Service

Scrumbags Project Proposal

Anthony Ciminello | Nicholas Petty | Joshua Rodriguez-Santiago

Raquel Rosa | Gavin Wolf

Executive Summary

We will be creating a web application called Service, which will allow people to notify property owners of places and things that need to be serviced. The application's goal will be to streamline and crowd-source building maintenance. We feel that a great deal of the spaces we inhabit and pass through in our daily lives are in need of care. With Service, ordinary people will be able to reach out to property managers, and let them know where work is needed. Our users will be the general public, and anyone who claims ownership of or responsibility for a property.

The application will stand out in its simplicity. Anyone can join as a regular user and submit "service requests," which are the primary component of the system. A service request is a short note that includes a picture and the location of something that needs to be fixed. The "property manager" is special user that take responsibility for a building or location. Property managers are required to submit verification documents before they can claim a property, but once verified, they can respond to service requests. Again keeping options simple, the response can either be agreeing to fix the problem, or not. All users will be able to view all service requests and resolutions.

Service connects us through a better environment.

Competitive Analysis

There are other products that perform this function. We differentiate by offer a simple user experience.

Overview and Scenarios

The system will be used by the general public and property managers. Normal users create accounts, create repair requests, and track their requests. Property managers create accounts and respond to repair requests for their properties. Additionally, an administrator will moderate requests and verify property managers.

Functional Requirements

1. Create account
2. Log in
3. Log out
4. Create repair request
5. View repair requests
6. Resolve repair requests

Non-functional Requirements

1. Mobile responsive
2. 1000 concurrent users
3. Database for 10,000 repair requests
4. Security via account system

System Development Infrastructure

1. WebRatio
2. Cameo Enterprise Architecture
3. GitHub
4. SourceTree
5. Trello
6. Circuit

Team

- Product Owner: Nick Petty
- Scrum Master: Anthony Ciminello
- Back-end Developer: Joshua Rodriguez-Santiago
- Front-end Developer: Raquel Rosa
- UI/UX Designer: Gavin Wolf
    - [testing git]

Deliverables

1. Proposal: October 24
2. BPMN Modeling
3. Test Set Document
4. Final Presentation