# CptS 489 Team Project App Descriptions

Fall, 2020

This course project provides you with the opportunity to participate in a multi-institutional research project that is exploring new approaches to teaching skills in legacy code development. As one of the first classes to participate in this project, some teams will be "bootstrapping" legacy code bases that students in subsequent CptS 489 classes will build on. One team (Recipe) will start with a legacy code base from the last offering of this course. Several epics (i.e., collections of user stories) will be defined for each project app.

## Online Studio-Based Learning Environment (OSBLE)

The OSBLE environment used in this class is a legacy software environment first developed in 2007. It uses ASP .NET on the back end (an MVC framework) and mostly JQuery on the front end. While its code base is relatively robust, it has become increasingly difficult to maintain, owing to the relatively old dev stack used.

To modernize OSBLE, this project reimplements the OSBLE in React, Node, and MongoDB. A number of epics will be defined to address different aspects of the new system, including user management, courses, assignments, the gradebook and the email system.

## Kliks

Kliks is a classroom response system implemented by Professor Emeritus Carl Hauser in Python, HTML and CGI (I believe). While the back-end system is solid, anyone who has used the front-end user interface is aware of its limitations. This project aims to reimplement, improve and expand the Kliks environment so that it is more useful and easier to use. The new Kliks will be fully integrated into OSBLE, so it can minimally share OSBLE's user management system. A number of epics will be defined to address different aspects of Kliks, including the student and instructor interfaces.

## AccreditAid

Every six years, EECS needs to renew its accreditation with ABET. This is a cumbersome process that involves many coordinated activities across both EECS and the College of Engineering and Architecture. The AccreditAid app will be designed to help EECS coordinate and complete the activities and documents required for accreditation. Several epics will be defined to address different aspects of the system:

- User management

## SpeedScore LIVE

On a recent sabbatical, your instructor developed SpeedScore Live, the world's first live tournament scoring system for speedgolf. The system was implemented using Google Sheets and Google Apps Script on the back end, and JQuery on the front end. Since 2017, the app has been used to score over 20 tournaments worldwide.

While the current system works fairly well, its implementation in Google Sheets and Google Apps Script has limited its scalability and responsiveness. In this project, you will reimplement SpeedScore Live in React, Node, and MongoDB while improving and expanding its functionality. Several epics will be defined to address the following aspects of the system:

- User management
- Player profile system
- Scorer interface
- Tournament director interface
- Live leaderboard

## SpeedScore TRACK

Throughout this class, I have used a personal speedgolf app to demonstrate course concepts and techniques. In this class, I have worked on a relatively small fraction of the overall functionality that I originally envisioned for the app. The app is attended to make it easy for speedgolfers to automatically log their rounds in great detail; to track the overall progress in the sport; and to allow them to share their rounds and progress with a community of speedgolfers. The round functionality will ultimately support geo-tracking, voice input, heartrate monitoring and perhaps even auto inferencing of round events through interpretation of accelerometer data. In addition, the app will necessarily maintain a database of courses with detailed geodata to support enhanced tracking. Several epics will address different aspects of this app, including

- User management
- Social feed
- Round logging
- Round tracking
- Golf Course input and management

## SpeedScore PLAY

The third app in the SpeedScore suite endeavors to be the "Uber" of the speedgolf world. The idea is to

- Course operator interface
- Speedgolfer interface
- Golf Course input and management (could be combined with Speedgolf TRACK)

## EECS Course Scheduler

This idea came out of the project brainstorming exercise. The idea is to build an app to help EECS students to navigate through their degree programs. An EECS student could log in to the app, specify their major, view a graph of their course requirements and options, and track their progress through their degree program. The app would interface with WSU's course schedule to provide students with specific courses offerings in upcoming semesters that can be used to fulfill degree requirements. Thus, the tool would allow students to understand the specific requirements of their degree program, track their progress and plan their future semesters. The following epics address different aspects of the app:

- User management (probably best to use WSU OAuth)
- Interface for administrators (e.g., academic advisers) to enter degree requirements
- Interface for students to view degree requirements
- Interface for students to enter progress (perhaps this could be done automatically through the registrar's grade reports?)
- Interface for students to plan future semesters (should be integrated with WSU course schedule)

## Recipe App

This idea came out of a previous class. The idea is to build a comprehensive app for recipe management. Users can easily add recipes through web links, copy-paste or even screen grabs. The app can use OCR to import apps that come from images. The app can allow the user to search and filter recipes based on meal, food, and dietary preferences. Eventually, it can support grocery management, so that recipes could be suggested based on a user's current inventory of ingredients. Bar code scanning could be used to enable users to add grocery items to their inventory. In addition, the app could build shopping lists based on what the user wants to eat on a given day or in a given week. The following epics address different aspects of the app:

- User management
- Interface for collating recipes from different sources (including the use of OCR)
- Interface for grocery management (adding/removing grocery items, possibly through the use of a barcode scanner)
- Interface for suggesting recipes based on grocery inventory
- Interface for building shopping lists based on recipe preferences

## Fantasy Football

The idea is to gamify fantasy football. Most apps don't keep track of the stats of the fantasy players themselves. Since an API already exists for football players' stats, the back end for this app would handle league information. As a fantasy user, a user could see their best and worst fantasy week, their win/loss ratio over the season, who they've lost to the most (if in multiple seasons), etc. These achievements/accomplishments can be attached to a user's account so people in public leagues can see how good, or bad, users' seasons have been.

## Twitch Client

Twitch is an online streaming platform. This project will build upon and extend features found in twitch's web chat client. The base functionality will include login/Oauth with Twitch (encrypted local credential storage); multi-Channel chat navigation; send chats (if user is logged in); and additional novel chat features such as spamming (allowing the user to define a message to spam in chat and then spam until user gets timed out). There is a possibility the Twitch player itself could be embedded in the app.

## Local Music

The idea is to help people to stay more informed about local music venues and events. A user would be able to create a profile and then search for venues near them. Then they could subscribe to those venues and receive notifications about events. They would also have a feed which they could scroll through which would show upcoming/newly posted events at the venues they are subscribed to. There is a possibility that we would add a feature to friend other users so that you could share upcoming events with them. Then, the venues themselves could create a different type of account in which they could post their events and setup a landing page with information about their venue. In a fully fleshed out app they may even be able to buy tickets to the event from the website.