CT30A9301

Code Camp on Platform Based Application Development

SureApp – An Interactive Insurance selling Application

Group 5
Isto Sipilä
Yongyi Wu
Fahad Rezwanul Islam
Markku Painomaa

1. Introduction:

1.1 Project Background:

In the Otso Case CodeCamp project the main idea was to build an Insurance Selling application that will run on Windows 8.1 mobile device. The business requirement was to enable sales force to sell and market insurance product to customer in location independent place, and make the selling process more interactive and introduce higher customer involvement. Therefore, with a quick but through brainstorming our team came with an application idea, that not only will make selling process fast but also make the selling fun.

SureApp is a mobile insurance sales tool, primarily being designed to help insurance sales force to have interactive one-to-one product selling and marketing.The App is developed on Windows 8.1 platform, using C# and XAML. The basic idea is to fast selling of insurance product and increase customer involvement. That suggest the insurance selling process to be more fun and interesting for the both sales and customer end.

1.2 Motivation:

One of the basic challenges is to sell customer rather complicated and virtual product in an open and outdoor environment. Expressing the minimum information to attract a customer is also a key to achieve sales success. SureApp is designed to eliminate those challenges by understanding customer's perception and desire. Hence, the most important idea of an offer will be presented by obtaining minimum amount of information input in the system. However, interested customers will also get on spot application possibility.

Another of our motivation was to get a better grip on C# programming together with interface designing with XAML. That is a very interesting blend of coding for Windows programming, moreover this challenging idea of the codecamp was the right place to practice such opportunity, where it was a real life problem solving scenario.

2. The Project Development:

2.1 Use Case of the project scenario:

Here is the basic use case diagram of our application, this represents the seller-customer interaction during the application usage phase. It begins with the salesperson-customer interaction before the sales starts by the salesperson asking the customer about the preferred

option which includes a quick approach - where customer will provide the salesman with inputs to reserve an appointment where the salesperson will communicate with the customer about the insurance. Detailed approach - includes a seller-customer interaction where the salesperson will offer the customer a possibility to gain a discount by winning in a game, however that is optional. Then the customer will provide the salesperson with information about his object to get estimation. After that if he still wants to proceed for application the salesperson will collect further inputs for the insurance application.
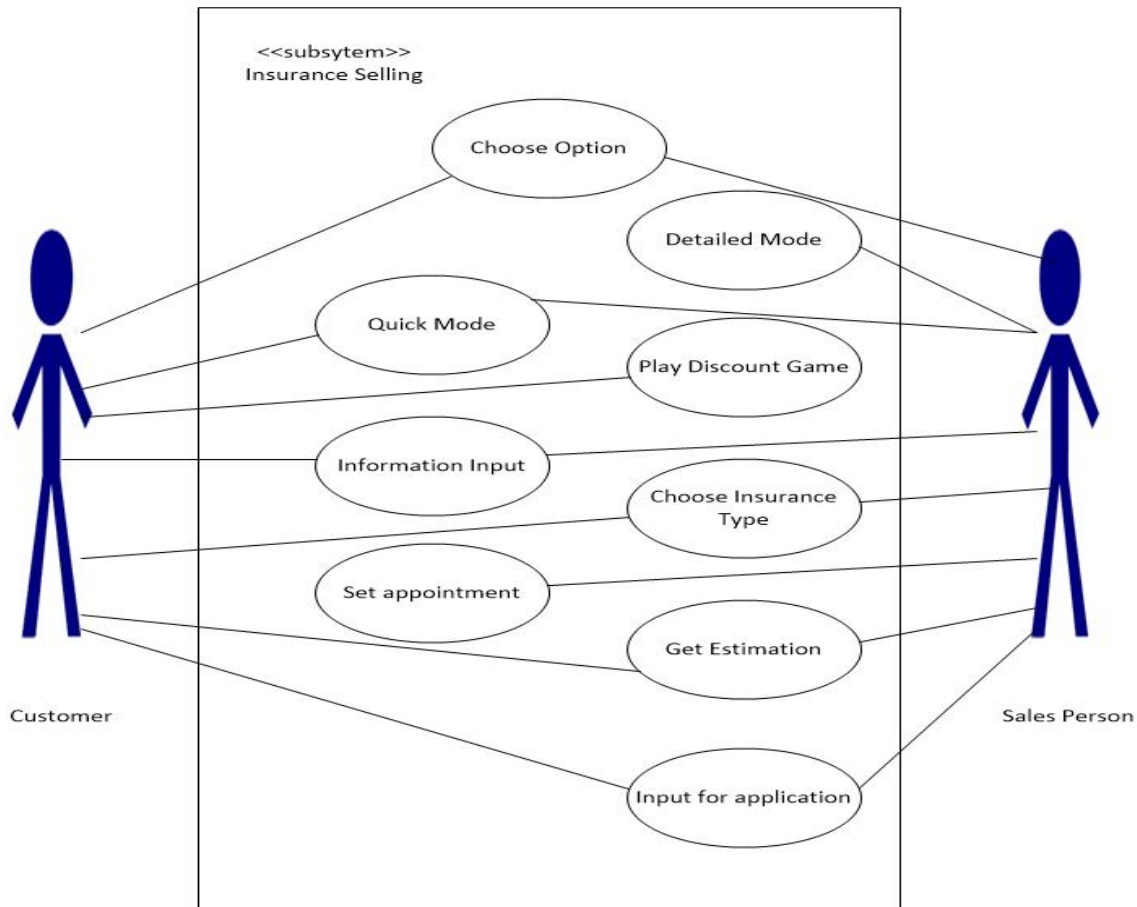

Figure: Use-Case Diagram for insurance selling

2.2 Development Platform and Language:

The development of the SureApp has been done on Windows 8.1. Programming languages that has been used are:
- C# - it's simple, powerful, type-safe, and object-oriented. Visual C# is modern, high-level, multi-paradigm, general-purpose programming language.

- XAML - Extensible Application Markup Language, pronounced "zammel", is an XML-based markup language developed by Microsoft. XAML is the language behind the visual presentation of a Web page.

The tools that has been used to develop the application are Visual Studio 2013 and Visual Studio Blend for perfection of the user interface.

2.3 Features Included:

The features included in the application are:
- Well scaled buttons for doubt free touch input
- Confirmation of page cancellation for avoiding data loss
- Browse through the input pages to update or modify inputs without data loss
- A scratch card game for discount
- Simple but well designed pages with elegant look

2.4. SureApp application Quality:

- Usability: The primary goal in our application was ensure it has usability, this is ensured in both seller's and customer's perspective. It was maintained by easy to understand picture buttons, explanatory pop-up modules and simple procedures.
- Trust: Customers are directly dealt with the seller, no critical information (e.g. banking, credit card) is needed to get the insurance.
- Portability: The application is designed for Windows 8.1 mobile platform, hence, it is portable for any location. Data can be transferred to the server with any kind of wireless technology available in the device.
- Understandability: The coding is done in a way that it is pretty easy for other to get to understand the modules and the objects. The naming was made with match with actual functionality in the front-end what extends understandability.
- Reliability: Loop indexes range-tested. Input data checked for range and type errors. Exception handling provided. Hence, we can confirm the reliability.

2.5. Challenges:

Platform Challenge: Windows 8.1 has several changes in namespacing when coming from older versions of Windows, so, conversion was a challenge when work was done in personal computers.

Language challenge: Few of the team members had very little working experience in C# and XAML, that was excellent learning opportunity while solving problems.

Coding Challenge: During developing the game module there were several problems, as random number generation from same array which would match the limited loop criteria and to obtain the number generated from the module. However, the problem was finally resolved. Challenges were also there while providing server with inputs that the server was rejecting, these problems were also finally resolved.

Design Challenge: Icons, buttons and background needed a lot of time to have similarity, well visuality and authenticity. Many of the preferred icons were not for free what made us the make our own.

4. Implementation and Deployment:

SureApp is implemented in several *Page*s and user moves between them by tapping menu buttons. The XAML page flow is in the figure below.

The core of the SureApp is inside the *HomeInsuranceData.cs* which has datacontract classes *HomeInsuranceData, HomeInsurancePriceData*, *HomeInsuranceResponseData*, *DataSummary* and *ApplicationId.* They correspond to the server's database objects. They're used to send and receive JSON data from server and also for transferring data between application pages.
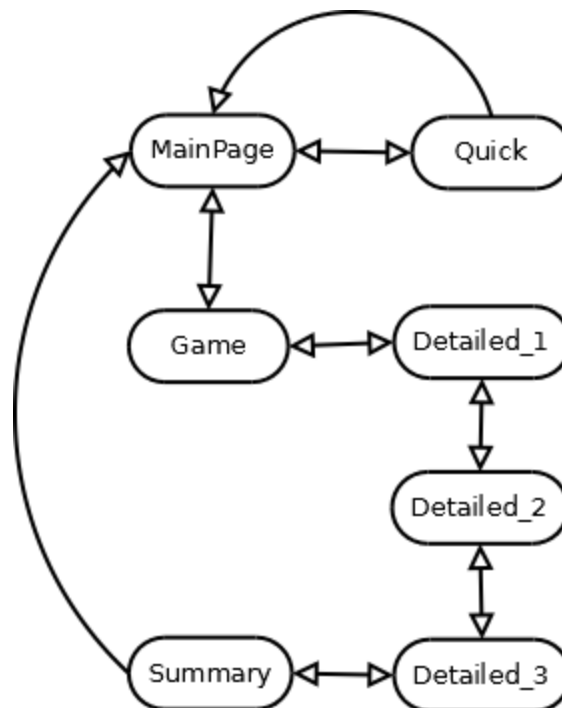

Figure. SureApp xaml-page flow.

Every Page has return or cancel button which always allows returning to previous page or cancelling the whole process. Most code is basic Windows App navigation and input handling. The only difference is the server connection code on *Detailed_2* and *Summary* pages. App starts at *MainPage* where user can go to *Quick* or *Detailed_1* pages. Exceptionally there's a *Game* page before *Detailed_1* which can be played or skipped. If played the *Game* page randomly picks you an insurance fee discount value that is in static variable *Game.discount*. Server connection is used first time in *Detailed_2* when user fills in all the needed input and *HomeInsuranceData* packet is sent to server and *HomeInsuranceReponseData* received. When moving on to *Detailed_3* the *HomeInsuranceReponseData* data packet follows along using Frame.Navigate(*Detailed_3*, *HomeInsuranceReponseData* data) and OnNavigatedTo(*Parameter* e). In *Detailed_3* the user fills the rest of the needed info and continues to *Summary* page. If all info is correct it is packed into *DataSummary* object and another server connection is established: *DataSummary* packet is sent and *ApplicationId* packet is received. Finally the app returns to *MainPage*.

5. Conclusion:

Since from the beginning of the CodeCamp there was a proper plan for our development. Immediately after we got the briefing we had a team brainstorming session, we were already aware of the technologies but we had to plan for the uniqueness of our application. At first we planned and developed the primary front-ends, the pages and required input options. Then we divided the modules to each of our members. The modules included, Design perfection, server connectivity, game, navigation, required pop-ups and synchronization.

There were several challenges such as Windows version conversion as few of us was developing with different version of Windows it was needed to have synchronization for proper output in Windows 8.1. Although there were challenges in both technology and learning part, it was a comforting end result after all. The application ran smoothly and all the modules functionality was maintained. It was a good learning opportunity, in perspective of real life application development.

Further area of development and enhancement in the application might include other insurance options based on the similar principle of House Insurance that is already incorporated in our application and customer loyalty program that we did not include for our application which was an optional functionality.