**Problem**

- BreazeHome is an academic project originated and lead by Dr. Masoud Sadjadi to help his students learn Agile Software Development in the context of a real-estate application that provides property information and services for home buyers, sellers, renters and realtors.
- BreazeHome would like to provide its users with County Records data in order to gain more insight on a property and prevent any surprise impediments.
- One of the possible impediments are Property Liens, which have the potential to significantly delay the home buying process. Property Liens must be satisfied by the previous homeowner, or transferred to the new homeowner before a property can be sold.

**Solution**

- Create a new BreazeHome REST API endpoint for Miami Dade County Records.
- Develop a script that will automatically update BreazeHome County Records data daily.
- Add a new section labeled “Investment Information” to the Property Detail page.
- Allow BreazeHome users to obtain lien data through the Property Detail page, granting them the ability to catch these potential delays early in the home buying process.
- This feature will allow potential homeowners and investors to decide whether or not a home is worth pursuing in a timely manner.

**Requirements**

- Create a new BreazeHome REST API endpoint to serve Miami Dade Records.
- Import the County Records data provided by the Miami-Dade CSV every day through the use of an automated script.
- During the import phase, place any useful Lien records in a secondary table, called Liens.
- When a user enters a Property’s Detail page, a lien search will be performed on the currently viewed property.
- If any liens are found on the current property, display information pertaining to each lien found in a results table located in the Investment Information section of the Property Detail page.
- If no liens are found on the current property, notify the user.

**Current System**

- The current system does not have any access to County Records.
- The current system does not contain any County Record information.
- Miami Dade FTP-API provides a paid service which supplies its users with a daily CSV file containing all new and updated County Records for that day.
- Miami Dade FTP-API has a 2 call per day limit.

**Implementation**

- Database: PostgreSQL
- Data Import Scripts: Python
- Web API: Django REST Framework (Python)
- Desktop Frontend: NodeJS, AngularJS, HTML, Bootstrap, SASS
- Model-View-Controller pattern with Client-Server architecture.
- Testing: Selenium, Django Unit Test
- OS: Ubuntu Linux

**System Design**

- Presentation Tier
  - BreazeHome Mobile Application
  - BreazeHome REST Interface API Services
  - BreazeHome CGI Interface Services

- Delivery Tier
  - BreazeHome Website
  - 3rd Party API

- Data Aggregation Tier
  - Chief API

- Fundamental Data Tier
  - Data filtering system
  - 3rd party Database

**Requirements**

- Create a new BreazeHome REST API endpoint to serve Miami Dade Records.
- Import the County Records data provided by the Miami-Dade CSV every day through the use of an automated script.
- During the import phase, place any useful Lien records in a secondary table, called Liens.
- When a user enters a Property’s Detail page, a lien search will be performed on the currently viewed property.
- If any liens are found on the current property, display information pertaining to each lien found in a results table located in the Investment Information section of the Property Detail page.
- If no liens are found on the current property, notify the user.

**Verification and Testing**

- Frontend: Selenium WebDriver
- Backend: Django APITestCase

**Screenshots**

- Liens Found
- Liens Not Found

**Liens Search Backend Sequence Diagram**

**Liens Search Frontend Sequence & Class Diagrams**

**Acknowledgement**

The material presented in this poster is based upon the work supported by Dr. Masoud Sadjadi and Yuzhou (Aaron) Feng. I am thankful to the help that I received from my group members Alex Dubuisson, Davi Guerra, Fernando Serrano, Lyda Caballero and Ronny Alfonso.