The Frank Innovation Zone (FIZ) provides resources for Washington State University students to complete personal or academic projects, primarily by housing manufacturing equipment within various fabrication studios, or makerspaces. These resources come at little to no cost for students. The FIZ tasked Team Tethys to develop a kiosk system displaying customized question sets to kiosks deployed throughout various makerspaces. The data obtained by the kiosk system could then be utilized to leverage funding from the university or donors by providing tangible documentation of the FIZ’s utility and value.

Makerspace Kiosk System
Sponsor: Frank Innovation Zone (FIZ)
Mentor: Dr. Aaron Crandall and Kirk Reinkens
Kalana Sahabandu, Paul Brennan, Patrick Keppler, Angel Castañeda

The makerspace kiosk system was decomposed into three subsystems.
- Website Frontend
- Kiosk Frontend
- Database Backend

Each subsystem warranted the utility of unique technologies to accomplish its required functionality.

Website Frontend
- HTML/CSS
  - JavaScript
  - jQuery
  - Bootstrap
  - Chart.js

Kiosk Frontend
- Java
  - JSON
  - Toast
  - Timer
  - FileWriter
- Android Studio
  - Android Volley

Database Backend
- Gunicorn
  - Flask
  - SQL Alchemy
  - Python
  - Pawslib
- HTTPS

kiosk.eecs.wsu.edu was configured to the IP address of a server hosted by the College of Electrical and Engineering and Computer Science designed to field web queries and manage the database for the makerspace kiosk system. Nginx was utilized to serve the website, while Gunicorn was configured to run the database application. Eight database tables were established along with 30 routes to navigate the backend and access stored data or write new information the database.

Tables
The primary tables included a kiosk table, where each row in the table maintains information for a single kiosk and has access to one or more question set table rows. Question sets are a collection of unique question ID’s, each question ID being associated with an individual question within the question table.

Routes
- 30 routes, or web queries, were established with specific functionality accomplished by each to communicate data between the database and kiosk or website frontend. The web queries are conducted over HTTPS, indicating a direct and secure connection has been established between the client and the server.

- A website for facility directors to create question sets, manage kiosks and view data
- An Android application to display question sets and field answers via touchscreen interactions
- A database maintaining all information allocated by the kiosk system

The main function of the kiosk application is to present customized question sets and their corresponding answers to the user.

A kiosk is initially configured to be a member of an existing facility and given a specific makerspace location. A question set is then assigned to the kiosk, which queries the database every five seconds to ensure the active question set is being displayed. Certain responses warrant follow-up questions, so specific answers will result in more questions, whereas others will conclude the question set. The selected response is then sent to the backend for database storage, where statistics are managed regarding responses per question.

The version control software used for managing the development of the kiosk system was GitLab, which provides Continuous Integration (CI) and Continuous Deployment (CD) frameworks. Combined with Docker, we were able to incorporate automated tests into our CI pipeline which execute after every commit.

The CI pipeline contains a majority of established tests for the kiosk application and backend routes. The technologies used in these test environments are the Android Software Development Kit testing tools, JUnit for kiosk testing and Newman command line tools for the backend routes. Performance testing was also conducted, revealing fast system communication response times.

Upon successful testing within the specified deployment branches, an application bundle is automatically pushed to the Google Play store under one of three classifications (closed alpha prototype, open beta prototype or official product release).

A complete kiosk system was delivered, featuring an Android application deployed to tablets operating as kiosks which securely communicate with a server configured to host the kiosk management website and system database. All vital functionality is validated by automated testing frameworks, which deploy the newest version of the kiosk application to the Google Play store.