Clients

Dedicated

Possibilities- winforms, wpf, xaml

“Windows Forms is a smart client technology for the .NET Framework, a set of managed libraries that simplify common application tasks such as reading and writing to the file system.”1 It is a tool used to create desktop clients. WinForms provides many useful controls. Controls can display textboxes, buttons, list boxes, menu bars, even web pages. Its design is very basic, nothing special on its own. It’s very user friendly and easy to learn. One can use drag and drop method to move controls in place on the panel. Double clicking on control will create skeleton for method that returns result of the action.

Windows Presentation Foundation (WPF) is another tool for creating desktop clients. “The core of WPF is a resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware. WPF extends the core with a comprehensive set of application-development features that include Extensible Application Markup Language (XAML), controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography.”2 Its design is more modern that WinForms’- corners more round and some coloring and shading works.

Choice

Our choice was WinForms. We chose to go with it for many reasons. First, for us, design is nearly not as important as functionality. Second, we did not want to spend time on a spike about new ways to create dedicated client. Third, it was made pretty clear that majority of the points which count towards project’s grade come from the backend of the application. Finally, we were already familiar with WinForms. For us, all these were quite strong arguments towards going with WinForms. Since design is not as important, counting towards the grade for this project, as functionality and we already knew how to create clients in WinForms, we decided not to do a spike on any other options. As a result, we selected WinForms for creating our dedicated client.

Web

Possibilities – mvc, web forms, web pages

Web Forms is part of ASP.NET web application framework. ASP.NET supports 4 programming models, the others are ASP.NET MVC, ASP.NET Web Pages and ASP.NET Single Page Applications. Web forms are regular web pages that user can request through any browser. Pages are written using HTML, server controls, client-script and server code. In visual studio one can use drag and drop method to place controls onto the web page and design the lay out. Overall it is very similar to WinForms, which is used to create dedicated clients.

MVC (Model-View-Controller) is a lightweight framework that separates an application into 3 components, all built to handle specific development aspect in the application. Model deals with all the data-related logic, like representing data, that is moved between View and Controller. View component deals with UI all the UI logic, which includes all textboxes, buttons and other controls. Controller’s job is to handle all the business logic and incoming requests. It has to manipulate data by using the Model component and communicate with View to show the outputs. MVC supports all ASP.NET functionalities, like authentication, master pages, ASP.NET routing, etc. MVC is frequently used in web development framework for creating extensible and scalable projects.

Choice - mvc

Our team chose ASP.NET MVC. We had 4 options to choose from- ASP.NET Web Forms, Web Pages, Single Page Application or MVC. At first we eliminated Single page Application and Web pages. Argument against Sing Page Application was that our application needs definitely more than 1 web page. Arguments against Web Pages were that we have never worked with web pages before and we did not want to do spike on it either. We decided to go with Web Forms, at first. It was easy to implement, userfriendly and works identically to WinForms, which we used to create desktop application earlier. Then we were introduced to ASP.NET MVC framework in last programming sessions. We had information that it is way more frequently used in actual businesses that web forms and that got us thinking. We argued over whether to go easier way and implement Web Forms, that we were already familiar with and did not have to do a spike on, or go with widely used MVC, that required us to spend time on learning about it. We decided that since we are studying for future jobs, using outdated framework really does not make sence. In the end MVC came out on top.

1 https://docs.microsoft.com/en-us/dotnet/framework/winforms/windows-forms-overview

2 https://docs.microsoft.com/en-us/visualstudio/designers/introduction-to-wpf