601.310: Software for Resilient Communities  
Spring 2018 (3 credits)

Instructors  
Amy Babay: babay@cs.jhu.edu, http://dsn.jhu.edu/~babay/  
Yair Amir: yairamir@cs.jhu.edu, http://www.cs.jhu.edu/~yairamir/

Prerequisites  
601.220 (Intermediate Programming) and 601.226 (Data Structures)  
Instructor permission required

Course Description  
This is a project-based course focusing on the design and implementation of practical software systems. Students will work in small teams (2-4 students per team) to design and develop useful open-source software products that support our communities.

Students will be paired with community partners and will aim to develop software that can be used after the course ends to solve real problems facing those partners today. Instructors will connect with the community partners and determine viable project areas prior to the course start. Students will meet with their community partners to analyze the challenges in their project area, agree on a concrete target project outcome, and gather requirements for their project. Based on these requirements, students will design and implement open-source software systems.

Projects for the pilot offering of the course will focus on communities at four levels:
1. JHU Community: improving the registration process and facilitating communication between students and advisors regarding course scheduling, working with the Semester.ly team and the JHU registrar
2. Reservoir Hill Neighborhood Community: enabling self-service discovery and scheduling of community center resources via a web-based calendar and scheduling system, working with the St. Francis Neighborhood Center
3. Baltimore City Community: providing visibility into city park usage patterns by analyzing existing camera data to answer questions about how many people are visiting the parks and when, working with the Parks & People Foundation
4. US Community: improving the resilience of the power grid through secure and intrusion-tolerant open-source SCADA systems

Course Goals  
Students will improve their ability to:
1. Analyze real-world problems and apply computing knowledge to devise appropriate software solutions.
2. Program effectively on a team
3. Communicate effectively (both orally and through writing) with both technical and non-technical audiences to understand requirements, develop designs, and present solutions
4. Independently learn to use (and potentially contribute to) existing software libraries and tools