Worcester Polytechnic Institute
Digital WPI

Major Qualifying Projects (All Years) | Major Qualifying Projects

April 2013

Utilizing Smartphone Resources with Homesite Insurance

Billy Sean Estrella
Worcester Polytechnic Institute

Gregory Scott Wheeler
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/mqp-all

Repository Citation

Estrella, B. S., & Wheeler, G. S. (2013). *Utilizing Smartphone Resources with Homesite Insurance*. Retrieved from https://digitalcommons.wpi.edu/mqp-all/2750

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

Utilizing Smartphone Resources with Homesite Insurance

A Major Qualifying Project
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
By

Billy S. Estrella (CS)

Jay T. Miller (CE)

Gregory S. Wheeler (CS)

Date: April 25, 2013

Professor Emmanuel Agu, Primary CS Advisor & CE Co-Advisor
Professor Guillermo Salazar, Primary CE Advisor
Professor Matthew Ward, CS Co-Advisor & Secondary CE Co-Advisor

Project No. CS-EOA-0021
Project No. CE-GFS-0021

1. Mobile apps
2. Smartphones
3. Home insurance

This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. The opinions expressed herein are those of the student authors and do not necessarily reflect the policies or views of the sponsoring agency or its personnel. WPI routinely publishes these reports on its web site without editorial or peer review.

Abstract

The process of providing real estate information about a property, a central component of purchasing homeowner's insurance, is a lengthy, complicated procedure that has a large capacity for error on the part of the customer providing the information. Because of advances in information technology, including the storage, processing, and presentation of information of all kinds, this process can be handled in ways that are faster, less prone to error, and easier to use in the absence of specialist knowledge. The goal of this project was to explore these technological advances in order to identify strategies that could be used to achieve the aforementioned improvements. We accomplished this goal by creating mock prototypes of a smartphone application designed to autonomously collect real estate information in such a way that a casual user of the app could handle the process with little difficulty and with reliable results. By implementing our recommendations, homeowner's insurance providers such as Homesite Insurance can streamline their processes for selling insurance policies to clients.

Authorship Page

This project has been a collaborative effort between Billy Estrella, Jay Miller, and Greg Wheeler of Worcester Polytechnic Institute. These students worked individually and as a group to conduct research, develop and test methods, and compile the report. Over the duration of this project, each team member contributed a fair and significant amount of time, effort, and substance to the project's successful completion.

The coding of the project's prototypes and its associated research was handled primarily by Billy Estrella.

The testing of the prototypes and the acquisition of the necessary civil engineering material was handled mainly by Jay Miller.

The composition of the report and assorted presentation materials was handled principally by Greg Wheeler.

Executive Summary

During the past decade, insurance has become a significant issue that many Americans are taking very seriously, both in terms of whether or not an individual chooses to purchase insurance and how the insurance infrastructure as a whole is handled (Iglehart, 2002). One of the types of insurance being considered by many Americans is homeowner's insurance. As with other types of insurance, the process of acquiring a quote regarding an insurance policy's cost, either for the customer's final purchase or simply as a means of comparing the policies offered by different companies, is a critical part of purchasing a homeowner's insurance policy. By implementing information-processing strategies made possible by technological advances, such as the smartphone and electronic information resources, the process of providing a quotation for a homeowner's insurance policy's cost can be streamlined and otherwise improved for all concerned.

One of the companies involved in the insurance business is Homesite Insurance, the sponsor for our project. Some years ago, Homesite Insurance took advantage of improving technology to create an online form for purchasing a homeowner's insurance policy. Since then, technology has progressed even further, most notably in the advancement of the modern smartphone. In response, Homesite Insurance decided to develop a new electronic form designed to be used within a smartphone application. Before they began developing their application, however, they asked our team to explore the technologies available to modern smartphones in order to identify ways that the smartphone version of the form could be improved over the version designed for personal computers. As such, the goal of our project was to generate ideas for the improvement of the process of acquiring a homeowner's insurance policy quote through the use of modern technological resources. After some debate, we decided to focus on the

autonomous collection of the data necessary to calculate the cost of an insurance policy. This feature, when implemented, would enable their customers to receive quotes on homeowner's insurance policies without having to personally provide as much information as they do currently. This improvement would also allow the overall accuracy of the information provided for the calculation of the quote to improve, as the process would become less dependent on the potential client being knowledgeable about the property in question and about real estate in general.

In order to achieve our goal through the aforementioned approach, our project team focused on two complementary objectives. The first was to generate ideas for possible means of autonomously collecting the information required by Homesite Insurance. To accomplish this objective we conducted an analysis of the information required by Homesite Insurance in order to issue an insurance quote. This analysis largely revolved around an electronic application form featured on Homesite Insurance's website. This form collected all of the data Homesite Insurance required from a client in order to provide him/her with an insurance policy. The form then allowed him/her to purchase the policy directly from the website. By identifying possible means of autonomously collecting the data this process required, our project team was then able to generate ideas for the utilization of these means in order to partially automate the process of obtaining a homeowner's insurance quote.

Our second objective was to implement a number of the ideas we generated in a mock prototype of Homesite Insurance's smartphone application. This prototype was later divided into multiple, single-purpose prototypes. By developing these prototypes, we were able to acquire a better understanding of the quote acquisition process as a whole and thus refine our ideas for its improvement. In addition, we were able to test out a number of our data-collection ideas in order to acquire a better understanding of their potential usefulness and ability to be implemented. We termed the applications we developed as a collective “mock” prototype for two reasons. First of all, our prototype was not meant to be as fully-functional as an application developed by Homesite Insurance would be; instead our prototype was meant to act as proof-of-concept for the ideas we generated. This factor is what allowed us to divide our prototype into multiple applications; the applications focused on demonstrating our ideas rather than putting them to actual use. Second, our application wasn’t meant to be a direct prototype of Homesite Insurance’s application. In other words, while Homesite Insurance would use the ideas we implemented in our prototype, they would develop their own application from the ground up. This was to ensure that the code implemented by Homesite Insurance met the standards of performance and security the company requires of its commercial software.

By accomplishing these objectives, our project team was able to make a number of recommendations to Homesite Insurance for the autonomous collection of the data necessary for providing homeowner’s insurance quotes to potential clients. These recommendations largely focused on identifying methods for collecting data about a client’s property without requiring the client to enter the information manually. These methods were chosen such that they could be utilized by a smartphone application, either through the operation of the application itself or through interactions with external or distributed processes. In addition, we created a set of smartphone applications that demonstrated a number of our ideas in order to better present their use and implementation. In this way, our project team was able to present Homesite Insurance with a number of possible means of collecting data independent of a potential client. Implementing these methods in a smartphone application will allow Homesite Insurance to provide possible customers with quotes on homeowner’s insurance policies while demanding less effort and information on the part of the prospective client.

Acknowledgements

First and foremost, we would like to thank the staff of Homesite Insurance for giving us the opportunity to conduct this project. We would like to extend special recognition to our liaison, Phil Mousseau, for his taking the time to attend our weekly meetings.

We would also like to thank our advisors, Professors Emmanuel Agu, Guillermo Salazar, and Matthew Ward, for their guidance and support over the duration of this project.

Table of Contents

Table of Figures

Chapter 1: Introduction

One of today's more ubiquitous markets is that of insurance, be it personal, real estate, business, or any other type of insurance. The focus of this project was the part of the business that consists of collecting information about a potential client, calculating an insurance premium based on the client's information, the type of insurance being purchased, and the company's own cost calculation methods, and allowing the client to purchase the policy. Our sponsor for this project was Homesite Insurance, an insurance company based in Boston, MA, which provides homeowners insurance primarily to homeowners, renters, and condominium owners (Homesite Group Inc., 2012). Homesite Insurance proposed this project so that we could solve a common problem in the business of providing homeowners insurance: the accurate, efficient collection of the information needed to issue an insurance quote. The root of this problem lies in the fact that this information is diverse, extensive, and may be outside the client's pool of readily-available knowledge.

In response to the complexity of providing homeowners insurance, a variety of business utilities have been developed to facilitate the process. Currently, many insurance companies provide custom-designed tools developed to simplify the policy purchasing process for all involved. For example, Homesite Insurance, like most modern companies, has a website that they use to promote and conduct their business (website URL: http://www.homesite.com/). One of the features of this website is an electronic form that a user can fill out in order to receive an insurance quote. The user can then purchase the policy directly from the website. This tool and others like it offer both a means of better conducting the business of insurance and a foundation for further developing these utilities. In addition to creating and employing their own utilities, insurance companies can also utilize the services of third-party and same-market businesses. One of these is the data vendor, a business or organization that collects data about properties and their current owners. These vendors typically sell this information to various buyers, often offering discounts to insurance companies that will be buying this information in bulk (Mousseau, 2012). As such, insurance companies can purchase information about a property or client from a reputable source, thus reducing their reliance on the information provided by the client being accurate. In addition, insurance companies often exchange information between each other, both to acquire information independent of the client and, in some cases, to provide the requesting company information about the other company's experience with that particular client.

In recent years, advances in computational technology have led to the rise of the modern smartphone, a handheld device that can access a vast collection of information resources and perform a plethora of computational feats. For a picture of the Samsung Galaxy S III, one of these modern smartphones, refer to Figure 1. This device emerged during a period where the usage of handheld devices, such as Personal Digital Assistants (PDAs), was significantly increasing among professional communities (Garritty, 2006). The steady increase in the use of handheld devices, including smartphones, has been due largely to the devices becoming more compact and computationally powerful as the technology advances, making them capable of performing more complex tasks while remaining convenient to use and carry (Albanesius, 2011). For example, smartphones now feature new tools (e.g., cameras, gyroscopes, and web access) and physical components (e.g., keyboards and USB ports) while retaining their relatively small size. As the number of individual functions that smartphones can perform increases, smartphone users can

Figure 1: The Samsung Galaxy S III (Android, 2013)

employ their devices in a greater range of activities, inspiring smartphone developers to further develop these capabilities and thus continue the cycle (Oulasvirta et al, 2012).

Relatively recently, two components of the modern smartphone have become prevalent in their increased usage: internet-based activities and self-contained smartphone applications (commonly referred to simply as “apps”). Because smartphones have become capable of accessing the internet, effectively any material uploaded to the internet can be accessed by a smartphone. This has led to the redesign of many websites and other internet services such that their format is automatically adjusted to better complement the screen size and user interface of a smartphone when accessed by one. (For a screenshot of MapQuest’s mobile interface provided by Google Play’s app store, refer to Figure 2.) In addition, smartphone users can use the internet to conveniently download and install apps. This allows a smartphone user to customize his/her smartphone with hand-picked applications, which can often utilize the host smartphone’s resources, creating a market for smartphone apps developed either by established companies or by casual developers. Homesite Insurance, in response to the popularity and potential of smartphones and smartphone applications, has decided to augment their online form with a smartphone app designed to take advantage of the various resources available to smartphone applications. In order to collect ideas and strategies that can be used in the development of their application, they proposed the following project to us.

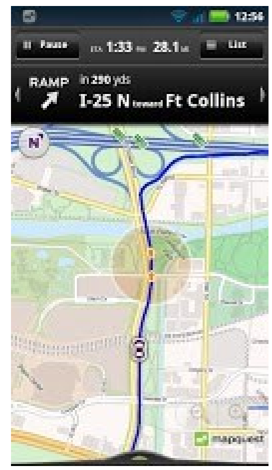


Figure 2: Screenshot of MapQuest's mobile interface

1.1: Project Vision

Modern software, in conjunction with current information resources, is capable of simplifying many processes that would otherwise be long and possibly error-prone. It was the intent of this project to explore how the use of smartphone technology can improve the process of purchasing homeowner's insurance, using Homesite Insurance as the project's focus. In particular, this project was intended to explore how the resources available to modern smartphones could improve upon the process established by Homesite Insurance's current online form. The goal of this application of modern smartphone resources was to streamline the process of purchasing insurance from Homesite Insurance while reducing the amount of effort demanded of the user.

Our efforts towards developing these strategies were largely focused on the fact that much of the information Homesite Insurance uses to calculate the costs of its policies can be found on the Internet; by designing a smartphone app to collect data from these sources, the amount of data that must be entered by the user can be greatly reduced. In addition, by collecting information about a home from sources that have proven to be reliable, these data, and thus the quote itself, are more likely to be accurate. In this way, developing a smartphone application that implements these strategies can improve customer experience, improve the accuracy of policy quotes, and add a useful tool to Homesite Insurance's customer-relations department.

1.2: Project Goals

The purpose of this project was to develop ideas for a smartphone application to be developed by Homesite Insurance, with the practical objective of creating a proof-of-concept prototype application. As such, our efforts focused on generating, critiquing, testing, and demonstrating ideas that a fully-developed app can use rather than actually developing a version of the final application. In particular, the application prototype we developed demonstrates how utilizing advances in technology can reduce the process' dependency on human input and improve user experience overall. This overall objective was comprised of three individual goals:

- Provide an analysis of the information required to obtain an insurance quote from Homesite Insurance. This analysis provided ideas for what technology can be used to assist the applicant in filling out the different fields of the application form.

- Design a smartphone app that implements at least three of our theories for designing a quote generation app. Our main focuses for this project were GPS tagging, calculating the accuracy of a generated policy estimate, and web crawling.

- Provide further ideas for how the form can be automatically filled in, including ideas that we did not implement in our own app as well as ideas that might be realized with future work or advances in technology.

Chapter 2: Background

The field of smartphone development is one that continues to grow and evolve rapidly. As the smartphones themselves have evolved, so have the applications designed for them. In this chapter, we will examine the different technologies behind these apps as well as their various applications. We will then explore some of the more modern resources that smartphone apps can utilize, as well as how they can be applied to our project. In our discussions we will focus on the more sophisticated resources, such as global position systems, rather than the simpler ones, such as built-in cameras. We will then turn our attention to Homesite Insurance's current online tools in order to define the groundwork of our project's efforts.

2.1: Related Technology

In the past two decades, advances in computational technology have allowed cellphones to evolve into the modern smartphone (Chowdhury, 2013). The notable elements of this evolution started with the cellphones' screens as touchscreens were developed and many phones began to feature multiple screens, the latter models being complemented by different design styles such as clam shells (commonly referred to as flip phones). Many phone models then received the integration of a built-in camera, made possible by advances in cellphone hardware. In 2007, Apple released the iPhone, the first cellphone to have an advanced touchscreen and operating system, making it the first smartphone. (For a complete picture of the first smartphone, provided by BusinessInsider.com, see Figure 3.) This phone, capable of accessing the Internet and running apps, set the standard for modern smartphones to follow.

Figure 3: The original iPhone

2.1.1: Existing Smartphone Applications

The development of the modern smartphone led to the rise of today's smartphone application ("app" for short). Many of the apps used today are designed to provide useful functionality for the user (Sung et al, 2012). Of these apps, some are designed to manage the files and resources possessed by the smartphone itself, while others simply provide a fast, easy way to access internet services. Some examples of the former include apps that track data use, either in the context of data plans or simply how much data each of the smartphone's different apps use, while examples of the latter include quick-reference weather forecasts and email browsers. Still other apps follow the tradition of the Personal Digital Assistant (PDA) in providing useful services in their own right. These apps include media players, financial trackers, organizers, and many other service models. Finally, a number of apps allow users to access different resources available to modern smartphones, including automatic call answering, location determination, and phone orientation. For some screenshots of smartphone apps currently in use, provided by Google Play's app store, see Figure 4 through Figure 7.

Figure 4: My Data Manager screenshot

Figure 5: Accu-Weather screenshot

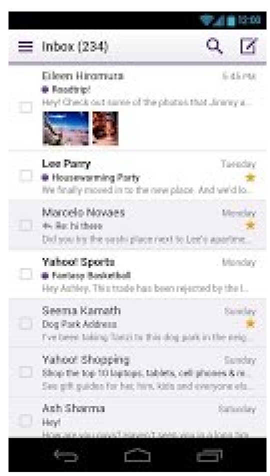


Figure 6: Yahoo! Mail screenshot

Figure 7: Jorte Calendar screenshot

As these various apps gained acceptance and popularity, the number of both commercial and casual developers expanded. Today, many companies provide smartphone apps that complement their business models, assisting either in customer relations or commercial services. For example, we found numerous instances of apps that could be downloaded and used in lieu of using a website or interacting with a representative, thus offering clients an alternative means of conducting business with the organization. In addition, we found a number of companies that provide apps that allow a user to access his/her account with the company, such as banking and online shopping organizations.

<u>2.1.2: Global Positioning Systems</u>

A common feature of modern smartphones, though absent from Apple's first iPhone, is a Global Positioning System unit, known more commonly by the acronym GPS. Current smartphone GPS units typically determine location in one of two ways: through a satellite connection or through positioning software provided by the smartphone's service provider. Of the two, satellite-based systems are more accurate, but more resource-intensive. Compared to dedicated positioning technologies, a standard GPS app can be almost as accurate (within a few feet) as its "official" counterpart (Nelson, 2012). In this way, a decent GPS component can provide a smartphone with highly useful location data, especially when the provided location doesn't have to be overly accurate. This data is routinely utilized by a plethora of apps, such as trail navigators and tourism guides (Hostetter, 2013). (For a screenshot from an app that provides thorough location information using the host's GPS unit, see Figure 8.) The data can also be used to determine one's current address, a function that we employed in our project.

Figure 8: GPS Status & Toolbox screenshot

2.1.3: Auto-Population

One of the more popular time-saving strategies when using electronic forms is the practice of auto-population (O'Bannon, 2008). This involves presenting the user with the form with some or all of the information already present as if the user had manually typed it in. This practice is most popular with forms that summarize/collect data already present on other forms, such as tax documents. However, the practice has also been taken up with online commerce sites that focus on customer retention. For instance, when a client sets up an account with Amazon's online store (website URL: http://www.amazon.com) and carries out a purchase transaction for the first time, Amazon's database stores the billing and shipping information provided by the user. The next time the client goes through the purchasing process this information is retrieved and presented to the user. This allows the user to confirm that the stored information is still relevant and can be applied to the new transaction, saving the user the trouble of entering the same information a second time. In short, Amazon collects and retrieves information about the user so that, on average, the user doesn't have to enter as much information as he/she would if the data weren't collected.

While collection and storage of customer data is a common practice amongst modern companies, this project focused on using the retrieval aspect of auto-population. Specifically, our efforts concentrated on utilizing services that can autonomously collect information about the transaction in question, thus saving the user the trouble of entering the information manually. To illustrate, if an electronic form contains fields that pertain to the an address, such as city, state, and zip code (for example, an online purchase form for a product that will ultimately be shipped to one's home address), the host service can use location information provided by the user's service provider to auto-populate the address fields with the user's current location. It is this idea of proactive auto-population that we explored over the course of this project.

2.1.4: Data Vendors and Web Crawlers

In working with Homesite Insurance, we learned that there are many data vendors on the Internet that provide information on pieces of real estate, including the types of houses that Homesite Insurance insures (Mousseau, 2012). In addition, several of these vendors provide a subset of the property information that a homeowner needs to provide in order to purchase a policy from Homesite Insurance. Considering this resource, our group decided that one of the best ways to gather reliable information about a property without requiring direct user input would be to query these data vendors for information about the property being insured. A strong advantage to this approach is that it only needs the house's address in order to query the data vendors' records and thus collect more information. In this way, the data vendors can be used to provide information about a property such that the amount of information that the client needs to enter manually is minimized.

In order to retrieve the aforementioned data from online data vendors, we decided to develop a web crawler, a program designed to autonomously browse the Internet in order to collect information. Because our prototype is a proof-of-concept, we decided to restrict the data vendors we investigated to ones that allow public access. These are in contrast to the business-oriented data vendors that Homesite Insurance uses, which require a purchased subscription in order to query records. In addition, we decided to restrict the scope of our project, and thus the records we had to consider, to the City of Worcester. One of the reasons for choosing this city was because of our discovering the Worcester Public Records, hosted by the City of Worcester (Worcester, 2013). In addition to the Worcester Public Records, we decided to investigate two more pervasive data vendors, namely Zillow.com (Zillow, 2013) and Trulia.com (Trulia, 2013). In this way, we were able to focus on a set of specific data vendors as we developed our ideas for and prototypes of web crawlers.

2.2: Homesite Insurance's Online Form

One of the tools Homesite Insurance has developed in order to facilitate the purchasing of an insurance policy is an electronic insurance application form that can be accessed from the company's website. Currently the main benefit of this service is that the process can be conducted without requiring an in-person inspection of the property being insured, a typical requirement with most property insurance transactions. The application form itself is composed of a list of questions that collect the information the website needs in order to offer an insurance quote, which the applicant can then use to purchase a policy directly from the website. The questions on the form are organized by category into six distinct sections, each of which must be completely filled out for the customer to receive a quote, while the form's access page requests the user's zip code, at which point the first section of the form is made available. For a set of screen captures of the online form, taken in October of 2012, see Appendix A: Current Online Form Screen Capture. For a section-by-section review of the form, see Appendix B: Sections of the Current Online Form. For a complete list of the form's fields, see Appendix C: Fields of the Current Online Form.

2.2.1: Preliminary Interviews

Before creating a prototype that featured a mobile version of Homesite Insurance's online form, we conducted a number of user interviews about the current online form. The goal of these interviews was to obtain a better understanding of how potential clients react to the current electronic form. This understanding facilitated our efforts to create a prototype of a user-friendly smartphone application. During the interviews themselves we had people answer a preliminary questionnaire and then fill out the form as if they were applying for homeowner's insurance, simulating the scenario of a client using Homesite Insurance's online application. We asked each of them to provide us with running feedback on the form as they completed it and informed them that they were free to stop filling out the form at any time; they were not required to finish it. The demographics of our interviewees and a sample of the feedback we received can be found in Appendix D: Sample Preliminary Interview Feedback.

Looking over the results of the interviews, a number of correlations became evident. During the preliminary interviews, it was clear that our interviewees felt that the main attraction of a smartphone app is the tradition of smartphone apps being simple to use; if the user felt that the app was too difficult to use then he/she would likely give up partway through. Conversely, if it provided strong convenience features, such as functional help features, auto-population, or the choice of requesting a quick "ballpark" quote over an exact one, then the app would be deemed highly useful and users would be inclined to see the process through to completion. In the absence of these features, however, the user would be more likely to make do with the electronic form currently featured on Homesite Insurance's website.

Considering the feedback on the current online form, however, a number of legitimate criticisms are apparent. First and foremost, users were affronted at the start by the form's request for their social security number. Through discussions with our sponsor, our team learned that this field does not provide any information that cannot be garnered through other means. As such, the disconcertion it causes is wholly unnecessary. Another problem that was identified is the extensive nature of the form; both we and our interview subjects became frustrated in how long it took to complete the form. The common consensus was that it would be too great a hassle to complete the form as it was on a smartphone, especially considering the compact properties of a smartphone's input methods, such as touchscreens and miniature keyboards. Finally, a number of the fields on the form impressed both us and our participants as ones that a casual homeowner would have considerable trouble answering off the top of his/her head. As such, we decided that the best thing that a smartphone application could do for these fields would be to either fill in the information automatically or otherwise assist the user in acquiring the information.

2.2.2: Criteria for a quote

Through discussions with our sponsor from Homesite Insurance, we learned that the final price of an insurance quote depends largely upon three different factors: the house's replacement cost, the customer's insurance risk score, and the property's occupancy information (Mousseau, 2012). These categories each have an attributed weight in the final price. At the suggestion of our sponsor, himself a senior business analyst at Homesite Insurance, we decided to design our prototype to calculate the replacement cost in lieu of the final policy cost. In this way, we could demonstrate the methodology of our proof-of-concept without having to worry about acquiring extensive information on the property being insured or performing overly-complicated calculations.

Looking into the calculation of the replacement cost of a house, we learned that Homesite Insurance derives the replacement cost from several criteria. The four primary criteria are:

The year the home was built
The home's square footage
The shape of the roof
The shape of the home's foundation

In addition to the primary criteria, there are several secondary criteria. These include:

The number of stories
The number of bathrooms

The number of rooms with cathedral or vaulted ceilings
The number of rooms with crown moldings
Kitchen countertop material
The floor type
The inside wall material
The exterior siding material
The roofing material
The foundation type
The garage type

To simplify the operations of our proof-of-concept, we focused on gathering this information and then using it to perform a mock calculation of a replacement cost. This process of gathering data and then performing a cost calculation was deemed a suitable simplification of the larger process of using Homesite Insurance's electronic application form.

Chapter 3: Design Process

Our practical objective for this project was to generate ideas that Homesite Insurance can implement in a smartphone app designed to improve upon the functionality of their current online application form. In order to properly cultivate these ideas, we decided to develop a smartphone app prototype to complement our project report. The primary functions of this prototype were to provide a means of demonstrating the application of our ideas and to provide a testing platform for these ideas. In this chapter we will discuss the process of developing our prototype, including the smartphones it was designed for and tested on and the development environments/tools we used. We will also discuss some of the problems we identified during the development process as well as some of the solutions the process brought to light.

3.1: Objectives

The prototype developed over the course of the project was meant to be a forerunner of the smartphone app Homesite Insurance proposes to develop based on the results of our project. This precursor prototype served three practical purposes. The first was to act as a proof-of-concept for the ideas we generated during our project; the prototype demonstrates how these ideas/methods can be used to meet the requirements/goals of the app that Homesite Insurance will subsequently develop. The second was to give us a platform for testing our ideas and theories; by putting our ideas through a test run while our project was in progress we were able to further develop these ideas and determine whether or not they were viable suggestions. The third purpose of the prototype was to provide the developers at Homesite Insurance with a starting point for developing their own app; they can use our prototype in conjunction with our report to better understand the resources they will require and to generate ideas and design strategies that they may not have come up with on their own.

It is important to note that our prototype is meant to be utilized from a conceptual standpoint rather than a code-based one; Homesite Insurance will need to create their version of the app from the ground up. They will need to do this in order to ensure that their app complies with any and all standards that the company requires of its products. Because our prototype's code was written for development purposes, these standards were not a part of our development criteria.

3.2: Requirements

As a proof-of-concept, our prototype had only a few solid requirements. These criteria fell into two different categories: procedure-based requirements and information-collection requirements. By satisfying these criteria, our prototype provided a logical structure for purchasing an insurance policy and demonstrated how the final app can gather and use data.

The requirements relating to procedure were relatively simple: the app needed to simulate a logical progression through the process of purchasing an insurance policy using a smartphone app. In other words, our prototype needed to demonstrate the functionality of each component of the app and how these components interacted with each other. This allowed our prototype to properly demonstrate how our ideas can be implemented in the final application. The tangible components that needed to be simulated included welcome screens, instructions, data-entry screens, and result screens.

The requirements pertaining to information collection and application were threefold. First, the prototype needed to accurately denote the information that the final app will be designed to collect, though it didn't have to cover all of the information that the final app will need to collect. Second, the app needed to demonstrate how the prototype's data-gathering procedure could be handled and influenced by the different components of the app. This requirement was particularly important given that the central purpose of our prototype was to identify means of collecting data autonomously. Finally, the collected information needed to be used in the calculation of a tangible result. For the purposes of our project the prototype was designed to estimate the replacement cost of the home in question in lieu of the cost of an actual policy, as this calculation was simpler to perform and required less data.

3.2.1: Data Required by Our Prototype

During the development of our mock prototype, we based our information gathering process on the data Homesite Insurance gathers using its online application form. However, our calculation process, designed to provide a replacement cost rather than the cost of an actual policy, did not require all of the information necessary for a complete quote. As such, we went through the fields of Homesite Insurance's online form, assigning the data elements to four non-exclusive categories. These categories are as follows:

- **Necessary for calculating an accurate replacement cost**: these data can be plugged into a formula for the construction cost of residential buildings provided by RSMeans, a company that provides construction cost data, in order to calculate a replacement cost estimate (RSMeans, 2011). This category includes square footage, number of stories, wall material, basement type, roofing material, air conditioning type, heating method, countertop material, garage type, presence of a fireplace, and the number of additional full/half bathrooms. If one or more of these data are not provided, the accuracy of the calculated estimate decreases.

- **Essential for calculating a replacement cost estimate**: these data must be provided in order for an estimate to be provided; without these data elements an estimate cannot be calculated. Currently, this category only has square footage and house style. It's worth noting that, while an address is not used to calculate a replacement cost, an address is required in order to autonomously collect data. As such, an address is required by the overall process, though not by the actual calculation being performed.

- **Can be automatically populated**: this category includes all data that can be collected without direct user input. The data in this category are those that can be collected by direct data processing (for example, getting a location through the smartphone's GPS unit) or by acquiring property reports from established data vendors. Natural exclusions from this category include the customer's name, date of birth, email address, and policy start date. For a sample property report provided by the Worcester Public Records, see Appendix E: Sample Property Report.

- **Unnecessary for calculating a replacement cost**: these data consist of the data collected by Homesite Insurance's online form that are not used to calculate a home's replacement cost. As such, though means of collecting these data were considered, implementing such means was outside the scope of this project.

3.3: Constraints

The constraints of our prototype were largely determined by the goals of our project and by the duration of the project. To elaborate, our project's central vision was the generation of ideas for the autonomous collection of information on a property, while the project's scope of practical development was the creation of a mock prototype. These two factors dictated a number of design choices made during the prototype's development. The two foremost constraints were as follows:

- **Minimize the amount of info that the user has to enter manually**: the less data that the user is responsible for directly providing, the better. From a practical standpoint, the goal here was to acquire as much information about a property as possible independent of the user. Having the user confirm the data that was collected was acceptable. The central drive behind this constraint was the fact that a user may not have all necessary information readily available.
- **Restrain the set of data sought by the prototype**: the smartphone application to be designed by Homesite Insurance will need to collect a significant amount of data. The prototype we developed, however, was designed to calculate a replacement cost in lieu of a policy quote. As such, our prototype needed to collect significantly less data. Reducing the amount of data that the prototype sought to collect allowed us to focus on the functionality of the prototype itself.

3.4: Solutions

The primary goal of our project was to generate ideas and methods that Homesite Insurance can use to accomplish the purpose of their prospective smartphone app: to provide clients with policy quotes through a process that is effective, easy to use, and provides a good overall customer experience. Our principle focus with regard to this goal was to identify ways of using the different resources available to modern smartphones in order to autonomously collect the property information necessary to provide an insurance quote, thus minimizing the number of fields left to the user. To elaborate, our team sought to identify means of deriving the necessary information from the immediate environment, from different data systems/services, and from internet-based information providers. The different means of derivation we identified and the information that can potentially be derived from them is discussed in the subsection below.

3.4.1: Potential Information Resources

In order to collect the information necessary for our prototype's calculations, we looked for ways to autonomously collect data using information and resources that we expected to be readily available to the final application. Because the final application will be run on a smartphone, these resources and the information they will provide largely revolve around the resources commonly available to modern smartphones. A list of the resources we explored follows.

- **Camera**: this feature allows the user to acquire a visual representation of the house in question. This image can hypothetically be processed to derive information about the house such as number of stories and approximate square footage. The image itself can also be used as an icon for a quote.
- **Global Positioning System (GPS)**: this component can use either satellite positioning or location data from service networks to provide a coordinate-based location for the device in question. Often it is used to place a GPS stamp on a picture taken using the device's camera; thus taking a picture can in effect provide a location instance.
- **Compass**: a GPS subcomponent that can be used to determine the direction a user is facing. When a picture is taken, this can be used to complement a GPS location instance in order to more accurately determine the location of the picture's subject.
- **Keyboard**: a standard interaction tool that can be either physically built into the smartphone or replicated virtually on a touchscreen. This tool allows users to enter information as simple text as with a true keyboard. Though simple to use and integrate, overuse of this tool is likely to be highly arduous to the user, as smartphone keyboards

are generally compact in size. As such, reliance on this tool should be minimized as much as possible.

- **Touchscreen**: an interaction tool that a user can use to enter information. This tool is distinct from the keyboard in that it can be used with image-based selection instead of text entry, thus allowing the app to present choices more intuitively and the user to make such choices with less effort and chance of error.

- **Internet access**: the primary user-independent information resource for our app. Once a coordinate location has been acquired, any one of a number of internet services can be used to derive a corresponding street address, which in turn can be used to query online data providers for property records that can provide further information (see Appendix E: Sample Property Report for a sample property report that was acquired from one of these data providers).

Chapter 4: Implementation

The prototype component of our project serves the purposes outlined in the previous chapter: the simulation of the process of purchasing an insurance policy from Homesite Insurance using a smartphone application. To elaborate, it walks the user through the steps of collecting information, both autonomously and manually, and generates an insurance quote. That being said, the prototype has a few simplifying factors that differentiate it from a true version of the final application. For instance, the app isn't designed to give an actual insurance quote; instead it calculates and returns an estimate of the house's replacement cost. In addition, it only seeks to acquire the data necessary for this calculation, rather than the larger set of data collected by Homesite Insurance's current online form. As such, our prototype functioned as a means of developing, testing, and demonstrating the ideas generated through the completion of this project.

4.1 Platforms and Development Tools

Our prototype was developed to run on the Android operating system, a Linux-based OS designed for mobile devices. We chose this operating system because of its immense popularity, widespread use, and abundant capabilities (Android, 2013). As such, this operating system offered us the raw computational capabilities our prototype required and the support we needed to create a functioning application within the duration of our project. Having chosen Android as our target operating system, it made sense to use Android smartphones as our testing platforms, though we also tested our prototype's ability to run on Samsung phones. Android applications such as the one we developed are meant to work on all versions of Android, but are compiled with a specific "target" version. Over the course of our project we came to focus on Froyo 2.2 and Jelly Bean 4.1.2 as our target OS versions, these being the versions that our test phones ran.

During the development of our prototype we investigated and utilized a number of development environments and tools. The utilities we found and critiqued are as follows:

- **Eclipse (Eclipse, 2013)**: an open-source programming environment that we eventually settled on as our main development tool. This environment is designed to facilitate the writing and organizing of programs in a variety of languages and allows the user to customize the environment using different plug-ins, including an Android plug-in that allowed our team to run our prototype on virtual Android devices as we developed it.
- **App Inventor (MIT, 2012)**: an online development environment that allows developers to create applications through a simplified drag-and-drop interface. This environment allows users to develop program and device behavior through the manipulation of visual objects rather than through the writing of executable code. Although this would have greatly simplified the process of building our application, allowing us to access smartphone resources without complex syntax, it did not allow us to implement the more advanced functionality that our project required. As such, we chose not to use this utility in the development of our prototype.
- **PhoneGap (Adobe Systems Inc, 2013)**: an open-source development tool that allows users to write smartphone applications in the language of their choice and then convert them into deployable applications. Although this offered us the ability to write different components of the app in languages suitable to that particular component, we found that it was easier to write the entire application as a single entity using Eclipse.

4.2: App Layout

Our prototype was constructed based on our expectations regarding Homesite Insurance's implementation of the application in terms of screen flow, component location/invocation, and resource usage. To elaborate, our prototype's layout and flow simulate both aesthetic screens, such as a splash screen and a welcome screen, and utility screens, such as data collection screens and a results screen. Our prototype also uses the host smartphone's assorted resources, such as its camera, GPS unit, and internet connection, in the same ways that we expect the application developed by Homesite Insurance to use them. Our prototype also includes data-processing components similar to those that the final product will require, such as a cost calculator and a web crawler. These components are detailed in the following subsections. For a flowchart of our prototype composed of screenshots of the different components, see Appendix F: Screenshot

Flowchart. For a modular flowchart of our prototype that includes underlying components, see Appendix G: Modular Diagram.

4.2.1: Screen Components

Our prototype was designed to simulate the tangible components that the final product would likely possess, so long as they were within the scope of our project. As such, our prototype simulates every screen component necessary for a logical progression through the use of the app. The different screens that our prototype implements are described below. For a screenshot-based flowchart of the different screens, see Appendix F: Screenshot Flowchart.

- **Splash Screen**: traditionally the first screen presented to the user, this screen is displayed while the application is loaded (a minimum duration is typically enforced). Once the application has finished loading, the user is automatically sent to the next screen.

- **Welcome Screen**: this screen is presented immediately after the splash screen and is the first interactive screen. From this screen the user can choose to start a quote (either with or without taking a picture), to view an old quote from a prior session, or to exit the app.

- **Picture-Taking Screen**: the purpose of this screen is for the user to take a picture. This provides both a relevant GPS stamp and an image by which the current quote can later be identified. At this point the address is calculated using the GPS stamp and presented to the user for confirmation/correction. Once the address is confirmed, the app progresses to the Data-Collection screen.

- **Data-Collection Screen**: this screen dynamically presents the user with questions to answer. If the user did not supply an address via a picture, it requests the address at this point. Once the user has provided the address, the app seeks to auto-populate information about the address if the user has agreed to auto-population. While this is taking place, the app asks the user questions about the property that cannot be auto-populated. Once these questions are completed, the app asks the user to confirm any and all fields that were auto-populated. The app then proceeds to ask the user to supply any information that failed to be collected autonomously. At any time the user can proceed to the View Quote screen for a quote that sacrifices accuracy for expediency. Once all questions have been answered, the user is automatically directed to the View Quote screen.

- **View-Quote Screen**: this screen shows the estimated replacement cost calculated using the information provided by the user. If any fields were omitted, the expected inaccuracy will be displayed. This screen also allows the user to act on the "quote" provided by the app in one of two ways: they can call a Homesite Insurance agent directly or email themselves the data collected via the app. The user is also given the option of returning to the main screen.

- **Quote Selection Screen**: this screen displays all quotes created by the user in prior sessions. When a quote is chosen, the data collected during that session is retrieved and the session resumes at the appropriate point in the process.

4.2.2: Smartphone Resources Used

Our prototype takes advantage of a number of different resources that modern smartphones provide to resident apps. Our prototype uses these resources to gather initial data, to autonomously collect additional data based on the data already acquired, and to allow the user to act on the results calculated by the app. These different resources and their uses are described below.

- **Camera**: perhaps the most distinctive feature of our app is the picture-taking component. This component provides the app with an image to associate with the quote being created and gives the app the opportunity to acquire a GPS stamp. The use of the camera is an important step, as it ensures that the user will be on-location at the time and thus will be providing a relevant GPS location.

- **GPS**: the phone's Global Positioning System unit is what really provides a GPS stamp; the act of taking a picture simply provides an opportunity to acquire a relevant one. That being said, the GPS can be used independently of the camera if the user cannot or does not wish to take a picture, e.g. if it is nighttime or if the weather is bad. The location stamp provided by the GPS unit is central to autonomously procuring an address.

- **Compass**: this tool can be used in conjunction with the GPS stamp to more reliably calculate an address, as it allows the application to determine "where the camera is pointing" instead of only "where the camera is currently located."

- **Internet Connection**: this utility is critical to our app's principle functionality: the automatic collection of data about a property. Because the data is stored online, the app needs to access the Internet in order to retrieve it. Another point to consider is that our prototype contains the address calculation, data collection, and cost calculation utilities within the app itself. However, these utilities can be relocated onto a server or other distributed resource when the actual app is developed, in which case the internet connection will become necessary to communicate with these distributed resources.

- **Voice-Call**: this utility, a standard feature of any current smartphone, can be used by the app to call a Homesite Insurance agent at a preset phone number.

- **Email Service**: this service can be used to send the data collected by the app to the user, either to the email address provided on the form or to a different address. While our prototype formats this email as simple plaintext, the final app may need to handle this feature differently.

4.2.3: Data-Processing Components

Our prototype utilizes mock implementations of three distinct data-processing components: an address calculator, a cost/accuracy calculator, and a web crawler. These components are described in detail below.

- **Address Calculator**: this component takes the GPS stamp provided by the smartphone's GPS unit and the direction reading provided by the smartphone's compass utility and queries Google Maps for a street address (Android, 2013). Our testing indicates that this feature is highly accurate, though it will be necessary to confirm the address with the user before it is used in any subsequent data collection. For further information about the coding of this utility, refer to Appendix H: Discussion on Coding the Address Locator.

- **Cost/Accuracy Calculator**: this component takes all data provided so far and performs a rough replacement-cost calculation using a general-purpose formula for calculating the construction cost of a residential building (RSMeans, 2011). This component also takes on the task of providing a rough accuracy calculation by providing high and low estimates based on what information has and has not been provided. This component is a stand-in for the formula(s) Homesite Insurance uses in their quote calculations.

- **Web Scraper**: this component takes an address and queries the Worcester Public Records (Worcester, 2013) for information about the property at that address. It then takes the returned data and parses it so that it can be used to automatically fill in the fields of the "form." This scraper was developed and implemented as a demonstrative stand-in for the data collector(s) that Homesite Insurance will develop for their own app. For further discussion about the idea behind web crawlers, refer to Appendix I: Discussion on Coding our Web Crawler.

4.3: Cost Calculation, Accuracy and Progress Meters

Our prototype features a calculation component designed to simulate the formulas Homesite Insurance uses to calculate the costs of its insurance policies. Because the formulas used by Homesite Insurance are outside the scope of our project, our prototype component instead estimates the replacement cost of the home in question. This estimate is acquired using a general-purpose formula for calculating a residential building's construction cost (RSMeans, 2011). In addition to providing this estimate, our implementation of the calculation component supports progress and accuracy estimates. It supports the former by keeping track of which fields have been answered and supports the latter by providing “high” and “low” estimates based on the questions that have not yet been answered. These features are described at length in the following subsections.

4.3.1: Cost Calculator

The cost calculator we implemented in our prototype was designed to serve as a basic test of functionality rather than to calculate actual policy quotes. To this end, our implementation calculates the replacement cost of a home rather than the cost of an insurance policy for that home. That being said, it was agreed that estimating this value was a reasonable stand-in calculation, as the replacement cost of a house is a major component of an insurance policy’s overall cost (Mousseau, 2012).

Simplifying the equations used by the calculation component allowed us to focus on the component’s functionality. For instance, the component had to be structured so that it used different formulas according to the style of the house in question, as different house styles (e.g. economy, average, custom, and luxury) incur different costs for the same material choices (RSMeans, 2011). Our implementation also provides functionality for providing an estimate when data is absent; in such a case, it provides “best-guess,” “high,” and “low” estimates by substituting in generic, worst-case, and best-case values respectively for fields that were not filled in by the user. This allows the user to acquire a “ballpark” estimate at any time and allows the app to continuously calculate how accurate that estimate is likely to be. For a screenshot of the app we designed to test our calculation class that

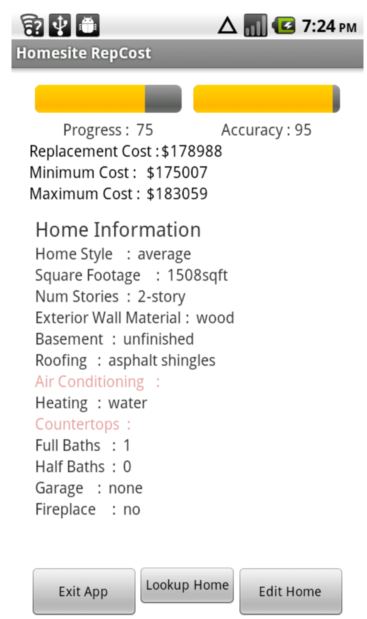


Figure 9: Screenshot of our quote calculation test app

shows the current, maximum, and minimum cost estimates, refer to Figure 9 on the previous page. For a description of how our replacement cost formula works, refer to Appendix J: Sample Replacement Cost Calculation.

4.3.2: Progress and Accuracy Meters

Our prototype features two bar meters that are updated each time the user provides a piece of information pertaining to the replacement cost calculation (refer to Figure 9). The simpler of the two is a progress meter that reflects how many questions related to the cost estimate have been answered. As described earlier, our prototype first asks questions that cannot be auto-populated in order to give our web crawler time to collect data from external sources. Once these questions have been asked and the data gathered, the calculation component can then calculate a starting value for the progress meter based on how many questions were auto-populated. As the user proceeds to fill in unanswered questions, the progress meter is updated to reflect the new degree of progress. Once all questions have been answered, the progress meter will display 100%. The formula for our progress meter is simply:

Decimal percentage = # of questions answered / total # questions

The other bar meter our prototype features is an accuracy meter. This meter is used to indicate the expected accuracy of the calculation component's cost estimate if the user were to ask for one at that point. The value displayed by the accuracy meter is determined by querying the calculation component for "best-

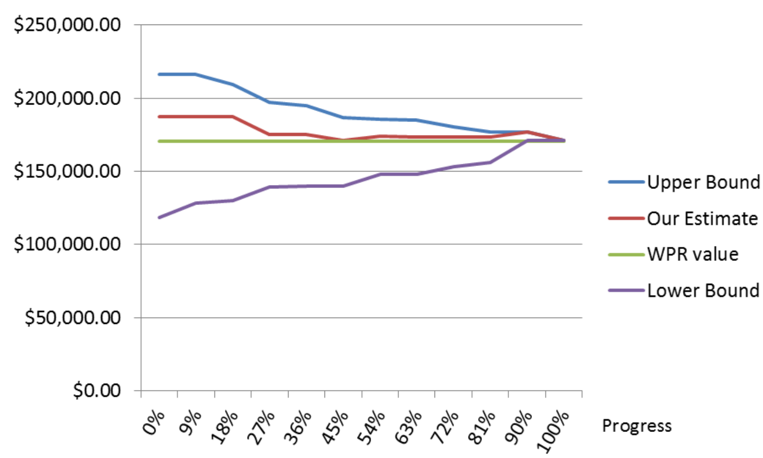


Figure 10: Progression of Cost Estimates

guess," "high," and "low" estimates, the latter of which are acquired by substituting "high-cost" and "low-cost" values into the cost calculations where values have yet to be provided by the user or by auto-population. This allows the app to estimate a range of values in which the final value is likely to lie and thus display the accuracy of the current estimated value as a function of the width of the range. Refer to Figure 10 on the previous page for a graph that shows how these upper and lower bound estimates converge as the user answers questions for one of the houses we tested. Once all questions have been answered the breadth of the range will be zero and the accuracy meter will display 100%. To keep our calculations simple we decided to use the following formula for our accuracy meter:

Decimal percentage = max{0, 1 – ((high estimate – low estimate) / current estimate)}

Refer to Figure 9 (two pages previous) for a screenshot of the app we designed to test our calculation class that shows how the progress and accuracy meters reflect the current state of the calculation. The calculation of these values is also covered at length in Appendix J: Sample Replacement Cost Calculation.

4.4: Data Collection

The main incentive behind the proposal of our project was the possibility of collecting data independent of the user. Realizing this possibility will allow the process of purchasing homeowner's insurance to become less arduous for the user and less prone to mistakes. The former is significant because a principle complaint during our preliminary interviews was that Homesite Insurance's online form was too long, indicating that a means of streamlining the process was necessary. The latter is simply an observation that if the user is asked fewer questions then there is less chance that the user will give up partway through or will answer incorrectly due to misinformation, frustration, or some form of misunderstanding. The main tools we focused on developing in order to autonomously gather data were GPS resources and web crawlers, each of which are described in the following subsections.

4.4.1: Smartphone Cameras and GPS Systems

The first step of our prototype's quote acquisition procedure is to determine the address of the home the user wishes to insure. The method we developed for determining this address is to acquire a GPS stamp, which provides a geographic coordinate location that is then used to query Google's Geocoder service for a street address. (For a screenshot of our address-lookup test app that illustrates this process, refer to Figure 11.) Our main prototype acquires the initial GPS stamp from the host smartphone's GPS unit. In our application, the acquisition usually takes place when the user takes a picture of the house being insured; because the picture needs to be taken on-site, the GPS stamp will most likely be relevant to the house's address. Should the user defer taking a picture, the prototype instead obtains a GPS stamp when asking the user to manually enter the house's address, using the returned address to tentatively populate the field's value. Because this latter method may be executed at a location other than at the house being insured and because the address returned by the Geocoder service is not guaranteed to be accurate, the user must always be asked to confirm the address. Once he/she has confirmed or corrected the address as appropriate, it can then be used by the other data collection component(s) to gather additional information.

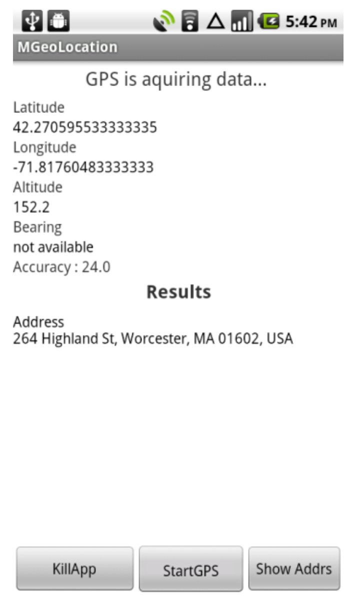


Figure 11: Screenshot of our address-lookup test app

4.4.2: Web Crawlers

Much of the property information used to calculate insurance quotes can be found online, either in public records or through data vendors. This includes much of the data required to perform our replacement cost estimates. As such, the retrieval of this information is the main focus of our prototype's automated data collection components. In order to simulate the collection of property information from online resources, we needed to develop a web scraper designed to query a chosen resource. To this end, our prototype features a web scraper designed to acquire data from the Worcester Public Records, a publicly-accessible database maintained by the City of Worcester (Worcester, 2013). For an analysis of the data provided by the Worcester Public Records and the other databases we considered in the context of our replacement cost calculation, refer to Appendix K: Information Provided by Online Databases. Because of how the Worcester Public Records stores and retrieves its data, we needed to manually execute the address-lookup procedure, record the page requests the web browser made, and code these directly into our scraper so that it could replicate the queries and get the data from the database. For further discussion on our web scraper, refer to Appendix I: Discussion on Coding our Web Crawler. Once supplied with the raw return data, our prototype parses the data into a format that it can process and uses it to auto-populate as many data fields as possible. Our prototype can then proceed to the manual data collection component, where any data that wasn't auto-populated can be supplied by the user.

Chapter 5: Results

Once we had completed our efforts to identify means of improving the process of purchasing homeowner's insurance using the resources available to modern smartphone applications, we performed a follow-up analysis of Homesite Insurance's current online form. During this analysis, we identified the fields of the form that we had found methods of answering automatically, the fields that we had found alternative means of answering from those currently implemented, and the fields that fit into neither of these categories. For the results of our analysis and the means and methods just described, see Appendix L: Means and Methods of Population. During our analysis, we found that much of the information we had been able to provide autonomously pertained to a house's construction details and similar property values. Because this information is least likely to be known to the customer, this was the information most important to acquire independently of the user. The remaining information was largely information specific to the customer and/or the policy that he/she wished to purchase; as such, this information is best left up to the user, as he/she will most likely be the most familiar with this information. In this way, our project achieved its primary goal of utilizing the resources available to modern smartphones to improve the process of purchasing a policy from Homesite Insurance; the methodologies we developed can be used to provide a customer with the information needed to purchase a policy that he/she is most likely to be unfamiliar with and would otherwise have to investigate while filling out the form.

The final version of our mock prototype served its purpose of demonstrating the ideas produced by our project. However, instead of consisting of a single smartphone application, our final prototype took the form of multiple independent apps, each demonstrating a different function of the complete application. We had started the project developing our prototype as a single application but found that incorporating all of the individual features and components together into a larger, all-inclusive application was causing serious delays. As such, we divided the prototype's functionality amongst several smaller applications, having found this to be considerably simpler as we developed these features on their own. In this way, we developed single-purpose applications that fully implemented and demonstrated different features of our prototype. These include a number of function-demonstration apps, such as those for address lookup and replacement cost calculation, and a concept-demonstration app that demonstrates the process of using the app from start to finish without going too deep into the operation of the individual features. These applications and the ideas behind them are discussed at length in this chapter.

5.1: Platforms

During the development of our prototype we focused on the Android operating system. The phones we tested our apps on were a mix of Android and Samsung models that ran a range of Android OS versions from Froyo 2.2 to Jelly Bean 4.2. In the end we finalized our prototypes using the API for Android version Jelly Bean 4.1.2, as this version was compatible with most of the phones we were testing with.

5.2: App Layout

The final layout of our prototype, referring in this case to the application that demonstrates the quote acquisition process from start to finish, clearly defines the different steps involved in acquiring a quote for a homeowner's insurance policy. The app in question begins by asking the user to provide the address of his/her house, either manually or by taking a picture using the cell phone's camera. The app then proceeds directly to the data collection screens, which are designed so that their use is intuitive. Once all the necessary data has been collected, the app then proceeds directly to the results screen, which offers the user a clear set of options. In this way, the flow of the app is an intuitive enactment of the abstract process of purchasing an insurance policy. In other words, the process allows the user to identify the client and the property, provide information about them, and receive an estimate as to what a corresponding policy will cost, much as the customer would if he/she were interacting with Homesite Insurance directly or through their online form. The composition of our other prototypes, namely those that are meant to demonstrate individual ideas and features, are also designed to be intuitive while offering a more complete insight into the process than the constraints on the main prototype allow for.

Returning to the "bare-bones" app, an issue that we felt was pertinent to the data-collection screens was the order in which questions were presented to the user. For example, we felt that it was important to ask eligibility questions, such as year of construction, early in the data collection stage. We also felt that it was important to group related questions together, such as roofing material and the year the roof was installed. This latter point also suggests the need to ask questions in a logical order from a "group" standpoint. Moving on, the first information we ask the user for is the house's address. We have this question asked first because we need this information in order to autonomously collect more information. In addition, it takes time for this additional information to be collected, increasing the need to ask this question early. The questions immediately following the first question are selected such that they cannot be autonomously collected, thus increasing the amount of time the data collector can operate before the app reaches a question that could be auto-populated. Once these have been answered, our app asks all unanswered questions, giving priority to those that are "necessary" to return a quote, i.e.

any information without which a quote estimate cannot be calculated, such as square footage. Once all questions have been asked and all information provided and confirmed, the application proceeds to the results screen as expected.

5.3: Data Collection

Our project's collection of prototype applications includes two that demonstrate data-collection components. The first contains an address acquisition function designed to determine an address using a GPS stamp. The second features a web scraper designed to query the Worcester Public Records (Worcester, 2013) for information about a property. These are described in detail in the following subsections.

5.3.1: Collectible Data

The data-collection components of our prototypes gather data in three stages. The first stage is to acquire a GPS stamp from the host smartphone's GPS unit, typically during the act of taking a picture. In the second stage, the geographic coordinate component of the location stamp is used to acquire a street address from Google's Geocoder service. If the GPS stamp could not be acquired or provides an incorrect address, the user must enter/correct the address manually. Once the address is acquired, the third stage uses it in conjunction with our web scraper mockup to retrieve property information from the Worcester Public Records (Worcester, 2013). The information thus provided includes parcel values, current and previous owners, land use, construction details, building valuation, and extra features. If the web scraper cannot retrieve any information from the City of Worcester that corresponds to the property's address, the user will be asked to provide all of the information the app requires.

5.3.2: Address Determination

One of the secondary prototypes we developed was designed to test our address-lookup process, which seeks to acquire the user's address without requiring the user to enter it manually. Our app achieves this by acquiring a GPS stamp directly from the smartphone's GPS unit, then using Google's Geocoder utility to acquire the appropriate address. In order to properly test this feature, this prototype is designed solely to carry out the process described above and display the result to the user. For the results of our tests, see Appendix M: Address-Lookup Test Results. These results allowed us to determine that the methodology used in this sub-prototype was largely accurate, returning the complete correct address twenty out of twenty-three trials and, when incorrect, only being mistaken about the house number.

As a result of this testing, we decided that, while well worth implementing, the address acquisition component needed to be incorporated into the app such that the user could adjust the returned address if need be. For example, if only the address' house number component was incorrect, then the user would only need to change that number, leaving the street, city, state, and zip code unchanged. As such, we decided that the address-confirmation screen should be implemented such that the user can correct or confirm the address in a single action. This minimizes the amount of input required of the user while guarding against errors in the collected data.

5.3.3: Data Vendors and Web Crawlers

In order to demonstrate our idea of using web crawlers to collect data about a property, we sought to develop web scrapers designed to query the Worcester Public Records (Worcester, 2013), Zillow (Zillow.com, 2013), and Trulia (Trulia.com, 2013) for information about a given properties when given its address. For a comparison of the data provided by these three databases, see Appendix K: Information Provided by Online Databases. However, because Zillow and Trulia are for-profit businesses, their websites are designed to passively prevent such activities. At the time of our project, this was accomplished by generating the data returns such that the element names were obfuscated and the format of the returned data changed from query to query. This allowed the data to be displayed and read by a human inquirer, but prevented a program from reliably deconstructing the data. As a result, we were unable to develop web scrapers that worked on their websites. In the end, we were able to develop a usable web scraper for the Worcester Public Records, though the design of the website significantly reduced its functionality. To elaborate, in order for the scraper to successfully query an address, the query had to first be conducted via a standard web browser in order to acquire the unique database identifier for that particular property. For further discussion on our web crawler, refer to Appendix I: Discussion on Coding our Web Crawler

Though the web scraper we developed has significant functionality restrictions, it is capable of successfully querying the Worcester Public Records. Upon being queried using an address' unique identifier, the website returns a webpage with information about the specified property's parcel values, current and previous owners, land use, construction details, building valuation, and extra features. Because this webpage has a consistent format, its components can then be programmatically analyzed in order to derive information that our app needs in order to provide a replacement cost estimate. That being said, should the Worcester Public Records introduce a change into their system, our web scraper may become unable to carry out this functionality. In other words, our scraper was only able to query the Worcester Public Records as it existed during the execution of our project.

5.3.4: Reducing Questions to Ask

Our primary focus over the course of this project was to devise a set of means of reducing the amount of information that a customer of Homesite Insurance is required to provide when purchasing an insurance policy. Our efforts for achieving this goal largely centered on acquiring this information independently of the user by utilizing the different resources available to modern smartphones. Using the data collectors we developed, namely the address-acquisition component and the web crawler, we were able to develop a prototype that autonomously collects first an address and then property information. As a result, the user only has to confirm and/or correct this information rather than produce it.

In this way, by collecting data autonomously the prototype can automatically fill in different fields of the form. For example, using the web scraper previously described, the construction details for a property, such as square footage, number of stories, wall material, and roofing material, can be collected without requiring the user to provide the information. As a result, the user only has to confirm the answers to these questions, effectively reducing the number of questions that the user has to actually answer. In addition, as noted before, these property values are the ones a customer is least likely to have on hand; as such, autonomously providing the user with these values is a highly useful feature that can greatly improve the user's experience.

5.4: Calculating a Quote

For the purposes of our project, we decided to calculate the replacement cost of a house in lieu of the cost of an insurance policy. We made this decision, recommended by our sponsor from Homesite Insurance (Mousseau, 2013), in order to simplify our prototype without sacrificing details from the overall design. To elaborate, fewer data are required in order to calculate a replacement cost than the cost of an insurance policy, while all of these data are also required for pricing the insurance policy itself. As such, no "new" data are required and the type of data being collected will be largely unchanged. In addition, because a house's replacement cost is a large factor in the cost of an insurance policy for that house, the substitution makes conceptual sense as well. Another consideration in this matter was that insurance companies such as Homesite Insurance use personalized, often complex formulas for determining the price of an insurance policy. Such formulas are well beyond the scope of this project, while the formulas for replacement cost estimates are generally more ubiquitous and much simpler. The formulas we employed in our project were derived from a collection of building construction cost data published in 2011 by Reed Construction Data Inc. (RSMeans, 2011). For a description of the formula we developed, refer to Appendix J: Sample Replacement Cost Calculation. For an analysis of how accurate our replacement cost estimates were, using replacement costs provided by the City of Worcester (Worcester, 2013) as a basis for comparison, see Appendix N: Calculated Replacement Cost Accuracy.

With this understanding of our calculations, we shifted our attention to another feature of our calculation component: the ability to request a replacement cost before all relevant data has been collected. This feature, developed in response to feedback we received from our initial user interviews, allows a user to receive a replacement cost before he/she has entered all of his/her information in exchange for a less-accurate quote. In order to provide a

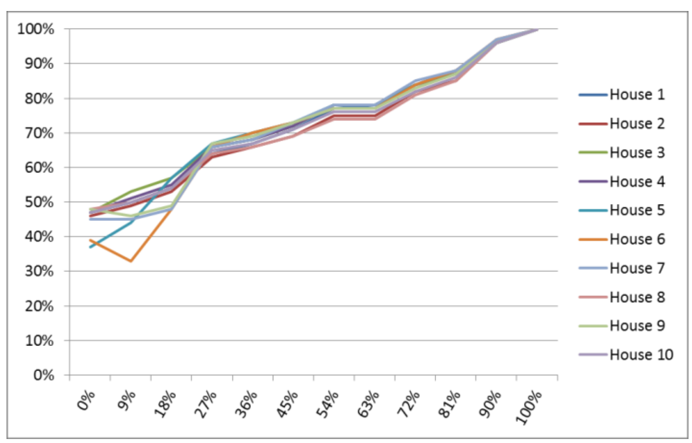
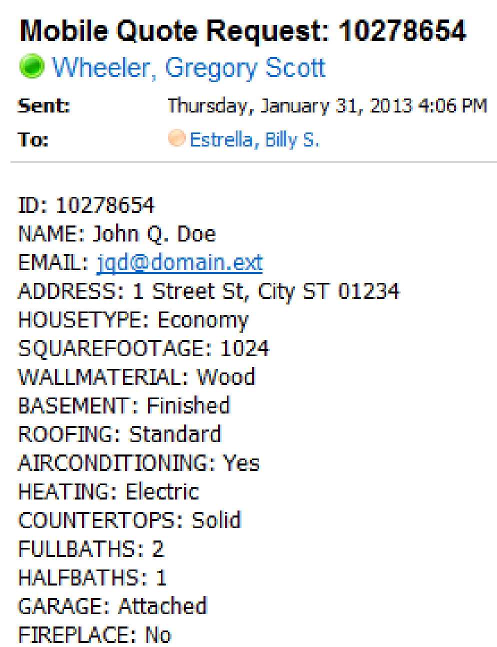


Figure 12: Quote Progress versus Estimate Accuracy

percentage value of the quote's inaccuracy, we added a feature to our calculation component that returns "best-case" and "worst-case" estimates based on what data has not yet been provided. These estimates are then used in conjunction with the returned estimate to provide a value for our accuracy meter to display. For a graph that shows how the accuracy of our estimates improved as the number of questions answered increased, refer to Figure 12 on the previous page. In testing, we found that our prospective calculations provided a decent "ballpark" estimate for a home's replacement cost while our accuracy meter adequately represented the accuracy of the actual figure provided. See sections 4.3.1: Cost Calculator and 4.3.2: Progress and Accuracy Meters in the Implementation chapter for more information on these features, as well as Appendix J: Sample Replacement Cost Calculation.

5.5: Purchasing a Quote

Our prototype, being a proof of concept, is meant to be demonstrative as opposed to functional. As such, we chose to represent the ability to purchase the policy presented by the app with the ability to call a preset number or to send an email containing the information gathered by the app, including the cost estimate, a simulated session ID, and the data gathered by the app, from the host phone (see Figure 13 for a screenshot of one of these emails). Because this part of the final app will be shaped largely by how Homesite Insurance implements the app into their business model and by the support infrastructure they develop to interact with the app, we decided not to develop any external support of our own.



Mobile Quote Request: 10278654

Wheeler, Gregory Scott

Sent: Thursday, January 31, 2013 4:06 PM

To: Estrella, Billy S.

ID: 10278654
NAME: John Q. Doe
EMAIL: jqd@domain.ext
ADDRESS: 1 Street St, City ST 01234
HOUSETYPE: Economy
SQUAREFOOTAGE: 1024
WALLMATERIAL: Wood
BASEMENT: Finished
ROOFING: Standard
AIRCONDITIONING: Yes
HEATING: Electric
COUNTERTOPS: Solid
FULLBATHS: 2
HALFBATHS: 1
GARAGE: Attached
FIREPLACE: No

Figure 13: Email containing faux quote details

We instead chose to demonstrate the ability of our prototype to communicate with other devices and services as a means of acting on the quotes it creates.

Chapter 6: Conclusions and Recommendations

This project was proposed in response to Homesite Insurance's intent to create a smartphone version of its current online form for purchasing homeowner's insurance. The purpose of this project was to develop ideas for enhancing this purchasing process through the use of resources available to modern smartphones. To this end we brainstormed means of collecting data independently of the customer, concurrently developing a mock prototype of the smartphone application previously mentioned as a catalyst to our process. In this chapter we will discuss some of the conclusions we came to regarding the development of the true application, as well as our recommendations for further developing this application. This chapter will also cover some of the issues necessary to consider when developing a smartphone application.

6.1: Platforms

The question of which platform to design the official smartphone application for is a complex one. The programming language used to write the application will have to be capable of handling the data processing required by the components of the app and will optimally be one that enjoys significant use. While continuous improvement of the capabilities of smartphone operating systems reduces the problems caused by the first requirement, the latter requirement is more problematic, as the prevalence of a particular mobile OS varies greatly with time. To illustrate, a report presented to the 2009 International Symposium on Information Engineering and Electronic Commerce claimed that roughly 47% of smartphones ran Symbian operating systems in 2008, while Apple, RIM, and Microsoft claimed 17%, 15%,

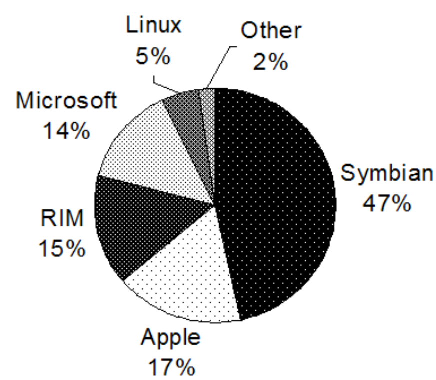


Figure 14: Market share of smartphone OS in 2008

and 14% of the market share respectively (see Figure 14 on the previous page) (Lin & Ye, 2009). In October of 2008, Google started selling phones that ran its Android operating system. In the four years since then, Android has become one of the most powerful and widely-used mobile operating systems (Android.com, 2013), ousting the Symbian smartphone platform from its leading position late in 2010 and claiming 36 percent of the market share of 2011's first quarter (Gartner, 2011). This development, taking place over as little as three years, illustrates how quickly the smartphone market can change in response to new operating systems and other factors.

In light of this rapidly-changing market, Homesite Insurance's development program will have to be tailored to the operating systems that are in popular use at the time of development. If conducted soon, this will likely be the Android OS, as its widespread use and its support from Google make it a popular OS among developers, increasing the chance that apps developed for the Android OS will be supported for a decent length of time. Homesite Insurance will, however, need to continue to monitor the OS's being used to run its application and will need to make sure that their app is compatible with both new OS's and new releases of current OS's. They will also need to monitor the availability of the OS's API (application programming interface), which will largely determine the ability of different phone models to utilize apps designed for the OS in question.

6.2: App Components

Our main prototype presents most of the components that the official app will require. However, our prototypes are simply bare-bones implementations intended to help us develop our ideas and our understanding of the project. As such, a number of features that will be present in the complete app were left absent or incomplete in our prototypes. In developing the final application, Homesite Insurance will need to refine the operation, presentation, and navigation infrastructure of the different components demonstrated by our prototypes. They will also need to develop additional components from the ground up, such as database usage, server-side operations, customer support, and error-checking. These different components are described at length in the following subsections.

6.2.1: Database Usage and Server-Side Operations

Because our prototypes were intended as proofs of concept, we investigated the ability of our prototype to communicate with an external database but did not make it a functional component of any of the individual apps, choosing instead to store all data locally. In developing the official app, we recommend that Homesite Insurance design the app to store the data it collects using an external database. This will relieve the app of the responsibility of storing information about a user's quotes. In addition, the database will provide a measure of security for the data, as the data will be preserved in the event of the smartphone running the app crashing or otherwise going out of commission. Most importantly, it will facilitate the performance of calculations and data collection independent of the abilities of the smartphone running the app.

To elaborate, we recommend that Homesite Insurance export the app's data collection and price calculation to an external server. By having a dedicated server collecting data on a property, the app is relieved of the processing demands of autonomous data collection. Because of the possible variety of smartphones that may run the final app, not having to rely on the host smartphone's resources is an important point to consider. Utilizing an external server also allows the calculation process to be handled server-side as well, which is advantageous for two reasons. Firstly, the autonomously-collected data wouldn't need to be sent to the app for the calculation to take place; all data could simply be sent to the database. Secondly, by keeping the calculations server-side, Homesite Insurance can limit the exposure of its policy calculation formulas, which are the company's intellectual property (Mousseau, 2012).

6.2.2: Customer Support

As development tools, our prototypes were meant to demonstrate and develop only the functional components of the overall app. As such, there was little in the way of simulated customer support. Because of the importance of customer experience in this project, we strongly recommend that Homesite Insurance pay special attention to developing this part of the app.

A potential feature we considered while developing the app is a pervasive help feature. Through further brainstorming we identified two potential uses for such a feature: guiding the user through the process of using of the app and explaining the different questions to the user. The first use is rather straightforward; the feature would provide instructions designed to explain the purpose and use of each screen to the user should he/she desire such an explanation. The second use is more involved; for the data-input screens, the help feature could offer detailed explanations of the different questions in the event that the user is unclear about what the question is asking for. During our analysis of Homesite Insurance's current online form, we found that several questions provided a drop-down that defined or clarified the terms of the question (see Figure 15 for an example). This feature, if implemented in Homesite Insurance's mobile application, could be a major asset in ensuring the accuracy of the data the user provides, as it reduces the app's reliance on the user being knowledgeable about real estate.

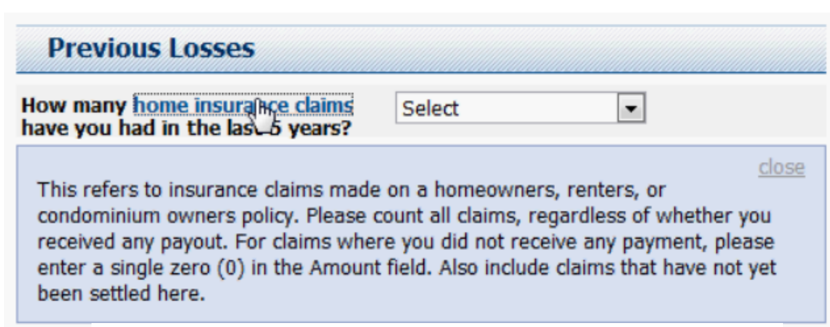


Figure 15: A drop-down textbox containing a clarification

6.2.3: Error Checking

During the discussion regarding autonomous data collection we mentioned the need to confirm the calculated address with the user before using it to collect property data. This act of confirming the address is particularly important because, if it is incorrect, it can result in the data collector collecting data on the wrong property, thus wasting time and resources. The need to confirm the address also highlights the need to confirm the data collected independently of the user; if a value is incorrect, the user needs a chance to catch the mistake before it is used in the application's calculation component. This can be achieved by presenting the user with any and all auto-populated fields at the start of the data-collection process and asking him/her to confirm and/or correct the data as appropriate. The optimal implementation of this feature would allow the user to edit data fields independently and to confirm data fields collectively, thus supporting smooth functionality in all situations while reducing the overall number of actions required of the user.

In addition to the ability to correct erroneous fields immediately after they have been auto-populated, the app needs to give the user the ability to correct mistakes as data is being collected and after all data has been collected. To elaborate on the former, if the user answers a question and proceeds to the next one before he/she realizes that he/she made a mistake, a good user interface would allow the user to go back to the question and change his/her answer. In order to satisfy the latter, the results screen needs to allow the user to view all information collected and edit a field if need be. Another useful function would be the prevention of obvious or logical errors, such as a negative number for "number of additional bathrooms." A final word on this feature is the need to understand how changes to the values provided for different fields affect those of other fields. For instance, Homesite Insurance will need to decide how their application will handle the scenario of a user allowing the app to collect data using one address and then changing the address field value to another address.

6.2.4: Look and Feel

In addition to functionality, an important component of a smartphone app is the look and feel of the interface. According to a research paper published in The TQM Journal, user experience when using smartphone applications is greatly influenced by the aesthetics of the smartphone being used (Nanda et al, 2008). Given the importance of the aesthetics of the smartphone being used, it stands to reason that the aesthetics of the app itself are also important to user experience (Stone et al, 2005). In developing our main prototype, we decided to implement a style of our choosing across the different screens (for a screenshot of a screen with this style, see Figure 16). Because our prototypes were meant for functional development rather than design exploration, however, this was meant only to demonstrate the ability to apply such a style. As such, we leave the final design style of the app's interface up to Homesite Insurance. Based on the findings of the aforementioned research paper, we would recommend that they focus on making an interface that is both simple and pleasing to look at.

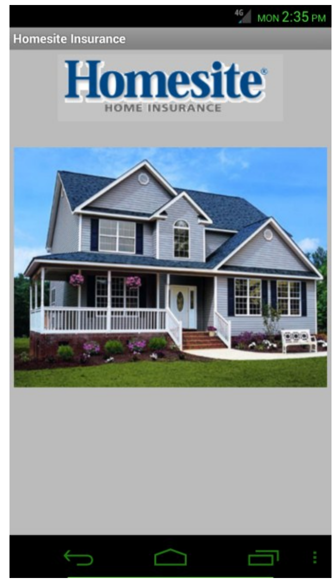


Figure 16: Screenshot with our stylesheet

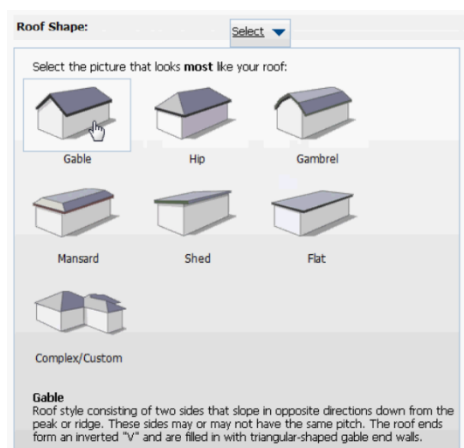


Figure 17: Selection window for roof shape

In addition to being aesthetically pleasing, the design of the final application will also need to be both intuitive and easy to use. An interface that fails to meet these criteria, either by being confusing or difficult to use properly, often results in poor user experience. In our analysis of Homesite Insurance's online form, we felt that, though the form was rather extensive, it was easy to use and incorporated a number of useful tools for entering data (see Figure 17 (previous page) and Figure 18 for two of these data-entry methods). We also appreciated the incorporation of visual cues to indicate different process-related properties about the different fields (see Figure 19 for an auto-population demarcation and Figure 20 for an invalid-value warning). However, because we conducted our analysis on laptops, these controls may need to be re-evaluated when transferred onto a smartphone app. For example, while a keyboard is an agreeable, easy-to-use device for a personal computer, slide-out keyboards such as the one shown in Figure 21 are less suitable for extensive use. For further ideas and design principles, we recommend referring to (Marchionini, 1991), (Oppermann, 2002), and (Stone et al, 2005).

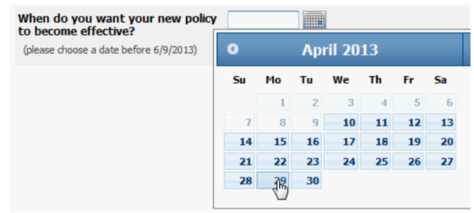


Figure 18: Calendar-based date entry

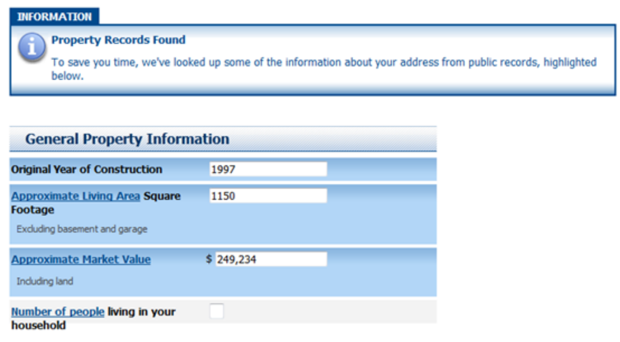


Figure 19: Highlighting of auto-populated data

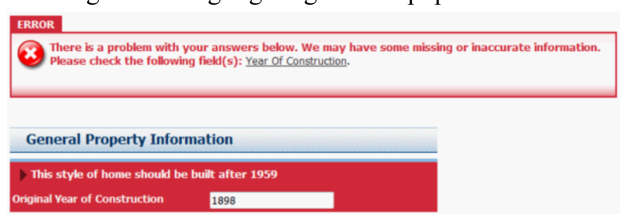


Figure 20: Warning message

Figure 21: Android LG Axis with keyboard

6.2.5: Navigation

Because our main prototype was meant to demonstrate expected walk-throughs of the app, it was not designed to handle all of the inter-screen navigations that an actual user may require.

To elaborate, our main prototype focuses on progressing from the welcome screen, through the data-collection screens, and finally reaching the results screen. When Homesite Insurance develops the official app, they will need to provide more navigational capability to handle the different scenarios that a user may encounter.

For example, the app will need to adjust the navigation options available from the results screen according to how it was reached. To elaborate, the app will need to allow the user to return to the data-collection component in order to alter any collected information, a feature not implemented in our prototypes. In addition, if the user skips to the results screen before providing all necessary information, the app will need to allow the user to return to the data-collection screens in order to fill in all unanswered questions. A final navigational consideration is the navigation options available from the "old quotes" screen. In this case, the screen will need to adjust these options according to the selected quote's degree of completion. For example, if the quote's fields were all completed, the app can progress directly to the "results" screen, whereas if the quote has some unanswered questions, the app may instead proceed to the data-collection screens to ask these questions. In this way, the app needs to be developed such that navigation between its components is functional and intuitive.

6.3: Data Collection

In developing our prototypes, we explored a number of different ideas regarding the collection of property information. Most of these were intended for the gathering of information independent of the user, though some were meant to adjust the questions being asked in response to how previous questions were answered. These different ideas are explored in-depth in the following subsections.

6.3.1: Data Vendors and Web Crawlers

Over the course of our project we developed and utilized a web scraper that queries the Worcester Public Records, an online real estate database (Worcester, 2013), for property information about a specific address. In this way, we can collect information about a property from an external source independent of the user. One of our ideas on this point was the possibility of creating and employing multiple web crawlers that query different data providers. This idea has three points of consideration. First, since the different web crawlers will be asking their respective providers for the same information, the results gathered by the individual web crawlers can be compared and contrasted in order to improve the accuracy of the data used to perform the policy calculations. Second, by querying multiple providers, the app can effectively reduce the impact of one provider not having information about a particular property. Third, since some providers, like the Worcester Public Records (Worcester, 2013), only provide information about local properties, Homesite Insurance may wish to develop region-specific web crawlers designed to use a reliable local provider if appropriate.

In developing our web scraper, however, we learned that this idea is not as straightforward as it sounds. Currently, many real estate databases allow public access to their records. Zillow.com in particular offers users the ability to redisplay information from their database on the user's own website(s). However, these databases are also designed to passively frustrate attempts to scrape their webpages in order to collect information. Some of the techniques used to do this are the obfuscation of the calls to the database, using obscure or complex means to populate page elements, and the variation of how the returned data is structured. In brief, creating a functional web scraper for a project like this will require a major investment of time and effort. In addition, if the website(s) being scraped change the format of their queries and/or returns, the web scrapers Homesite Insurance designs will have to be adjusted in response to these changes whenever they occur.

As such, we do not recommend that Homesite Insurance use web scrapers in the operation of their own application. Instead, we recommend that Homesite Insurance look to collect property information from online resources through means that are properly supported by these resources. For example, Homesite Insurance currently uses a number of data vendors, discussed at length in the background chapter, to gather data about customers looking to buy homeowner's insurance (Mousseau et al, 2012). As such, it would make sense for Homesite Insurance to design their smartphone application (or its server-side support infrastructure) to query these data vendors as a means of collecting property information as well. The details of Homesite Insurance's subscriptions with these data vendors will determine how feasible implementing this feature is.

6.3.2: Reducing and Adjusting Questions

This subsection expands on the idea of answering questions independent of the user. One of the ideas we had considered over the course of the project was the application of acquired information to subsequent questions in order to further simplify the data-collection process. The basis for this idea is the fact that the characteristics of a home are largely determined and/or limited by the house's style (RSMeans, 2011). For example, one of the fields of Homesite Insurance's current online form is the shape of the house's roof. Depending on the style of the house, certain roofing shapes can be ruled out (for example, if the house is of Victorian style, it is safe to say that the roof will not be flat). Thus, the question can be presented to the user with the ruled-out options omitted, leaving fewer options for the user to choose from. In this way, the information gathered by the app can be used to simplify subsequent questions, thereby simplifying the process for the user.

Continuing with the idea of style-specific home characteristics, we observed that the current online form had a number of fields that appeared to be style-oriented. These fields included "ceiling height," "number of rooms with crown molding," and "number of rooms with cathedral or vaulted ceilings." Considering these in the context of the quote calculation, we observed that these questions deal more with the style of the rooms being insured than the number or size of the rooms; thus they are more determinate of the formulas used to calculate policy costs than of the numbers being plugged into those formulas. During the creation of our own cost calculator we found it necessary to ask the user to specify a specific house style in order to determine the replacement cost formula we used. Checking Homesite Insurance's current online form, we found that it also requests the house style as well as asking the questions previously mentioned. We recommend that Homesite Insurance remove the secondary style-related questions if possible and focus on making sure the primary style question is comprehensive and well-explained, carrying this idea into other groups of questions where possible.

6.3.3: Image Processing

One of the ideas for collecting data that our team gave strong consideration to was the possibility of analyzing a picture taken by the user in order to extract different pieces of information about the property. This process of analyzing images is generally referred to simply as "image processing," though the discipline of computer vision is closely related. The idea of using this process was largely inspired by the impressive capabilities that image-processing software has achieved in recent years. For example, a software field known as augmented reality, which involves the incorporation of virtual and "real" environments, has shown immense growth over the past several years. With the immense computing power of mobile devices, many smartphone apps have been developed that use augmented reality. Given the ability of modern smartphones to connect to the internet and their ability to acquire real-time images of the user's environment using built-in cameras, many smartphone apps have been developed to provide information about the area that the user is currently viewing (Prindle, 2013). (See Figure 22 for a sample screenshot of Wikitude's augmented reality Heads Up Display.) For more in-depth discussions of augmented reality, refer to (Pence, 2010) and (Azuma, 1997).

Figure 22: Wikitude screenshot

Because of the complexity of image processing and the time constraints on our project, we chose not to try to develop a prototype capable of performing image processing. We did, however, brainstorm a number of possible uses. One idea we had was to identify a user's house based on image comparisons in order to resolve address ambiguity. To elaborate, the process could start by using the GPS stamp provided by the user's phone to calculate an address. This address could then be used to acquire a "street view" picture of the house at that address from Google. This image could then be compared with the picture the user took in order to determine if they are of the same house, as a large part of image processing is recognizing similarities between images (Partio, 2002). If the images are not similar enough to indicate a match, images of nearby addresses could be requested in an attempt to find a possible match. This idea, though promising, does have its share of shortcomings, namely the unavoidable discrepancies between the images provided by the user and by Google due to differences in environmental conditions, differences in the positions from which the pictures were taken, and any changes that may have occurred between the takings of the two pictures. Whether the benefits of incorporating this feature would be worth the time and challenges involved is ultimately up to those developing the final application.

Another possible use for image processing our team identified is the identification of different elements of the image in order to identify structural characteristics of the home. (For a screenshot of an app designed to detect edges, see Figure 23.) A basic characteristic that could be identified in this way is the number of floors a home has. This could be acquired by isolating the windows of the house and counting the number of distinct rows they form. Another basic element that could be acquired using image processing is the shape of the roof. This could be acquired by having the user point the camera of their smartphone at the edge of the roof so that it can track the roof's edge. It could then use another common feature of image processing, shape identification, to identify the shape of the roof. A secondary idea to this one was to perform the edge isolation in real time using augmented reality, allowing the user to adjust his/her position in order to facilitate the process. Again, whether or not this feature is worth the work involved is up to Homesite Insurance.

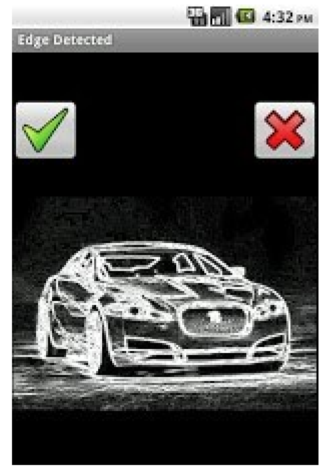


Figure 23: Edge Detector Lite screenshot

Given the scope of image processing and its ever-increasing capabilities, there are probably numerous opportunities for deriving property information from images that we did not come up with. For more information about the different methods of information processing, refer to (Partio, 2002). For a popular open-source project that performs image processing, refer to (OpenCV.org, 2013).

6.3.4 Speech Recognition

A proposal that was made late in the project was the use of speech recognition, the conversion of recorded speech to textual information, as a means of entering data. Specifically, this was identified as an alternative to entering words and numbers via a keyboard. This is an important option to consider, as extensive typing on a smartphone can be quite arduous due to the small size of the average smartphone keyboard. Currently there are many software products being developed that focus on speech recognition, including the Dragon product line (website url: www.nuance.com/dragon), and there are indications that such software is migrating onto smartphone platforms such as Android. However, it is important to ensure that, if it is implemented, speech recognition is clearly offered as an alternative to keyboard input, rather than as a replacement; if a user has difficulty using speech recognition, he/she should not feel compelled to use it.

6.4: Quote Calculation and Purchase

In order to simplify the calculations performed by our prototypes, we chose to provide the user with an estimate of the house's replacement cost instead of the cost of an actual policy. Homesite Insurance, when they develop their own version of the app, will need to design it so that it provides an actual quote that can then be used by Homesite Insurance to sell the user an insurance policy. This will involve using the formulas that Homesite Insurance uses in order to calculate the costs of their policies. Because these formulas are considered intellectual property and are generally withheld from public knowledge, we recommend that Homesite Insurance does not incorporate their formula directly into the app. Instead, we recommend that they implement their formulas on a dedicated server under their control. This will isolate the user from the implementation of the formulas and will also allow Homesite Insurance to adjust their formulas more easily, as they will only be implemented in one piece of software.

Another aspect of the quote provision process to consider is the process of purchasing a policy using the quote provided by the app. To elaborate, Homesite Insurance will need to implement and incorporate a means of purchasing an insurance policy, either through direct payment as Homesite Insurance's online form currently allows (Homesite Group Inc, 2012) or by contacting a Homesite Insurance agent in such a way that the agent is automatically informed about the policy being offered to the user (Mousseau, 2012). Either way, both the app and the infrastructure will have to be designed to work together to perform the operation(s).

Something important to consider here is that, in our implementation of the application, the user is capable of reaching the results screen before all information has been collected in order to receive a quote estimate with an understood margin of error. If Homesite Insurance decides to support this feature in their own implementation of the app, they will need to do so such that the user cannot purchase a policy before its final cost has been determined. One way to handle this precaution would be to simply block the purchase options until all questions have been answered. At the same time, it would be feasible to allow the user to contact Homesite Insurance via the app in order to act on the information currently gathered without acting on the quote estimate itself.

References

Adobe Systems Inc. *PhoneGap*. Retrieved 3/11, 2013, from http://phonegap.com/

Albanesius, C. (2011). “Smartphone Data Use Skyrockets”. *PC Magazine Online*, DOI: GALE|A259383392

Android.com. *Android*. Retrieved 3/11, 2013, from http://www.android.com/about/

Azuma, R. T. (1997). “A Survey of Augmented Reality”. *Presence-Teleoperators and Virtual Environments*, *6*. (4), 355-385.

Chowdhury, R. *Evolution of Mobile Phones: 1995 - 2012*. Retrieved 2/16, 2013, from http://www.hongkiat.com/blog/evolution-of-mobile-phones/

Eclipse Foundation, The. *The Eclipse Foundation open source community website*. Retrieved 3/11, 2013, from http://www.eclipse.org/

Garritty C, El Emam K. (2006). “Who's using PDAs? estimates of PDA use by health care providers: a systematic review of surveys”. *J Med Internet Res*. 2006 May 12;8(2):e7.

Gartner. (2011). “Gartner Android Smartphone Market Share Accelerates 604300.” *CoInsight*, DOI: GALE|A256915256

Homesite Group Inc. *Homesite Home Insurance*. Retrieved 10/14, 2012, from http://www.homesite.com/

Hostetter, K. *Backpacker Magazine - Real GPS Unit vs. Smartphone GPS Apps*. Retrieved 2/16, 2013, from http://www.backpacker.com/gear/ask_kristin/319

Iglehart, J.K. (2002). "Changing Health Insurance Trends". *The New England Journal of Medicine (0028-4793)*, 347 (12), p. 956.

Lambert, S. (2012). *ERA Real Estate adopts augmented reality to connect with potential homebuyers*. Retrieved 3/19, 2013, from http://www.qrcodepress.com/era-real-estate-adopts-augmented-reality-to-connect-with-potential-homebuyers/857529/

Lennighan, M. (2009). “GPS smartphone adoption growing”. *Total Telecom Online*, DOI: GALE|A204622230

Lin, F. & Ye, W. (2009). “Operating System Battle in the Ecosystem of Smartphone Industry”. (Electronic). *2009 International Symposium on Information Engineering and Electronic Commerce*: DOI: 10.1109/IEEC.2009.136

Marchionini, G. (1991). “Psychologoical Dimensions of User-Computer Interfaces”. *ERIC Digest*, (ED337203), 3/19/2013.

MIT. (2012). *MIT App Inventor*. Retrieved 11/15. 2012, from http://appinventor.mit.edu/

Mousseau, P., Miller, J., Estrella, B., & Wheeler, G. (2012-13). Homesite Insurance ’13 MQP: Communications with sponsor.

Nanda, P., Bos, J., Kramer, K.-L., Hay, C., & Ignacz, J.. (2008) "Effect of smartphone aesthetic design on users' emotional reaction: An empirical study", *The TQM Journal*, Vol. 20 Iss: 4, pp.348 - 355

Nelson, P. (2012). *Technology News: How To: Souping Up Your Smartphone's GPS*. Retrieved 2/16, 2013, from http://www.technewsworld.com/story/76478.html

O'Bannon, I. M. (2008). “Scan, Organize ... and Now Auto Population”. *CPA Technology Advisor*, 18(5), pp.52-58. ISSN: 10688285

OpenCV.org. *OpenCV [Open Source Computer Vision]*. Retrieved 1/3, 2013, from http://opencv.org/

Oppermann, R. (2002). “User-interface design”. *Handbook on information technologies for education and training*, 233-248.

Oulasvirta, A., Rattenbury, T., Ma, L., & Raita, E. (2012). “Habits make smartphone use more pervasive”. *Personal and Ubiquitous Computing*, 16(1), pp.105-114. DOI: 10.1007/s00779-011-0412-2

Partio, M. (2002). *Content-Based Image Retrieval Using Shape and Texture Attributes*. Master of Science thesis for Tampere University of Technology.

Pence, H. (2010). “Smartphones, Smart Objects, and Augmented Reality.” *The Reference Librarian*, 52(1-2), pp.136-145. DOI: 10.1080/02763877.2011.528281

Prindle, D. (2013). *Best Augmented Reality Apps*. Digital Trends. Retrieved 3/19, 2013, from http://www.digitaltrends.com/mobile/best-augmented-reality-apps/

RSMeans, (2011). *Building Construction Cost Data 2011*, Book. RS Means, Reed Construction Data Inc: Kingston, Massachusetts, USA.

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User Interface Design and Evaluation*. Morgan Kaufmann.

Sung, D.; Skipworth, H. (2012). *Best Android utilities apps*. Retrieved 2/16, 2013, from http://www.pocket-lint.com/news/42852/best-android-utilities-apps

Trulia.com. *Trulia*. Retrieved 3/23, 2013, from http://www.trulia.com/

Worcester, City of. *Search Public Records*. Retrieved 3/11, 2013, from http://www.worcesterma.gov/e-services/search-public-records

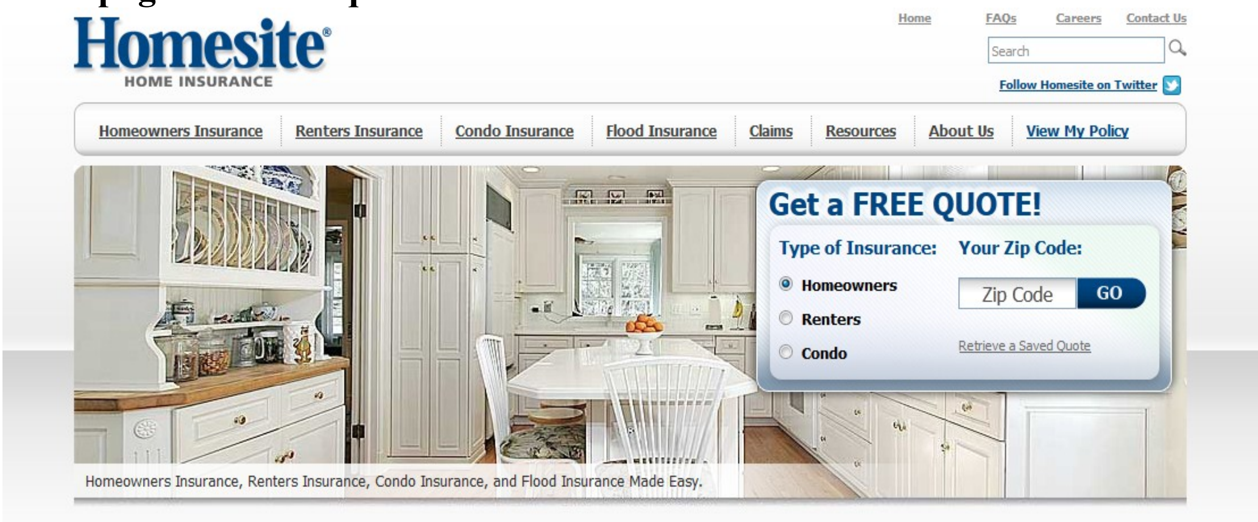
Yin, S. (2010). "Smartphones May Cannibalize GPS Systems". *PC Magazine Online*, DOI: GALE|A242197451

Zillow.com. *Zillow*. Retrieved 3/14, 2013, from http://www.zillow.com/

Appendix A: Current Online Form Screen Capture

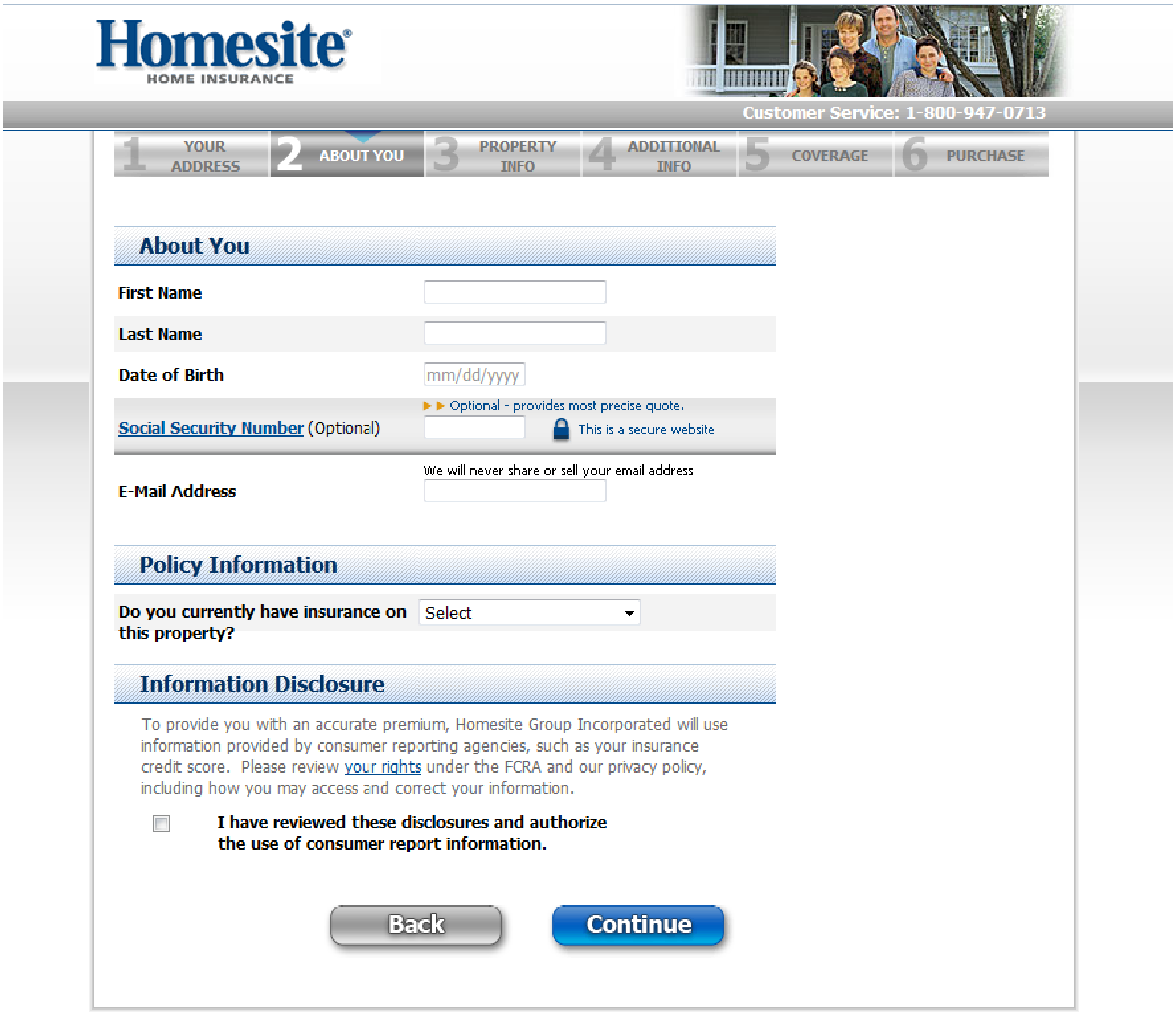
Screen captures taken in October of 2012.

Homepage link for a quote:



Step 1: Your Address

Step 2: About You



Homesite® HOME INSURANCE

Customer Service: 1-800-947-0713

1 YOUR ADDRESS | 2 ABOUT YOU | 3 PROPERTY INFO | 4 ADDITIONAL INFO | 5 COVERAGE | 6 PURCHASE

About You

First Name

Last Name

Date of Birth mm/dd/yyyy

Social Security Number (Optional) ▸▸ Optional - provides most precise quote. This is a secure website

E-Mail Address We will never share or sell your email address

Policy Information

Do you currently have insurance on this property? Select

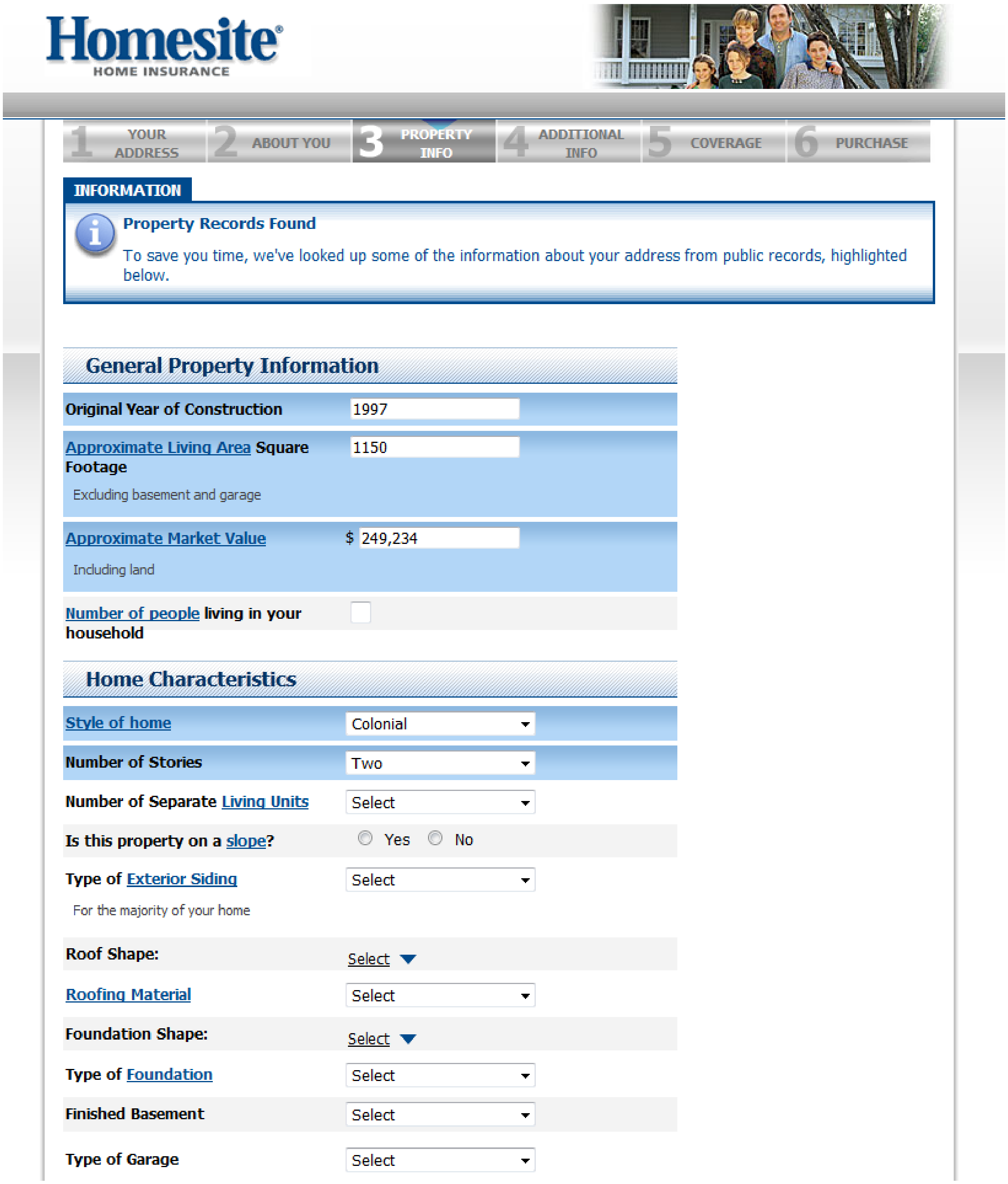
Information Disclosure

To provide you with an accurate premium, Homesite Group Incorporated will use information provided by consumer reporting agencies, such as your insurance credit score. Please review your rights under the FCRA and our privacy policy, including how you may access and correct your information.

I have reviewed these disclosures and authorize the use of consumer report information.

Back Continue

Step 3: Property Info



Homesite® HOME INSURANCE

1 YOUR ADDRESS | 2 ABOUT YOU | 3 PROPERTY INFO | 4 ADDITIONAL INFO | 5 COVERAGE | 6 PURCHASE

INFORMATION

Property Records Found

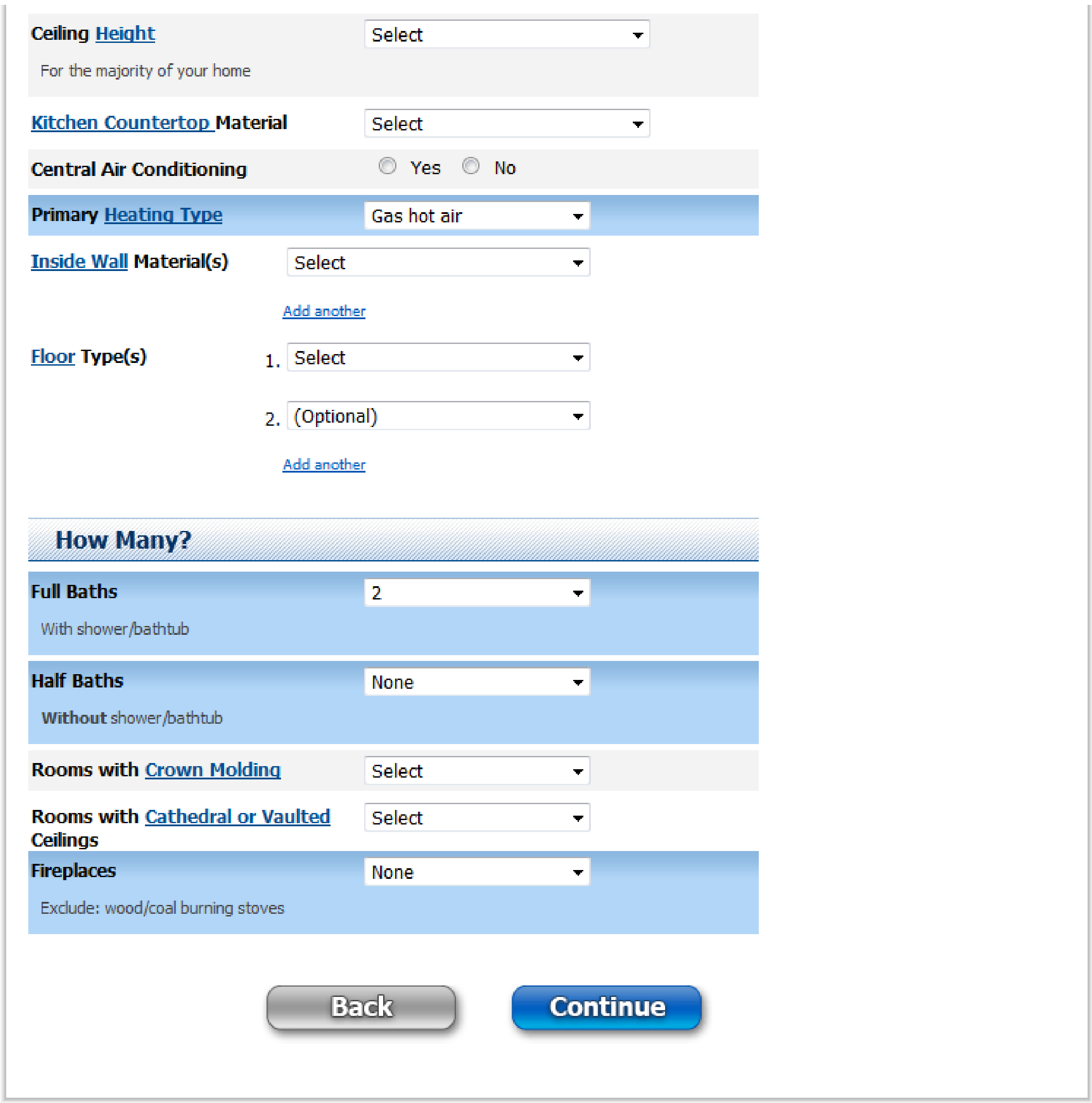
To save you time, we've looked up some of the information about your address from public records, highlighted below.

General Property Information

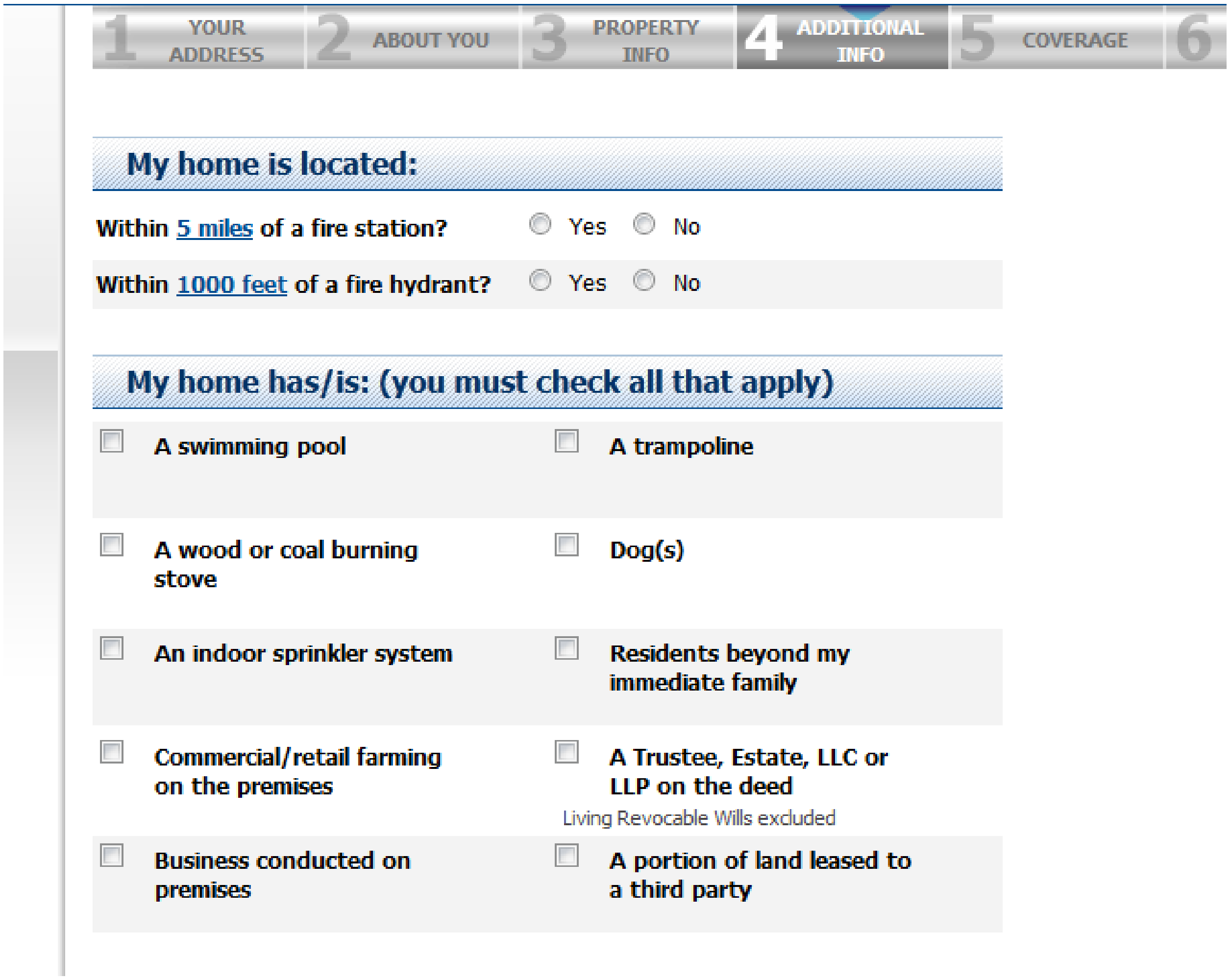
Original Year of Construction	1997
Approximate Living Area **Square Footage** Excluding basement and garage	1150
Approximate Market Value Including land	$ 249,234
Number of people **living in your household**	

Home Characteristics

Style of home	Colonial
Number of Stories	Two
Number of Separate Living Units	Select
Is this property on a slope?	○ Yes ○ No
Type of Exterior Siding For the majority of your home	Select
Roof Shape:	Select
Roofing Material	Select
Foundation Shape:	Select
Type of Foundation	Select
Finished Basement	Select
Type of Garage	Select


Ceiling Height
Select
For the majority of your home
Kitchen Countertop Material
Select
Central Air Conditioning
Yes
No
Primary Heating Type
Gas hot air
Inside Wall Material(s)
Select
Add another
Floor Type(s)
1. Select
2. (Optional)
Add another
How Many?
Full Baths
2
With shower/bathtub
Half Baths
None
Without shower/bathtub
Rooms with Crown Molding
Select
Rooms with Cathedral or Vaulted Ceilings
Select
Fireplaces
None
Exclude: wood/coal burning stoves
Back
Continue

Step 4: Additional Information



1 YOUR ADDRESS | 2 ABOUT YOU | 3 PROPERTY INFO | 4 ADDITIONAL INFO | 5 COVERAGE | 6

My home is located:

Within 5 miles of a fire station? ○ Yes ○ No

Within 1000 feet of a fire hydrant? ○ Yes ○ No

My home has/is: (you must check all that apply)

☐ A swimming pool ☐ A trampoline

☐ A wood or coal burning stove ☐ Dog(s)

☐ An indoor sprinkler system ☐ Residents beyond my immediate family

☐ Commercial/retail farming on the premises ☐ A Trustee, Estate, LLC or LLP on the deed

Living Revocable Wills excluded

☐ Business conducted on premises ☐ A portion of land leased to a third party

Other

Do you want to add another person to your policy as a secondary insured? ◯ Yes ◯ No

What year was your roof installed? [Select ▾]

How many times has your Home Insurance been cancelled for non-payment in the past 3 years? [Select ▾]

Do you have a burglar alarm which is any of the following? [Select ▾]

Local – The alarm can be heard by those inside the home and within a short distance
Central - The alarm is reported to a central monitoring company who contacts police, fire or emergency services
Direct - The alarm is transmitted directly to police, fire or emergency services

Do you have a fire alarm which is any of the following? [Select ▾]

Local – The alarm can be heard by those inside the home and within a short distance
Central - The alarm is reported to a central monitoring company who contacts police, fire or emergency services
Direct - The alarm is transmitted directly to police, fire or emergency services

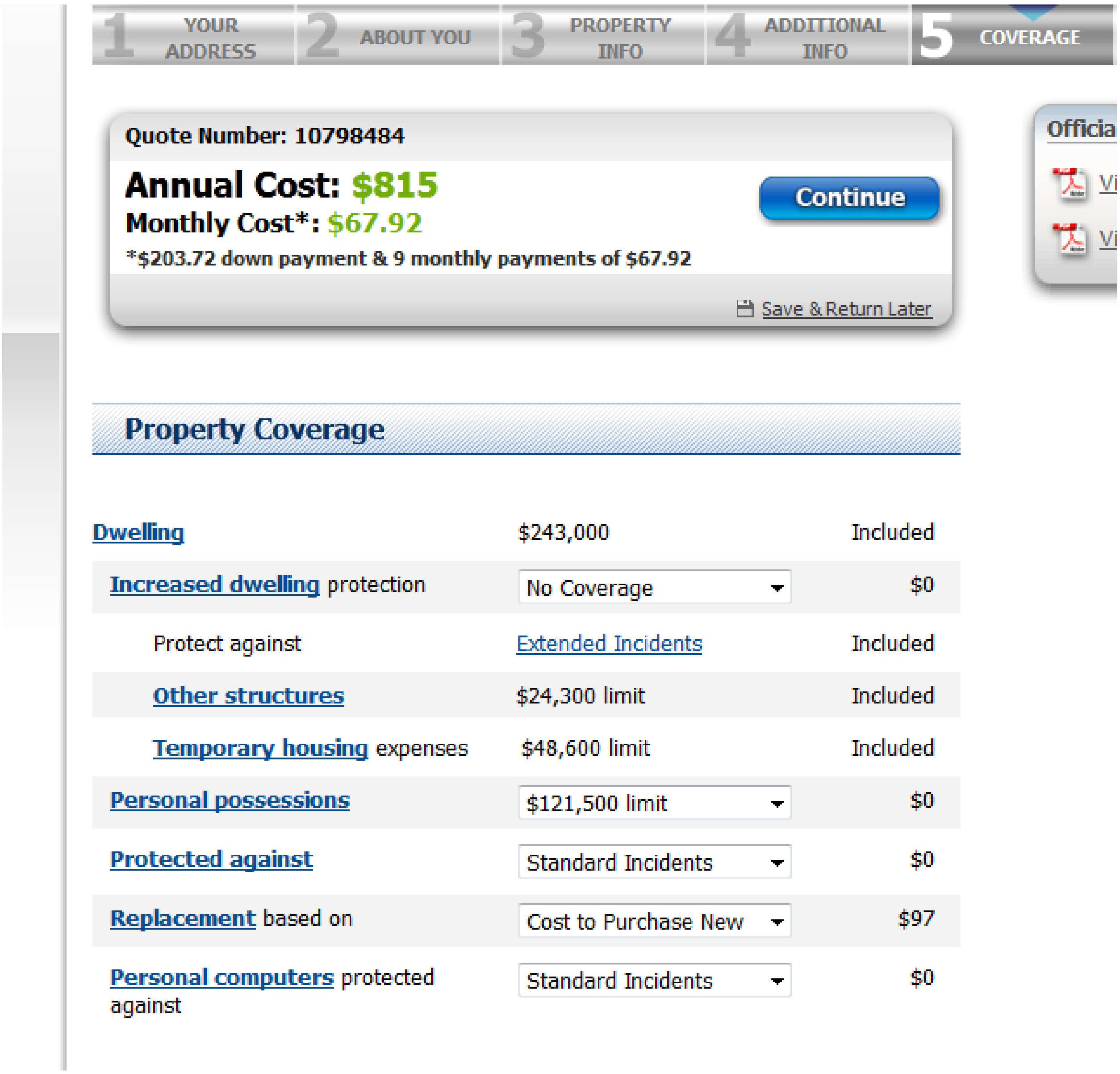
How will you be using this property as of 01/30/2013? [Select ▾]

Previous Losses

How many home insurance claims have you had in the last 5 years? [Select ▾]

Back Continue

Step 5: Coverage



Additional Coverage

Dwelling & Personal Possessions

Earthquake protection	No Coverage	$0
Water backup/sump overflow protection	No Coverage	$0
Identity theft expense protection	No Coverage	$0

Dwelling & Personal Possessions

Earthquake protection	No Coverage	$0
Water backup/sump overflow protection	No Coverage	$0
Identity theft expense protection	No Coverage	$0

Deductibles

Regular loss	$500 deductible	$0

Incident Specific Deductibles

Liability Coverage

Personal liability	$300,000 limit	$30
Medical liability (for visitor injuries)	$3,000 limit	$8
Defamation of character liability	No Coverage	$0

Discounts

Age of Home	-$37
Total	**$815**

Back | Continue

Appendix B: Sections of the Current Online Form

Step 1; Your Address: The form automatically fills in the City, State, and zip code fields using the zip code provided by the access screen. The customer only has to enter the house's street address and the date that the policy is to become effective. As such, this part of the form is relatively straightforward.

Step 2; About You: In this section the user is asked to enter their personal information. This includes the user's name, date of birth, Social Security Number, email address, and current insurance information. Though most of these fields are routine, the presence of the Social Security Number field appears to be a point of contention for most people looking at the form. Through personal communication with our sponsor from Homesite Insurance, himself a senior business analyst, we learned that the Social Security Number is used to obtain a customer insurance risk score from other insurance companies. At the same time, this score is often provided by data vendors as well, in which case only a person's name and address information are required. As such, it may make more sense to present this question to the applicant later in the process, making sure that it is actually necessary and explaining to the user why it is needed.

Step 3; Property Info: Our screenshot of the application process' third stage shows an instance of Homesite Insurance using data vendors to reduce the amount of data that the user must enter manually by auto-populating some of the form's fields. Through communications with our sponsor, we learned that Homesite Insurance already purchases information from a number of data vendors in order to collect data on customers seeking to purchase an insurance policy. In the case of the property information present in this section of the form, Homesite Insurance primarily uses CoreLogic, highlighting the fields that were automatically populated in blue as shown in our example.

This section is divided into three different subsections, as described below.

- *General Property Information*: Basic information about the home, including the year of construction, approximate living area, and approximate market value. Having the ability to automatically fill in these fields can provide the applicant with a useful starting point, as it is generally easier to adjust a ballpark value obtained from a data vendor than it is to calculate the value from scratch. That being said, it is still important to make sure that the data being used is accurate. Our own use of the form indicated that this isn't always the case, an important issue in this section because the data collected here are used in part to determine whether or not the user is eligible for an insurance policy.
- *Home Characteristics*: This is section of the form requires a lot of data. Examining the form, we found that some of the fields of this section offer picture-based selection to assist users with non-trivial questions. These offer a visual representation of the question being asked, reducing the likelihood that a user without detailed knowledge of the subject will make a mistake. This alternative method of gathering information could be useful in other parts of the form as well.

- *How Many?*: This section asks a few quantity-based questions, such as the number of bathrooms, fireplaces, and rooms with certain qualities. The subjects we interviewed expressed the opinion that simply reaching this point was a lengthy task, indicating that much work is needed to improve the usability of the application process.

Step 4; Additional Info: This section asks for information that is not covered in the previous sections.

- *My Home Has/Is*: This section asks about various insurance-related conditions, such as how accessible the home is to fire crews and whether or not the home has a dog, a pool, a wood/coal stove, etc.
- *Other*: This section covers everything not yet covered, including fire/burglar alarms, intended use, previous losses, and other topics of potential interest.

Appendix C: Fields of the Current Online Form

Form access window
Type of Insurance
Zip Code

Step 1; Your Address
Street Address
Apartment, unit, floor, etc.
City
State
Zip Code
When do you want your new policy to become effective?

Step 2; About You
About You
- First Name
- Last Name
- Date of Birth
- Social Security Number
- E-Mail Address

Policy Information
- Do you currently have insurance on this property?

Information Disclosure
- Authorization of the use of consumer report information

Step 3; Property Info
General Property Information
- Original Year of Construction
- Approximate Living Area Square Footage
- Approximate Market Value
- Number of people living in your household

Home Characteristics
- Style of Home
- Number of Stories
- Number of Separate Living Units
- Is this property on a slope?
- Type of Exterior Siding
- Roof Shape
- Roofing Material
- Foundation Shape
- Type of Foundation
- Finished Basement
- Type of Garage

- Ceiling Height
- Kitchen Countertop Material
- Central Air Conditioning
- Primary Heating Type
- Wiring type
- Inside Wall Material
- Floor Type(s)

How Many?

- Full Baths
- Half Baths
- Rooms with Crown Moldings
- Rooms with Cathedral or Vaulted Ceilings
- Fireplaces

Step 4; Additional Information

My Home Is Located

- Within 5 miles of a fire station?
- Within 1000 feet of a fire hydrant?

My Home has/is: (you must check all that apply)

- A swimming pool
- A trampoline
- A wood or coal burning stove
- Dog(s)
- An indoor sprinkler system
- Residents beyond my immediate family
- Commercial/retail farming on the premises
- A Trustee, Estate, LLC or LLP on the deed
- Business conducted on premises
- A portion of land leased to a third party

Other

- Do you want to add another person to your policy as a secondary insured?
- What year was your roof installed?
- How many times has your Home Insurance been cancelled for non-payment in the past 3 years?
- Presence/type of a burglar alarm?
- Presence/type of a fire alarm?
- How will you be using this property in three months?

Previous losses

- How many home insurance claims have you had in the last 5 years?

Appendix D: Sample Preliminary Interview Feedback

Demographic Information

Subject	Gender	Age	Occupation	Tech. Savvy
1	Female	33	Geneticist	Casual user
2	Male	30	Software Developer	High
3	Male	40+	MIS/Product Developer	High
4	Female	49	Accountant	Rudimentary
5	Male	40+	Accountant	Low
6	Male	40+	Consultant; live software training	Average
7	Female	40+	Management; manufacturing	Low
8	Male	32	Mechanical Engineer	High

Up-front questions

If you could get a home insurance quote through your cell phone would you?

If it was easy enough, but how would I fill out the application on my cell phone? How would they inspect my home?

If it was possible to use the camera on your cell phone to help answer questions would that be useful?

If I was shopping for a house, it might be nice to be able to get a quick idea of how much insurance is. If I could just take a picture and get a number that would be really useful. Just so I'd know if the price was outrageous.

When you purchased home insurance how would you describe the process?

I'd say that it's daunting and you really rely on the expertise of the agent. Especially true for my house on Francis St, because it's an old house.

Would you be willing to run through an online quote with me on an insurance company's website?

Yes.

Online form notes: (feedback during walkthrough was requested)

Part 1: Your Address

I probably wouldn't be inside my house shopping for an insurance quote on my house. If I was I don't think I'd use my cell phone to do this application.

Social Security Number: I wouldn't even continue this application if they wanted me to put my Social in. They shouldn't need that just to give me a quote.

Is the home insured now?: you wouldn't be inside the house if you are looking for insurance on a new home. If I was inside my house, I wouldn't be uninsured.

Part 3: Property Info

Really? There's no way I could do this on a cell phone!

Approximate Living Area: How would I use this help feature? It's on a cell phone!

Approximate Market Value: I'd have to look it up, that would be a major pain on my cell phone.

Home Characteristics: (Observation) She started scrolling down the form to see how many questions were left, then made comment: "This is only step 3"

Roofing Material: How would I know this, can't they give me a picture like the roof shape! (Observation) She started to become aggravated at this point of the application. When told she didn't have to finish if she didn't want to, she replied; "It's ok I'll finish it, but I would never do this on a cell phone."

Part 4: Additional Information

Oh my God, REALLY!!!

My Home is Located:

How would I know if the fire station is 5 miles or not!

The fire hydrant is across the street but why are they even asking me this!

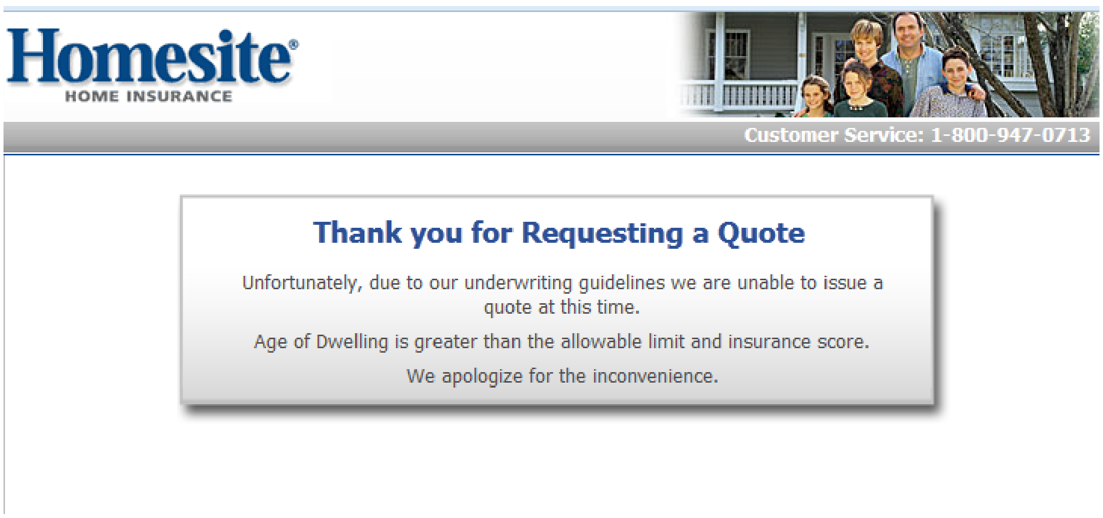
My home has/is:

They want to know if I have a DOG!

What year was your roof installed?: Why is this here! They already asked me the roof questions.

This should be with those questions.

Results:



This is ridiculous! They made me finish the whole thing before they told me my house was too old. They have to be joking.

Appendix E: Sample Property Report

*Note: This property report was acquired from the Worcester Public Records (Worcester, 2013) in November of 2012. All names have been removed from this copy of the report to protect individuals' privacy.

50 FRANCIS ST

MBLU :	12/ 017/ 00017/ / /
Location:	50 FRANCIS ST
Owner Name:	########,########
Account Number:	12-017-00017

Click to enlarge

SEARCH FOR SIMILAR SALE PROPERTIES

Parcel Value

Item	Current Assessed Value	FY 2012 Assessed Value
Buildings	$138,600	$138,600
Extra Building Features	$0	$0
Outbuildings	$900	$900
Land	$47,100	$47,100
Total:	$186,600	$186,600

Owner of Record

########,########

0027 WRENWOOD ST

SPRINGFIELD, MA 01119

Ownership History

Owner Name	Book/Page	Sale Date	Sale Price
########,########	42154/ 222	12/5/2007	$0
######,####### # ## #	39153/ 198	6/12/2006	$247,900

Land Use (click here for a list of codes and descriptions)

Land Use Code	Land Use Description
1010	SINGLE FAMILY

Land Line Valuation

Size	Zone	Neighborhood	Assessed Value
7480 SF	RL-7	103	47,100

Construction Detail

Item	Value
STYLE	Conventional
MODEL	Single Family
Grade:	Average
Stories:	1.75
Occupancy	1
Exterior Wall 1	Aluminum/Vinyl
Roof Structure:	Gable
Roof Cover	Asphalt
Interior Wall 1	Plasters
Interior Flr 1	Hardwood
Heat Type:	Steam
AC Type:	None
Total Bedrooms:	4
Total Full Bthrms:	1
Total Half Baths:	0
Total Xtra Fixtrs:	1
Total Rooms:	7

Building Valuation

Item	Value
Living Area	1,508 square feet
Replacement Cost	184,743
Year Built	1898
Depreciation	25%
Replacement Cost Less Depreciation	138,600

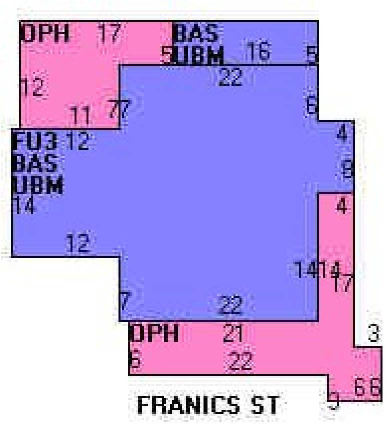
Outbuildings (click here for a list of codes and descriptions)

Code	Description	Units
SHD1	SHED FRAME	80 S.F.

Extra Features (click here for a list of codes and descriptions)

Code	Description	Units
No Extra Building Features		

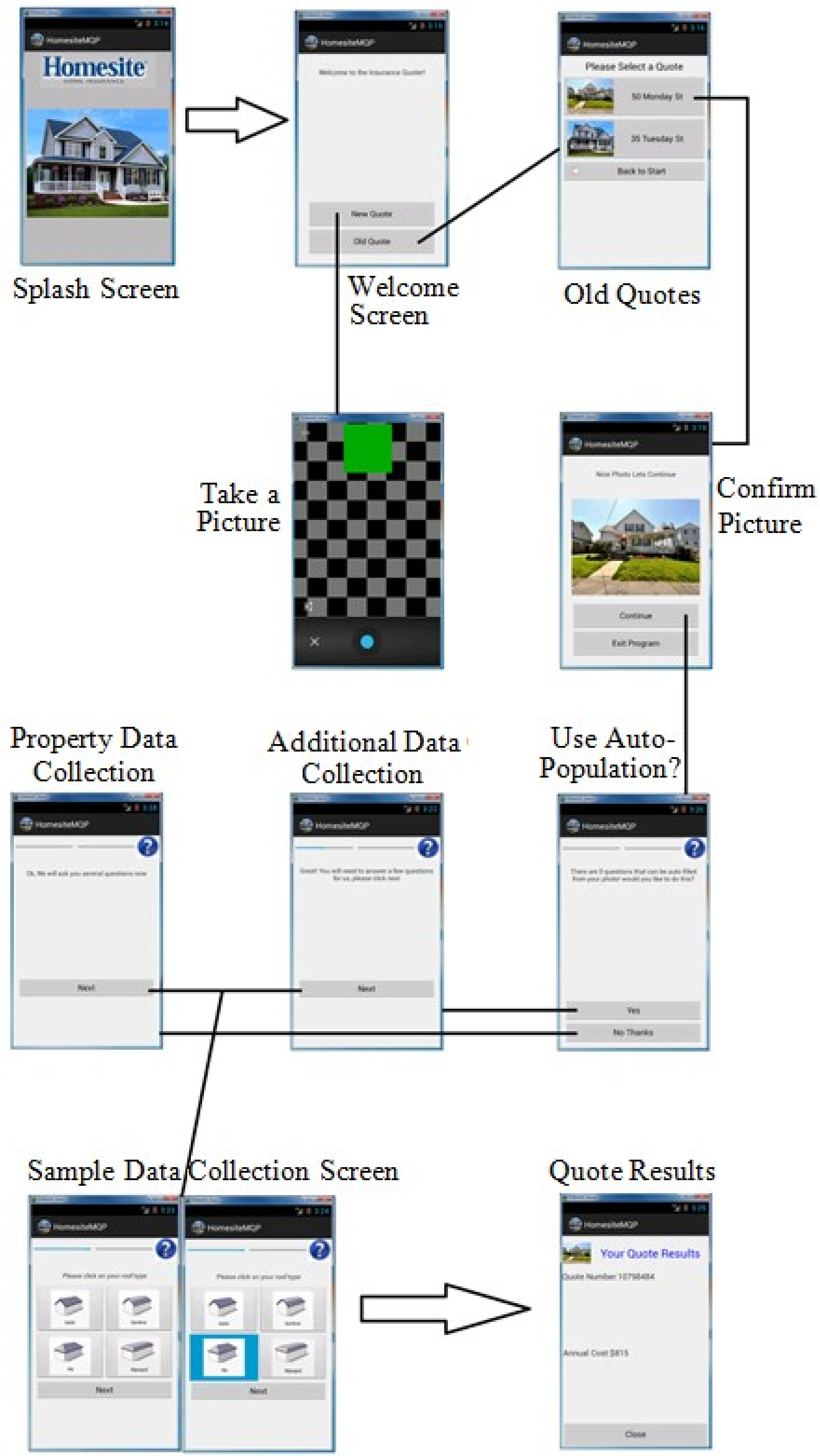
Building Sketch (click here for a list of codes and descriptions)



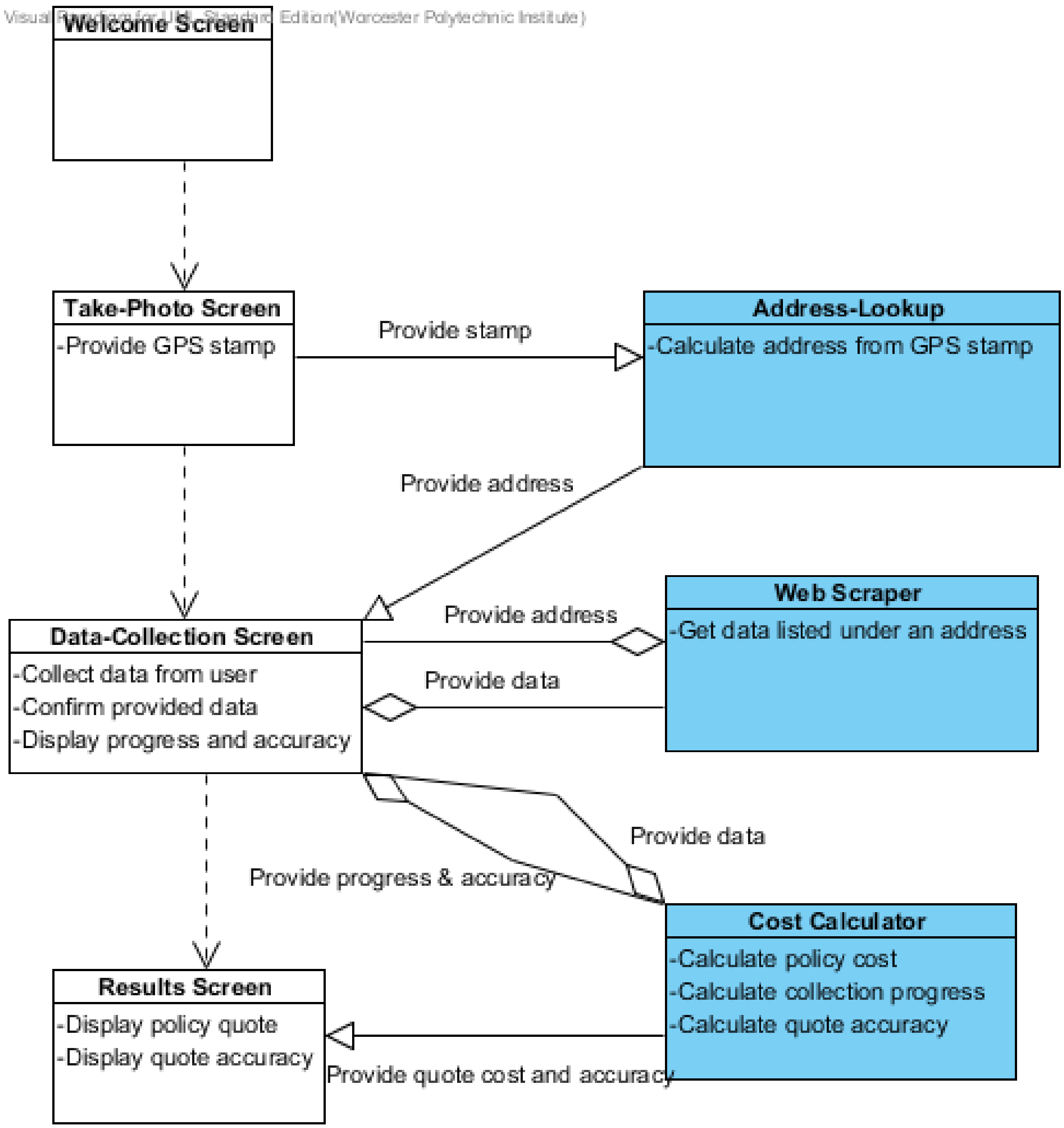
Subarea Summary (click here for a list of codes and descriptions)

Code	Description	Gross Area	Living Area
BAS	First Floor	896	896
FU3	Upper, 3/4 Finished	816	612
OPH	Open Porch	395	0
UBM	Basement, Unfinished	896	0
	Total	3003	1508

Appendix F: Screenshot Flowchart



Appendix G: Modular Diagram



Appendix H: Discussion on Coding the Address Locator

The address determination methodology we implemented in our project operates in a two-step process. First, the process acquires a latitude/longitude location coordinate from the host smartphone's GPS unit. Next, the process uses Google's geocoder utility to translate this coordinate into a street address.

For Android's tutorial(s) on this subject, refer to:

http://developer.android.com/training/basics/location/locationmanager.html

In order to implement out process, we needed to make use of the Android system service "LOCATION_SERVICE." To get access to this service, we needed to:

- Define LocationManager and link to it the System Service LOCATION_SERVICE (android.location.LocationManager, Context.LOCATION_SERVICE)
- Choose a LocationProvider from which to get updates, in our case LocationManager.GPS_PROVIDER
- Define a class that implements LocationLister (android.location.LocationListener) to capture the and process the raw data that comes from the GPS device

Sample Code

```
// define a LocationManger and assign to it the location services LocationManager
LocationManager myLocManager = (LocationManager)getSystemService(Context.LOCATION_SERVICE);

// set up our Location Listener as the listener to receive data from the GPS_PROVIDER using the
// method: requestLocationUpdates(String provider, long minTime, float minDistance, listener)
// minTime (in milliseconds) allows us to define a time interval in which to request updates
// minDistance (in meters) allows us to define a distance interval in which to request updates
// LocationListener allows us to attach our location listener by which we can manipulate data
myLocManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 500, (float)0.5, myListener);

/** myListener is a class that implements LocationListener **/
// when we implement LocationListener we are required to implement 4 methods
public class myListener implements LocationListener{

        @Override
        public void onLocationChanged(Location location) {
        }
        @Override
        public void onProviderDisabled(String provider) {
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
}
```

The onLocationChanged(location) method is where our app collects and stores the information about the current location. This method will be called any time the minTime or minDistance intervals are reached. In our development we stored these locations and then displayed them via the user interface.

Now that the app has the coordinate and other location related information stored, the process can attempt to identify the address closest to that coordinate. Google currently provides a great resource for doing this called the Geocoder class. This class provides processes such as ours with a method called getFromLocation(), which takes in a latitude, a longitude, and a maximum number of returned Addresses. This is demonstrated in the code below.

Sample Code:

```
// define a Geocoder to get location
// the context may be either passed to the current location
// or we can use getApplicationContext() to get the Application's context
Geocoder geocoder = new Geocoder(context, Locale.getDefault());

// check for errtrapping
if (noErrors){
	List <Address> addresses = geocoder.getFromLocation(lat, lon, max);
	// get address information from list addresses.get(0).getAddressLine(i)…
}
```

In performing this process, we were able to collect a street address in addition to a number of other pieces of information. These include altitude, bearing, and various measurements of accuracy. Our implementation of the overall application had no use for the altitude value, though a future implementation might find a use for it. The bearing reading, however, did suggest a means of improving the accuracy of our results. The LocationManager.getBearing() method returned a standard directional bearing, if available. The accuracy of this reading is imperfect, as the bearing is calculated by creating a set of vectors using recent location points (i.e. the path the host GPS unit recently traveled). In our tests we were to acquire a useful bearing by walking 10 feet toward the house we desired information for, thus aligning the aforementioned vectors in the desired direction. During these tests, setting our GPS unit's update interval to ½ meter provided at least two points in the direction of travel. The results of our test across limited samples (21) showed that 19 (90%) of the bearing information was within 88% of the actual "direction of the house," based on the perpendicular angle of the street's bearing. In the context of determining opposite directions (180-degree differences), a large variance may be acceptable for determining the direction in which the phone is facing.

When we tested home address lookups, there were two documented cases where the geocoder returned an address composed of the two houses directly across the street from the house we were seeking to identify. If we had further developed our means of determining the host phone's orientation, it may have been possible to determine the correct house number. In addition, in cases where the address was returned as a range, it may have been possible to use this refined orientation reading to select a "best choice" from this range.

Theories for future Expansion

- Use calibration techniques used by smartphone cameras as a part of autofocusing to approximate distance to the house in question.
- Use a compass tool for direction readings instead of bearing. Complement this by using the accelerometer to assist in calculating orientation.
- Use the methods previously discussed to attempt a "best-guess" as to a correct address when a range of addresses is returned.

Appendix I: Discussion on Coding our Web Crawler

One of our project's demonstrative components was a web crawler we designed to gather information from the Worcester Public Records. A web crawler is simple in operation; it composes a URL using information about the URL structure of the website being queried and then parses the webpage data returned by the server in response to the URL request.

To illustrate this concept, we'll explain how the web crawler we developed over the course of our project operated. We began by accessing the page of a specific property on the Worcester Public Records website using a standard web browser and capturing the request for the page in question. We then gave this request to the web crawler so that it could imitate the request and thus receive the webpage from the server. Once the crawler had the page, it extracted property information from the page data by searching for identifying element names and processing the elements themselves based on the structure of these elements. This searching and processing was made possible by our analyzing the page's structure beforehand and then programming the crawler to automatically extract the information we sought from the returned page data.

Optimally, our web crawler would have been able to generate the page request for any given address in order to receive the webpage containing data about that address. However, the Worcester Public Records associates a unique identifying number with each property it covers, which it places in the webpage request to the server providing the data. Because the full list of these identifying numbers is known only to the Worcester Public Records, it was impossible for us to construct a page request for an abstract address. In other words, the only way to produce the correct page request was to browse the site manually as described above, capture the request made by the site itself, and have the crawler reproduce it.

Appendix J: Sample Replacement Cost Calculation

For this example, we will be calculating a replacement cost for 50 Francis St, Worcester, MA. The replacement cost provided by the Worcester Public Records for this property is $184,743.00.

In order to calculate a replacement cost, we used survey data provided by Reed Construction Data Inc. (RSMeans, 2011). For this house, we specifically used the data pertaining to 2-story houses of "average" style. This data is provided below for your convenience.

Base cost per square foot of living area

	Living Area										
Exterior Wall	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**	**2200**	**2600**	**3000**	**3400**	**3800**
Wood Siding – Wood Frame	124.25	112.50	107.05	103.25	99.35	95.25	92.50	87.15	82.15	79.75	77.70
Brick Veneer – Wood Frame	131.40	119.15	113.25	109.15	104.90	100.60	97.60	91.75	86.40	83.80	81.55
Stucco on Wood Frame	120.15	108.70	103.55	99.85	96.15	92.20	89.60	84.50	79.70	77.45	75.50
Solid Masonry	146.40	133.00	126.20	121.50	116.55	111.85	108.25	101.40	95.35	92.30	89.60
Finished Basement, Add	20.25	20.05	19.35	18.90	18.40	18.10	17.75	17.15	16.70	16.35	16.10
Unfinished Basement, Add	7.90	7.40	7.00	6.70	6.40	6.20	6.00	5.55	5.25	5.10	4.90

Alternatives ***Add to or deduct from the cost per square foot of living area***

Cedar Shake Roof	+ 1.25
Clay Tile Roof	+ 2.25
Slate Roof	+ 3.15
Upgrade Walls to Skim Coat Plaster	+ .57
Upgrade Ceilings to Textured Finish	+ .49
Air Conditioning, in Heating Ductwork	+ 2.64
In Separate Ductwork	+ 5.34
Heating Systems, Hot Water	+ 1.95
Heat Pump	+ 2.54
Electric Heat	– .63
Not Heated	– 2.97

Modifications ***Add to the total cost***

Upgrade Kitchen Cabinets	+ 3528
Solid Surface Countertops (included)	
Full Bath – including plumbing, wall and floor finishes	+ 6252
Half Bath – including plumbing, wall and floor finishes	+ 3768
One Car Attached Garage	+ 13,058
One Car Detached Garage	+ 17,227
Fireplace & Chimney	+ 5952

As the data suggests, the replacement cost of a house is acquired by taking the base cost per square foot as defined by the house style, number of stories, wall material, basement state, and the amount of living area. This base cost is then adjusted according to alternatives such as roofing material, wall and ceiling upgrades, and air conditioning and heating systems. This is then multiplied by the living area to get a base cost. Finally, adjustments are made according to modifications such as kitchen cabinet upgrades, extra full or half-baths, the presence of a garage, and the presence of a fireplace and chimney. The final result is our replacement cost estimate.

To get our estimate started, we have already specified house style and number of stories as average and two respectively. However, in order to begin calculating cost estimates, we also need to provide the amount of living area, 1,508 ft^2. At this point, calculations can start being made. In order to do this with a lack of information, we developed sets of "generic," "best-case," and "worst-case" data to substitute in for data not yet entered. Our substitution data for average houses is as follows (note that living area cannot be substituted, just as home style can't).

Field	Generic	Low	High
Stories	2 stories	Bi-Level	3 stories
Wall Material	Wood	Wood	Solid
Basement State	Finished	None	Finished
Roofing Material	Asphalt Shingle	Asphalt Shingle	Slate
Air Conditioning	Via Heating Ductwork	None	Via Dedicated Ductwork
Heating	Standard Heating	None	Heat Pump
Countertops	(included)	(included)	(included)
Additional Full Baths	One	Zero	Two
Additional Half Baths	One	Zero	Two
Garage Type	Attached	None	Detached
Fireplace	Present	None	Present

Performing the calculations, the following results are acquired:

Current estimate: $217,223.32
Lowball estimate: $151,222.24
Highball estimate: $271,595.44
Current progress: 9%
Current accuracy: 45%

Note that because we already provided the number of stories, the current progress is at 9%, the current progress being calculated by determining how many of the above fields have been filled in (in this case, one out of eleven). The accuracy is acquired by taking the range of the two edge estimates ($120,373.20), dividing it by the current estimate ($217,223.32), subtracting the quotient from 1, and then multiplying it by 100 to convert it from decimal to percentage form. A check is then made to make sure that the percentage is positive; if it is negative, the accuracy is simply displayed as zero.

For the next step in this process, we enter the information that directly affects the cost rate, namely the wall material and the basement state. In this case, these are wood siding with wood framing and unfinished respectively. The newly-computed results are as follows:

Current estimate:	$198,825.72
Lowball estimate:	$161,325.84
Highball estimate:	$225,676.84

Current progress: 27%
Current accuracy: 68%

With two more questions answered, the progress has increased to 27%. Also, because the estimates are using the provided information rather than their respective assumed values, the range has narrowed. As a result, the accuracy has improved by 23%.

Next, we enter the information that further alters the cost rate, namely the roofing material and the type of air conditioning and heating systems present. In this case, we use the values asphalt shingle (standard choice; no alteration), none, and heat pump respectively. The new set of results is:

Current estimate:	$198,674.92
Lowball estimate:	$169,634.92
Highball estimate:	$212,873.92

Current progress: 54%
Current accuracy: 78%

Again, the progress and accuracy have improved. It's worth noting here that the estimates are closing in on $184,743.00, the value provided by the Worcester Public Records.

The final round of entries deals with modifications to the overall price after the cost rate is multiplied by the living area. These include countertops, the number of additional full and half-bathrooms, the type of garage if any, and the presence of a fireplace. In this case, countertops are included and thus are irrelevant (but need to be specified to satisfy the progress meter), there are no additional bathrooms of either type, there is no garage, and there isn't a fireplace. The final set of results is:

Current estimate: $169,634.92
Lowball estimate: $169,634.92
Highball estimate: $169,634.92
Current progress: 100%
Current accuracy: 100%

As one can see, both the progress and accuracy meters are at 100%, all questions having been answered and thus all values necessary for our calculation provided.

Now that we have our own replacement cost estimate, we calculate the estimate's percent error compared to the cost provided by the Worcester Public Records. In this scenario, our replacement cost formula failed to take into account a quality of the house in question that was considered in the calculation of the latter replacement cost. As such, our estimate was $15,108.08 off. The formula we used for the percent error itself is |actual – calculated| / actual. As such, the percentage error in this example is 8.18%.

Appendix K: Information Provided by Online Databases

Information required by Homesite Insurance

Field Name	WPR[2]	Zillow	Trulia
House style[1]	X		X
Family model	X	X	X
Grade/slope	X		
Stories[1]	X		X
Occupancy	X		
Exterior wall material[1]	X		X
Roof structure	X		
Roofing material[1]	X		X
Interior wall material	X		
Flooring type[1]	X	X	
Heating type[1]	X	X	X
Air conditioning type[1]	X		
# full bathrooms[1]	X	X	X
# half bathrooms[1]	X	X	
Exterior buildings	X		X
Living area[1]	X	X	X
Year built	X	X	X
Assessed property value	X	X	X
Building sketch	X		
Floor-specific details	X		
Basement details[1]	X	X	X
Garage		X	X
Fireplace[1]		X	
Foundation type			X

[1]Required for our replacement cost calculation
[2]Worcester Public Records

Additional information

Field Name	WPR[2]	Zillow	Trulia
# bedrooms	X	X	X
Total # rooms	X		X
Replacement cost	X		
Depreciation	X		
Rep. cost - depreciation	X		
Owner on record	X		
Ownership history	X	X	
Land line valuation	X		
Extra fixtures	X		

Outbuildings	X		X
Gross area	X		
Total land area		X	X
Rent estimate		X	
Mortgage estimate		X	
Attic		X	

Property information required by Homesite Insurance not provided by these databases

- Current policy information
- Ceiling height
- Wiring type
- Kitchen countertop material
- Rooms with crown moldings
- Rooms w/ cathedral/vaulted ceilings
- Proximity to fire station/fire hydrant
- Swimming pool
- Trampoline
- Wood/coal-burning stove
- Dog(s)
- Indoor sprinkler system
- Residents beyond immediate family
- Commercial/retail farming
- Trustee, Estate, LLC, or LLP on deed
- Business on the premises
- Land leased to third party
- Year of roof installation
- Burglar alarm
- Fire alarm

[1]Required for our replacement cost calculation
[2]Worcester Public Records

Appendix L: Means and Methods of Population

Location Information

Field Name	**Means of Derivation**	**Method of Entry**
Geographic coordinate	Can be acquired using the cell phone's built-in GPS unit	Not required
Street address	Can be acquired by querying Google's Geocoder service with the geographic coordinate provided by the GPS unit	Basic text field
Apartment, unit, floor	N/A (user only)	Basic text field
City/State	Derived directly from the Zip Code field	1: Basic text fields. 2: Dynamically-populated dropdowns.
Zip code	1: Can be acquired by querying Google's Geocoder service with the geographic coordinate provided by the GPS unit. 2: Can be acquired from location information provided by the smartphone's service provider	Basic text field

Policy Information

Field Name	**Means of Derivation**	**Method of Entry**
New policy start date	N/A (user only)	1: Calendar-based input control. 2: Formatted text field(s). 3: Numerical drop-downs.
A: Currently possess insurance? B (conditional): Date of expiration	Collaboration with other insurance companies	A1: Yes/No dropdown. A2: Checkbox. B1: Calendar-based input control. B2: Formatted text field(s). B3: Numerical drop-downs.

Identifying Information

Field Name	Means of Population	Method of Entry
First/Last name	May be stored in a smartphone's "personal contact info" utility	Basic text fields
Date of birth	N/A (user only)	1: Formatted text fields. 2: Numerical dropdowns.
Social Security Number	N/A (user only)	Basic text field
Email address	May be stored in a smartphone's "personal contact info" utility	Basic text field
Authorization to use consumer report information	N/A (user only)	Checkbox

Property Information

Field Name	Means of Derivation	Method of Entry
Year of Construction	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Living Area (sqft)	Real estate database/data vendor	Basic text field
Approx. Market Value	Real estate database/data vendor	Basic text field
Occupancy	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Home Style	Real estate database/data vendor	Dropdown with style choices
Number of Stories	1: Real estate database/data vendor. 2: Process picture of house exterior.	1: Basic text field. 2: Numerical dropdown.
Number of Separate Living Units	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Grade/slope (> 30°)	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Exterior wall material	1: Real estate database/data vendor. 2: Process picture of house exterior.	1: Dropdown with material choices. 2: Selection window with images of material choices.
Roof shape	1: Real estate database/data vendor. 2: Process (static) picture of house exterior. 3: Process (real-time) image capture of house using image processing and augmented reality.	1: Dropdown with shape choices. 2: Selection window with images of shape choices.
Roofing material	1: Real estate database/data vendor. 2: Process picture of house roof.	1: Dropdown with material choices. 2: Selection window with images of material choices.
Foundation shape	Real estate database/data vendor	1: Dropdown with shape choices. 2: Selection window with images of shape choices.

Type of foundation	Real estate database/data vendor	1: Dropdown with material choices. 2: Selection window with images of material choices.
State of basement	Real estate database/data vendor	Dropdown with state choices
Type of garage	Real estate database/data vendor	Dropdown with choices
Ceiling height	1: Real estate database/data vendor. 2: Process picture of house exterior.	1: Basic text field. 2: Numerical dropdown. 3: Dropdown with ranged choices.
Kitchen countertop material	1: Real estate database/data vendor. 2: Process picture of countertop.	1: Dropdown with material choices. 2: Selection window with images of material choices.
Central Air Conditioning	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Primary heating type	Real estate database/data vendor	Dropdown with heating choices
Wiring type	Real estate database/data vendor	Dropdown with choices
Inside wall material	Real estate database/data vendor	1: Dropdown with material choices. 2: Selection window with images of material choices.
Floor type(s)	1: Real estate database/data vendor. 2: Process image of floor.	1: Dropdown with material choices. 2: Selection window with images of material choices.
Full baths	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Half baths	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Rooms with crown moldings	1: Real estate database/data vendor. 2: Derive from house style.	1: Basic text field. 2: Numerical dropdown.
Rooms with cathedral or vaulted ceilings	1: Real estate database/data vendor. 2: Derive from house style.	1: Basic text field. 2: Numerical dropdown.
Fireplace(s)	Real estate database/data vendor	1: Yes/No or numerical dropdown 2: Basic text field
Within 5 miles of a fire station	1: Real estate database/data vendor. 2: Derive from state regulations. 3: Use Google Maps.	1: Yes/No dropdown 2: Checkbox
Within 1000ft of a fire hydrant	1: Real estate database/data vendor. 2: Derive from state regulations. 3: Use Google Maps.	1: Yes/No dropdown 2: Checkbox
Swimming pool	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Trampoline	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Wood/coal-burning stove	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox

Indoor sprinkler system	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Commercial/retail farming	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Trustee, Estate, LLC, or LLP on deed	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Business conducted on premises	1: Real estate database/data vendor. 2: Business registry.	1: Yes/No dropdown 2: Checkbox
Land leased to third party	Real estate database/data vendor	1: Yes/No dropdown 2: Checkbox
Year of roof installation	Real estate database/data vendor	1: Basic text field. 2: Numerical dropdown.
Burglar alarm	1: Real estate database/data vendor. 2: Security company registries.	Dropdown with alarm types
Fire alarm	1: Real estate database/data vendor. 2: Security company registries.	Dropdown with alarm types

Owner Information

Field Name	**Means of Derivation**	**Method of Entry**
Dog(s)	Canine registry	1: Yes/No dropdown 2: Checkbox
Residents beyond immediate family	Derived from house model/style and number of units	1: Yes/No dropdown 2: Checkbox
A: Include a secondary insured? B (conditional): Identifying information	N/A (user only)	A1: Yes/No dropdown A2: Checkbox B: Same as main Identifying Information section
Number of insurance cancellations due to non-payment	Collaboration with other insurance companies	1: Basic text field. 2: Numerical dropdown.
Intended property use in 3 months	N/A (user only)	Dropdown with use choices
Number of home insurance claims within last 5 years	Collaboration with other insurance companies	1: Basic text field. 2: Numerical dropdown.

Appendix M: Address-Lookup Test Results

*All tests performed in Worcester, MA.

Test info

Trial #	Actual Address	Latitude	Longitude	Altitude	Bearing
1	50 Francis St	42.309709	-71.309709	172	171
2	54 Francis St	42.309780	-71.794840	177	168
3	60 Francis St	42.309850	-71.794620	176	165
4	63 Francis St	42.309950	-71.794413	178	11
5	59 Francis St	42.309909	-71.794680	174	348
6	55 Francis St	42.990300	-71.794910	175	353
7	26 Ascadilla Rd	42.313880	-71.788040	197	159
8	25 Ascadilla Rd	42.313900	-71.788066	197	1
9	20 Ascadilla Rd	42.313870	-71.788250	197	187
10	19 Ascadilla Rd	42.313880	-71.788460	200	352
11	18 Ascadilla Rd	42.313820	-71.788350	198	155
12	7 Ascadilla Rd	42.313870	-71.788591	195	349
13	3 Westland St	42.270716	-71.817231	147	177
14	4 Westland St (duplex)	42.270760	-71.816992	144	89
15	5 Westland St	42.270905	-71.817061	131	242
16	10 Westland St (duplex)	42.270868	-71.816963	152	121
17	7 Westland St	42.271057	-71.817134	140	305
18	264 Highland St	42.270336	-71.817503	130	86
19	132 Russell St	42.269018	-71.814722	138	29
20	136 Russell St	42.262253	-71.814688	135	347
21	140 Russell St	42.269455	-71.814551	134	85
22	144 Russell St	42.269539	-71.814608	132	71
23	148 Russell St	42.269638	-71.814612	133	64
24	152 Russell St	42.269758	-71.814508	141	201
25	156 Russell St	42.269959	-71.814581	136	32

Test results

Trial #	Returned Address	Acc. Level	Actual Bearing	Accuracy %	° Accuracy
1	50 Francis St	5	175	98.889%	4
2	54 Francis St	5	175	98.056%	7
3	60 Francis St	5	175	97.222%	10
4	63 Francis St	5	355	95.556%	16
5	59 Francis St	5	355	98.056%	7
6	55 Francis St	5	355	99.444%	2
7	26 Ascadilla Rd	5	160	99.722%	1
8	26-28 Ascadilla Rd	10	340	94.167%	21
9	20 Ascadilla Rd	5	160	92.500%	27
10	18-20 Ascadilla Rd	10	340	96.667%	12

11	18 Ascadilla Rd	5	160	98.611%	5
12	7 Ascadilla Rd	5	340	97.500%	9
13	3 Westland St	5	279	71.667%	102
14	4,6 Westland St	5	99	97.222%	10
15	5 Westland St	5	279	89.722%	37
16	4,6 Westland St	5	99	93.889%	22
17	7 Westland St	5	279	92.778%	26
18	264 Highland St	5	355	25.278%	269
19	132-134 Russell St	10	99	80.547%	70
20	136 Russell St	5	99	68.889%	112
21	140 Russell St	5	99	96.111%	14
22	144 Russell St	5	99	92.222%	28
23	148 Russell St	5	99	90.278%	35
24	152 Russell St	5	99	71.667%	102
25	156 Russell St	5	99	81.311%	67

Test summary

Group	**Trials**	**Total**	**Description**
Correct	1-7, 9, 11-13, 15, 17, 18, 20-25	20	The correct address was returned.
Inaccurate	8, 10, 19	3	The correct street was returned but a range of house numbers was returned.
Invalid	14, 16	2	These were taken at duplexes, meaning that there were effectively two addresses for that house. These trials were discarded.

Appendix N: Calculated Replacement Cost Accuracy

Address	City of Worcester	Our Calculation	% Accuracy	% Error
3 ABINGTON	$142,785	$141,393	99.025%	0.975%
5 ABINGTON	$138,194	$143,099	96.451%	3.549%
28 BAILEY ST	$133,670	$125,853	94.152%	5.848%
90 BAILEY ST	$97,562	$98,694	98.840%	1.160%
7 CHESTER ST	$276,529	$282,517	97.835%	2.165%
61 CHESTER ST	$156,131	$149,361	95.664%	4.336%
27 DILLON ST	$136,151	$135,035	99.180%	0.820%
46 DILLON ST	$153,858	$132,221	85.937%	14.063%
4 ECHO ST	$155,838	$158,126	98.532%	1.468%
8 ECHO ST	$130,279	$123,114	94.500%	5.500%
383 BURNCOAT ST	$173,889	$175,484	99.083%	0.917%
50 FRANCIS ST	$175,701	$175,007	99.605%	0.395%
54 FRANCIS ST	$170,820	$178,664	95.408%	4.592%
318 MILL ST	$177,253	$171,687	96.860%	3.140%
323 MILL ST	$150,618	$150,529	99.941%	0.059%
6 VEGA	$176,045	$187,330	93.590%	6.410%
8 ZOAR	$181,003	$162,589	89.827%	10.173%
24 FALMOUTH	$128,936	$117,882	91.427%	8.573%
3 ALBEMARIE	$156,771	$138,017	88.037%	11.963%
12 LORNEZ	$160,815	$147,378	91.644%	8.356%
19 BANCROFT	$287,376	$266,201	92.632%	7.368%