

Code Camp Insurance app project report

Group 3

Mikko Turunen

Janne Salopaasi

1 IDEA

For this Code Camp LTC-Otso requested us to develop an application which would aid insurance salesmen to sell insurances. The problem statement had three use cases which had different priorities.

1.1 HOME INSURANCE SELLING

Create an app which asks for some basic information about the customers housing and then calculates a price for home insurance. After the price of the insurance has been calculated sales person should be able to create an insurance application with the app and deliver it into a web-service.

1.2 COLLECT SALES LEADS FROM A CUSTOMER

If the customer is in too much of a hurry for home insurance price calculation the sales person should be able to collect customers basic contact information and his insurance needs. After the information has been collected it should be delivered to a web-service.

1.3 CUSTOMER LOYALTY PROGRAM

If the customer already has an insurance with the associated company ask if the customer is interested in joining a customer loyalty program and fill the loyalty program application.

2 DESIGN AND GOALS

The project started with the creation of a prototype of our app. We discussed a bit about the use cases and we all agreed that the app should be simple and fast to use. Then we proceeded to make some drafts about the apps visual appearance on paper and lastly we created an interactive prototype with Marvel prototyping tool. Some goals, and priorities for these goals, were also set and initially we were going to create solutions for all of the three use cases but eventually we decided to focus on the home insurance use case and thus the two other use cases were dropped.

2.1 CREATING A GUI

The highest priority goal was to create a GUI with C# and XAML and we were able to create a crude GUI in a matter of minutes. After creating a GUI and testing how to add some functionality into it we split our effort into GUI building and core functionality programming.

2.2 CONNECTING WITH REST

Connecting with the provided RESTful server was considered to be a high priority goal as the insurance price calculation was done on the server. We managed to successfully use the RESTful server in the second day of the Code Camp and some of our REST code was shared during the Code Camp as an example material.

2.3 CREATING INTERACTIVE FUNCTIONALITY WITH GUI

After we were able to send data to the server and receive data from it we began integrating the REST features into the GUI. This phase was quite straight forward and was finished on the noon of the third day.

2.4 CREATING A RESOLUTION INDEPENDENT GUI

This was the most time consuming part of our project and after struggling with the GUI for rest of the third day we decided to drop two of the use cases and focus on the GUI. Possibly the biggest factor in the issues with GUI was that it was hard to find relevant XAML examples as most of the search results related to WPF or in some cases the examples just did not work. Eventually we managed to

create a resolution independent GUI with a single limitation: 1024x768 resolution in landscape mode breaks the layout. Apart from the aforementioned peculiar resolution all of the simulators resolutions work fine with all four orientations and we are quite happy with our final product.

3 FEATURES

Our product is simple and fast to use. The app is lightweight and it uses a scrolling single page with snap points for a fluid user experience. The app utilizes most of the provided RESTful servers JSON messages: POST:price calculation & storing application; GET: single application with ID; DELETE: single application with ID. The GUI is resolution and orientation independent but it is locked into portrait mode as we felt that portrait mode was better for the user experience. Lastly all of the local data can be reset with a single button press.

The screenshot displays a mobile application interface for insurance calculation. The background is a dark purple color. At the top, there are three logos for 'Code Camp' and three 'Reset' buttons. The main form contains the following elements:

- Address:** A text input field containing 'Esimerkkikatu 1'.
- Postal Code:** A text input field containing '12345'.
- Floor Area:** A text input field containing '123.4'.
- Build Year:** A dropdown menu showing '2015'.
- Property Type:** A dropdown menu with options: House (selected), Apartment, Rowhouse, Pairhouse, Summerhouse, and Sauna.
- Insurance Start:** A date picker showing '05' and '28'.
- Insurance End Date:** A date picker showing 'May', '28', and '2022'.
- Billing period:** A dropdown menu showing 'Yearly'.
- Currency:** A dropdown menu showing '€ (Euro)'.
- Calculated price:** A central display showing '321.45€'.
- Buttons:** 'Order Insurance', 'Confirm', 'Delete application', and 'Calculate Insurance Price'.
- Confirmation:** A 'Confirmed' message at the bottom.

4 THOUGHTS ABOUT THE PROJECT

We started out as a three man group but the third member of the group dropped out after the first day so the project was essentially a two man effort. Both of us had no experience with C# or XAML but we found out that C# is quite easy to pick up with Java programming experience. We also noticed that Blend, the graphical XAML GUI editor by Microsoft, was easy to use and producing prototypes with it was fast and straightforward. On the flip side of the coin we felt that Blend was not that useful for creating resolution independent GUI as we ended up editing XAML in the code editor more than using the graphical user interface. One thing we realized on the third day of the course was that some kind of a version control system is mandatory for a project of this kind. After the third day only one of us could program at a time as we had no rational way to merge our commits without a VCS and this probably slowed us down quite a bit. All in all we think that we learned quite a bit about programming and project management during this project.

5 THOUGHTS ABOUT THE COURSE

We were both new to Coding Camps and we think that it is a pleasantly different form of a course when compared to traditional courses. The course was well organized and there were no significant delays on the time table and there were no show stopping issues which might drive people out of the course. Our group caused probably the biggest delay during the final presentations and we learned our lesson from it: Test your stuff beforehand. The course was definitely a positive experience and we are looking forward for future Code Camps.