumint(9) NOT NULL,
  `teacher_id` mediumint(9) NOT NULL,
  PRIMARY KEY (`class_id`,`teacher_id`),
  KEY `teacher_id` (`teacher_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `Teaches`:
--   `class_id`
--       `Classes` -> `class_id`
--   `teacher_id`
--       `Teachers` -> `teacher_id`
--

-- --------------------------------------------------------
```

```
--
-- Table structure for table `users`
--
-- Creation: Apr 23, 2016 at 09:21 PM
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `first_name` varchar(255) NOT NULL,
  `last_name` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `activated` tinyint(1) NOT NULL DEFAULT '0',
  `userNumber` mediumint(9) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`userNumber`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`first_name`, `last_name`, `password`, `email`, `activated`,
`userNumber`) VALUES
('Matthew', 'Canton',
'$2y$10$cNEkVqF6daPpn/Pu8VzBTuU0tXN19WOLs46MfkzzS28UkVzHeM4oa',
'matt@gmail.com', 0, 1),
('Test', 'Test',
'$2y$10$EoYbUdg6RhlqvFiGN26MyOruijz989hAR5hIXZiH151p6Wng3rFGa',
'test@test.com', 0, 2),
('Test', 'Test',
'$2y$10$78ZLoaBCn.SnkZdCXD/iQehCnkD/B00AHRJvn/o.AxdErGICEOwFe',
'matta@gmail.com', 0, 3),
('Test', 'Test', '$2y$10$hJFieUf0yk/z29mzw3Si.u/C17e2YtBmCs96Lp9DI2BzfcsZuglW6',
'test@test1.com', 0, 4),
('test', 'test',
'$2y$10$1QdQ/RlOXV0eK/OAsDfLU.aPAIHmXRms9ZOs/x9o617j3OwECHfo6',
'test@test.test', 0, 5),
```

```
('Student', 'Student',
'$2y$10$4v5AVUiPvBy14PCJf7NtW.gxnA8cRXlKjS2Qa735XgwHJxkAcN7r6',
'student@student.com', 0, 6),
('Apple', 'Apple',
'$2y$10$0bm7ZwC28viTFy1hHaSqz./j5HANW4llBhLlyXiqGHtqoWeTuoUXK',
'apple@apple.apple', 0, 7),
('Darryl', 'Papke',
'$2y$10$pBXYEBJLqLQkGrLG55Mwru1xqw1FlpS1EtwT0zURQO5VyHCxN1pXG',
'darrylpapke2@gmail.com', 0, 8),
('George', 'George',
'$2y$10$ptMBJs23EqzmNpGRh7GOIet/9RiNatCMkJ1xO8GPGgwiZ/zr7KPHe',
'George@George.George', 0, 9),
('Jack', 'Stone',
'$2y$10$EZK2ufUnOakkcCWCb2YWduDIqZWWEgk/9lwuJLqwgzC6hNAYh5FsC',
'JackStone@Jackstone.Jackstone', 0, 10),
('Garrett', 'Grimsley',
'$2y$10$KO1cvv8d8Kx68gbWsyoMNONIzIVlHhhrq/JoBvT0lbTsbt3JGifE2',
'gag8520@uncw.edu', 0, 11);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `Assignments`
--
ALTER TABLE `Assignments`
  ADD CONSTRAINT `Assignments_ibfk_1` FOREIGN KEY (`teacher_id`)
REFERENCES `Teachers` (`teacher_id`) ON DELETE CASCADE ON UPDATE
CASCADE,
  ADD CONSTRAINT `Assignments_ibfk_2` FOREIGN KEY (`class_id`) REFERENCES
`Classes` (`class_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `Submissions`
--
ALTER TABLE `Submissions`
```

```
  ADD CONSTRAINT `Submissions_ibfk_1` FOREIGN KEY (`userNumber`)
REFERENCES `users` (`userNumber`) ON DELETE CASCADE ON UPDATE
CASCADE,
  ADD CONSTRAINT `Submissions_ibfk_2` FOREIGN KEY (`class_id`) REFERENCES
`Classes` (`class_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `Submissions_ibfk_3` FOREIGN KEY (`teacher_id`)
REFERENCES `Teachers` (`teacher_id`) ON DELETE CASCADE ON UPDATE
CASCADE;

--
-- Constraints for table `Takes`
--
ALTER TABLE `Takes`
  ADD CONSTRAINT `Takes_ibfk_1` FOREIGN KEY (`userNumber`) REFERENCES
`users` (`userNumber`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `Takes_ibfk_2` FOREIGN KEY (`class_id`) REFERENCES
`Classes` (`class_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `Teaches`
--
ALTER TABLE `Teaches`
  ADD CONSTRAINT `Teaches_ibfk_1` FOREIGN KEY (`class_id`) REFERENCES
`Classes` (`class_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `Teaches_ibfk_2` FOREIGN KEY (`teacher_id`) REFERENCES
`Teachers` (`teacher_id`) ON DELETE CASCADE ON UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Source Code Documentation

A. Each class has a header comment block at the top of each file. Header comment blocks include the author's name, date, filename, the file's purpose, where it fits in the system, expected input and possible output, and expected extensions/revisions.
B. Each module/method has a comment block above it that describes the author, when it was written/revised, where/when it is called, expected input and possible output.
C. Line comments are used to explain a subset of instructions inside a module/method to describe declarations and blocks of code in which the purpose in not obvious.
D. Other Standards
   •camelCase for page names longer than one word
   •underscore for PHP variables
   •Variable names should describe their purpose. Exception for loop counting variables.
   •always use brackets in control statements
   •use clear and concise comments
   •keep most PHP at the beginning of the document, placing other PHP tags in document only when necessary.

Below is a code example of our proper documentation standard.

```php
<?php
//=====================================================================
// author-Matthew Canton
// Apr 26 2016
//
// teacherPrompt.php
// prompts user to enter information for entry to database
// connects to index.php in teacherDashboard
// expected input: Information to Create a Prompt
// possible output: Information to Create a Prompt sent to teacherPromptDisplay.php
//=====================================================================
?>
<!DOCTYPE html>
<html lang="en">
    <?php # index.php
        require_once('mysqli_connect.php');
    ?>
    <head>   <!--     <script src="script.js"></script> -->
        <meta charset="utf-8">
        <title>TashiTalk</title>
        <link rel="stylesheet" href="
        https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <script src="../siteWideResources/jqery-2.2.2.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"
        ></script>
        <link rel="stylesheet" href="resources/css/style.css">
        <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    </head>
    <body>
        <!-- Nav bar -->
        <?php include 'navbar.php' ?>
        <br>
        <br>
        <?php
            //set default value of variables for initial page load
            if (!isset($teacher_id)) { $teacher_id = ''; }
            if (!isset($class_id)) { $class_id = ''; }
            if (!isset($prompt_name)) { $prompt_name = ''; }
            if (!isset($prompt_text)) { $prompt_text = ''; }
        ?>
            <style>
                label{padding-left: 2cm;}
                h3{padding-left: 2cm;}
                textarea{margin-left: 200px;}
            </style>
            <br>
            <br>
            <h3>Please Enter Information to Create a Prompt</h3><br>
            <!--
                    author: Matthew Canton
                    Apr 26 2016
                    used once teacherPrompt.php is opened
                    expected input: user input for prompt
                    possible ouput: information for prompt sent to teacherPromptDisplay
            -->
            <form action="teacherPromptDisplay.php" method="post" padding-left: 2cm; enctype=
            "multipart/form-data">
```

```
                <div id="prompt_form">

                    <label>Your Teacher ID:</label>
                    <input type="text" name="teacher_id" value="<?php echo htmlspecialchars(
                    $teacher_id); ?>"><br>

                    <label>Appropriate Class ID:</label>
                    <input type="text" name="class_id" value="<?php echo htmlspecialchars(
                    $class_id); ?>"><br>

                    <label>Prompt Title:</label>
                    <input type="text" name="prompt_name" value="<?php echo htmlspecialchars(
                    $prompt_name); ?>"><br>

                    <label>Prompt Description:</label><br>
                    <textarea name="prompt_text" rows="5" cols="40"><?php echo
                    htmlspecialchars($prompt_text);?></textarea><br>

                    <!-- Prompts user to add file -->
                    Select image to upload:
                    <input type="file" name="fileToUpload" id="fileToUpload">
                    <input style="margin-left:100px" type="submit" name="submit" value=
                    "Submit"><br>
                </div>
            </form>

        <!-- Scripts -->
        <script type='text/javascript' src="../siteWideResources/jquery-2.2.2.min.js"
        ></script>
        <script type='text/javascript' src="
        https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
    </body>
</html>
```