Glimpse Iteration 5

Iteration 5 has been fairly well in terms of completion of our stories. We have several issues regarding database configuration and with our models that delayed us a bit. However, most of work was completed, 14 out of 16 user stories were completed. The demo is going to be sent in few days from now, because we told ourselves that it would be more presentable with added content. We decided send it after Sprint 5 is completed.

Team members

Name and Student id	GitHub id	Number of story points that member was an **author** on.
Sam Kazerouni 26658415	sammykaz	21
Omer Deljanin 27064438	bosniak47	20
Jonathan Phillips 26699596	PhillipsJP	5
Eric Leon-Rodas 26648754	RiceAndBeanz	26
Joseph Daaboul 27077890	jdaaboul	21
Asma Laaribi 29326197	asmalaaribi	15

Project summary (max one paragraph)

Glimpse is a location based application that aims to provide the most up to date information related to local promotions and sales to its buyers. Local businesses often have small scale promotions or one day deals that are communicated to clients in person as they enter the store. This app will give these businesses a platform to advertize such sales to a wider audience. It will help with giving them exposure to potential clients who are passing by. The main objective of the application is to efficiently advertise promotions that might have otherwise gone unnoticed to likely clients. Buyers will be able to see the sales that are taking place in the stores or businesses close by. It is providing them with relevant information based on their current location. Glimpse will offer its users a map view of the available promotions as well as a mosaic view where these promotions will be displayed as tiles. One of the main features of the application is the follow option which allows users to be more selective and follow their favorite businesses and see their promotions on the map. The users will also be able to receive notifications from the vendors they are following as soon as there is a new promotion. Another great feature is the live chat option which provides a communication channel between the users and the business. In terms of tracking how much exposure a promotion has received, the number of users that have seen it will be displayed and visible for the business it originated from. We envision this mobile application to be an innovative marketing tool that is able to keep up with daily changes and challenges of a local commercial market.

Velocity

Iterations	Completed User Stories	USP	Velocity	Comments
1	0	0	0	Initial Configuration
2	0	27	0	Configuration Issues & No Unit Tests were Completed, Stories moved to next Iteration
3	8	59	59	
4	5	40	40	
5	13	80	80	
6	N/A	N/A	N/A	

Average Velocity = 45 (Not taking into account Iteration 1)

Plan up to next release

ID	Name	Priority	USP
Iteration 5 96 Stories - 4 weeks (USP)			

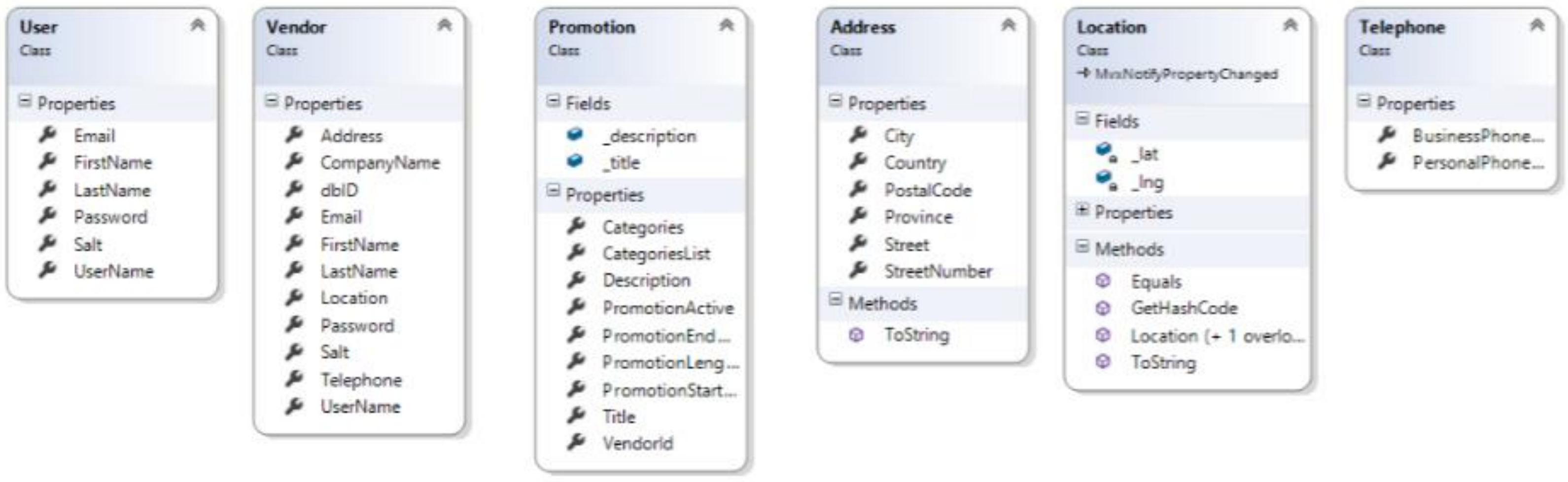
18	View detailed content from a selected promotion pin.	High	13
1	Vendor: View pictures on the promotion details view.	Medium	8
36	Handle amount of promotion pin to display when zooming in & out	Medium	13
95	Store users within radius of promotion in database	Medium	8, Moved to Next Sprint
41	View Tiles Page	Medium	8
99	Add Vendor promotions in the tiles view	Medium	13
84	Web: View Login Page	Medium	3
85	Web : Login Vendor	Medium	3
93	Web : Sign up (Vendor only)	Medium	3
86	Web : Navigation bar (navigate to other pages)	Medium	3
91	Web : View Profile page	Medium	3
89	Web : Create a Promotion (using categories, image, description...)	High	3
92	Web : Customize image when creating promotion (crop, color saturation, add text layers)	Medium	13
87	Web : View Map	Medium	2
88	Web : Ability to view all promotions on the map when logged in as a vendor	Low	8, Moved to Next Sprint
33	Import Map layout Json	Low	5

Total			80
Release 2 96 Stories - 2 weeks (USP)			
95	Store users within radius of promotion in database	High	21
125	Display Tiles in the TilesView	Low	13
20	Filter promotions by category	Medium	13
5	Follow/Unfollow vendor	Medium	5
6	Receive notifications from followed vendors	Medium	5
8	Pictures of a tile appear in a slideshow manner	Medium	8
118	Web : Add "Expired" label to expired promotions in the vendors my promotions	Medium	5
117	Web : Re-post an expired promotion	Medium	5
94	Web : Generate graph (based on number of users in radius of geolocation)	High (Depends on completion of #95)	13
88	Web : Ability to view all promotions on the map when logged in as a vendor	Low	8
90	Web : Add newly created promotion on the map	Low	5
Total			101

Overall Arch and Class diagram

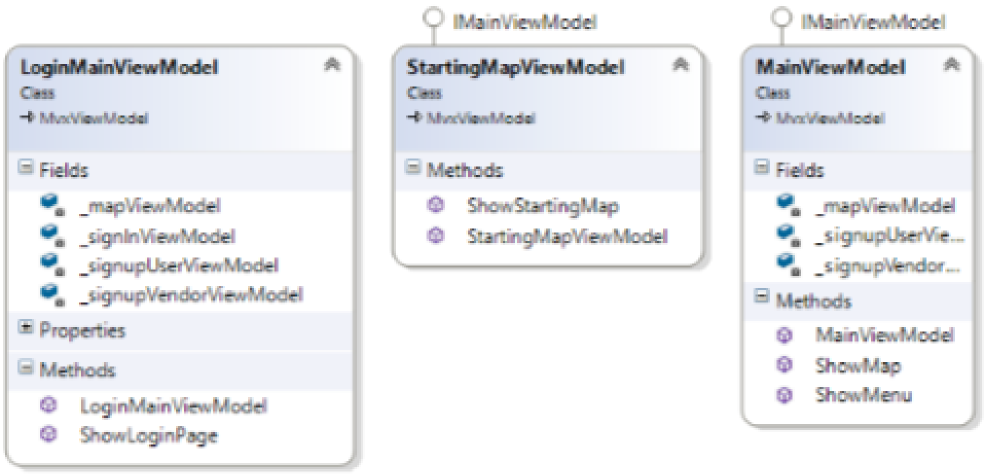
The Models package contains all the models required which are mapped to the ones used in the database. These are the current models for our application and will be updated in the future.

Models Package:



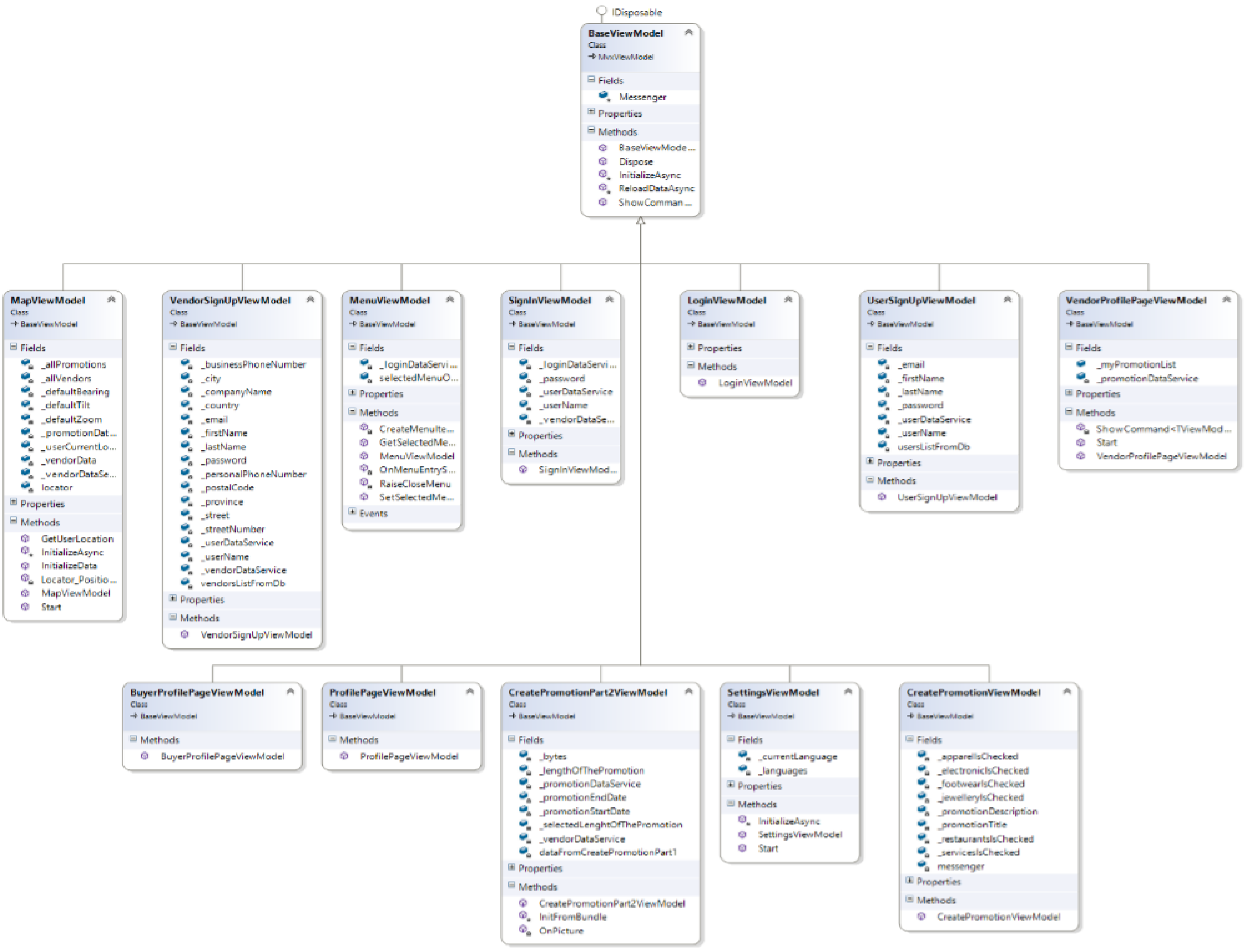
This package contains the view models for each activity in our application, currently we have LoginActivity, MainActivity and StartingMapActivity. These view models are used to update the views of these activities and contain logic related to user interface.

Activities View Models Package:



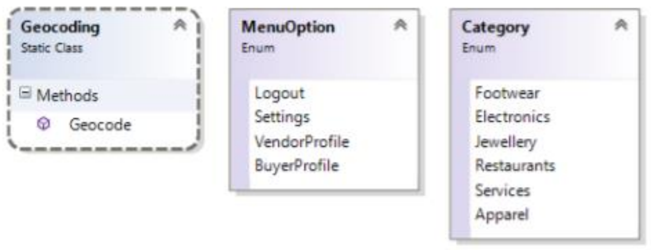
This is another View Models package, it updates the views, contains some business logic related to views, however these each fragment in our application has an associated view model which takes care of the fragments view.

Fragments View Models Package:



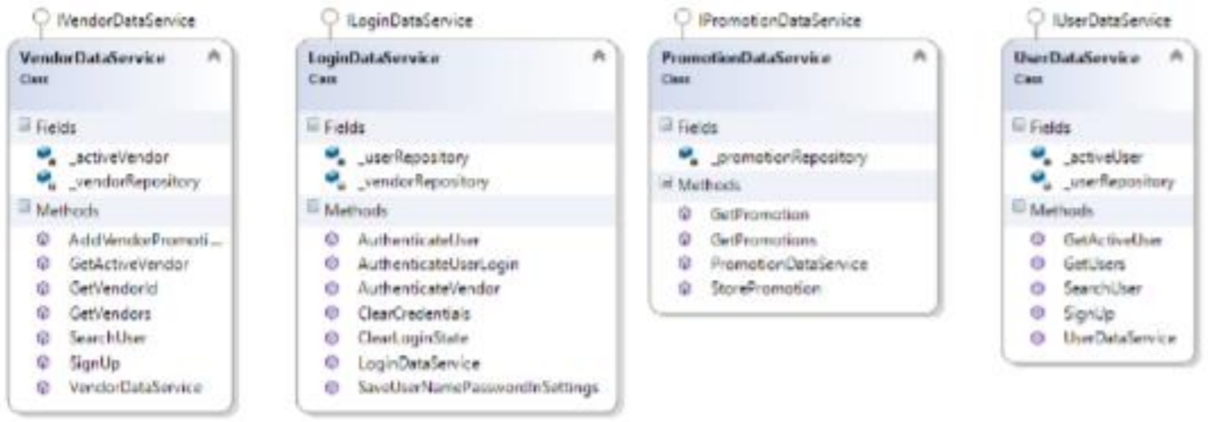
The Utility package contains all utility classes and enums of the application. These utility classes are converters, tools, etc…

Utility Package



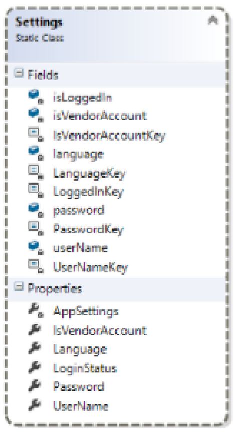
The Service packages are the most important packages in our application because they contain most of business logic related to the core of our application. They have IRepository types in their constructor using dependency injection, which retrieves a repository type and can interface to the database easily. There is no instantiation of these types and the singleton pattern is used for each repository type. This is all maintained in the framework. The Data services contain business logic related to data.

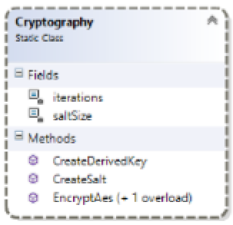
Data Service Package:



This is another service package which is very important. The General Service package contains core business logic and doesn't contain business logic related to data.

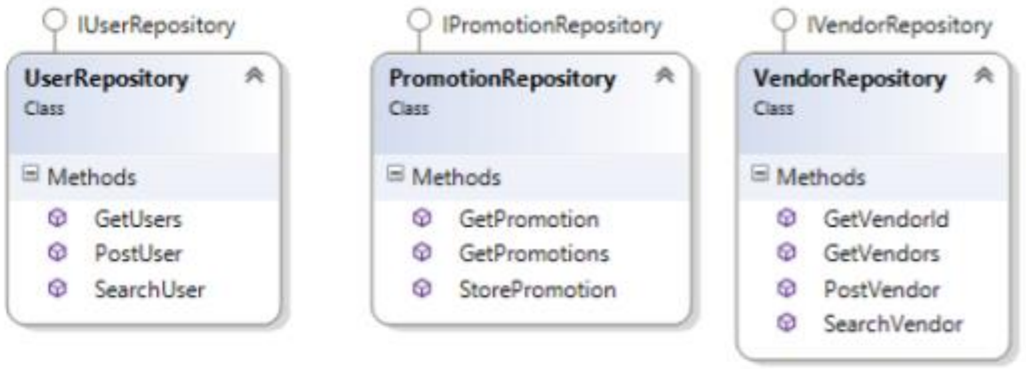
General Service Package:

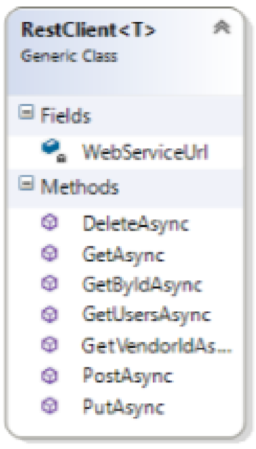




Repository Package:

The repository classes are the commands to make calls to the database and use the RestClient class shown below. These are classes are the gateway to the database, they use the RestClient to send HTTP requests to the webservice which then sends a POST or GET to database. The response from webservice is then returned to the Restclient and passed onto the Repository.

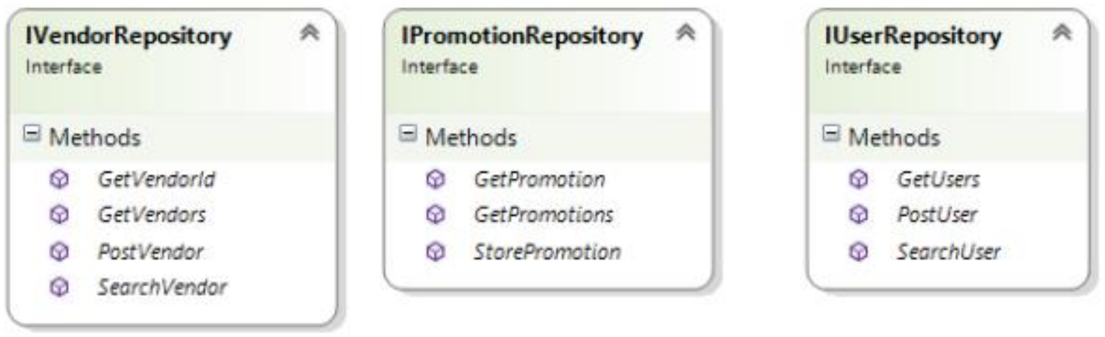




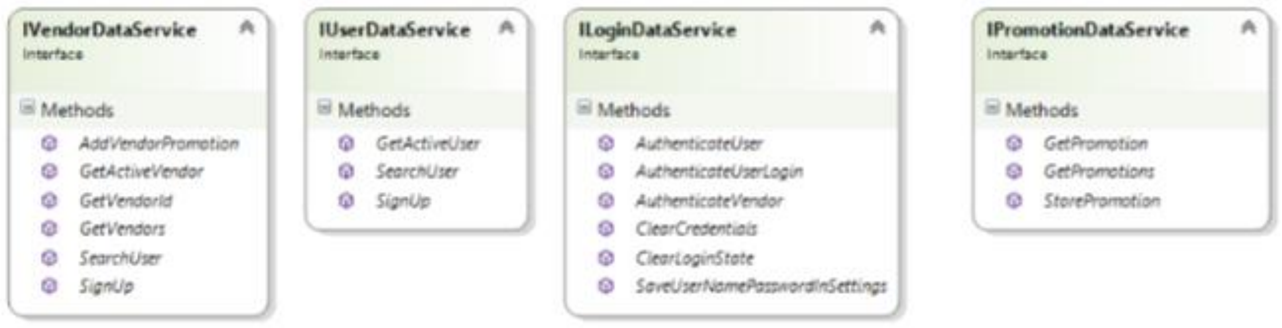
Contracts

These are the contracts packages, the repository and service contracts contain several methods which are used in our repositories and services respectively. We define them here before we use them in our application.

Repository Contracts Package:



Service Contracts Package:



Infrastructure

MvvmCross Framework (Application framework)

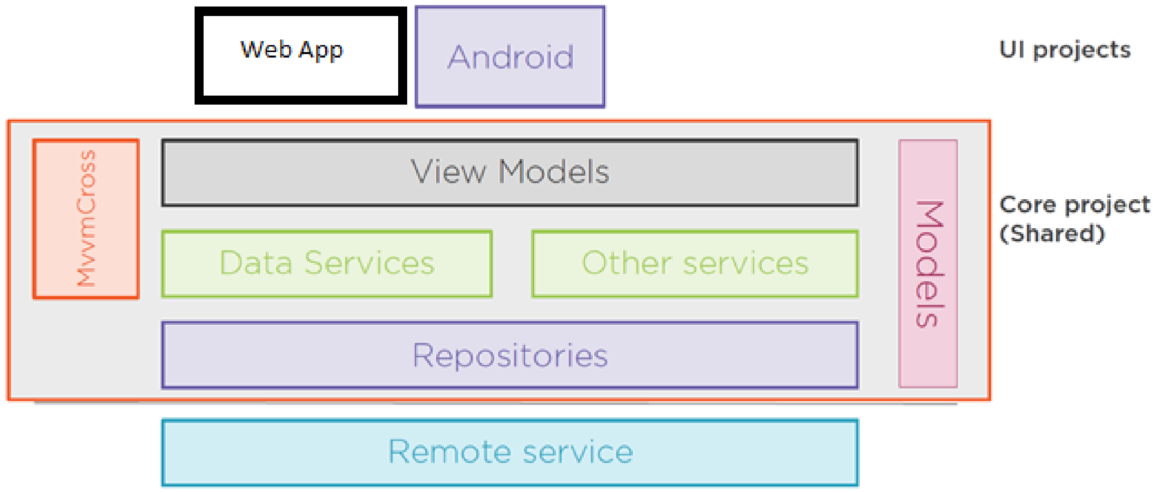
https://mvvmcross.com/

MvvmCross is a cross-platform mvvm framework that allows developers to create cross platform apps. Model-View-ViewModel is an software architectural pattern. MVVM is a variation of Martin Fowler's Presentation Model design pattern. Like Fowler's Presentation Model, MVVM abstracts a view's state and behavior. However, whereas the Presentation Model abstracts a view (i.e., creates a view model) in a manner not dependent on a specific user-interface platform.Several reasons why we are using

MVVM and Presentation Model both derive from the model–view–controller pattern (MVC). MVVM facilitates a separation of development of the graphical user interface (either as markup language or GUI code) from development of the business logic or back-end logic (the data model). The view model of MVVM is a value converter meaning the ViewModel is responsible for exposing (converting) the data objects from the model in such a way objects are easily managed and consumed. In this respect, the view model is more model than view, and handles most if not all of the view's display logic The view model may implement a mediator pattern, organizing access to the back-end logic around the set of use cases supported by the view.

We took advantage of all the benefits that are provided by MVVMCross, such as bindings, converters, IoC, etc. For the Portable Class Libraries, the other option to sharing code is file-linking. PCL is neat, tidy, and requires no overhead when your common code library changes. File-linking is a manual process and can begin to get cumbersome after constant project changes. Furthermore, extensibility of the application can be easily done using the concept of plugins. Plugins can be added to the framework which can make our application grow much faster.

The disadvantage of using MvvmCross framework is that there a learning curve.



Microsoft Azure Database
https://azure.microsoft.com/en-us/services/sql-database/

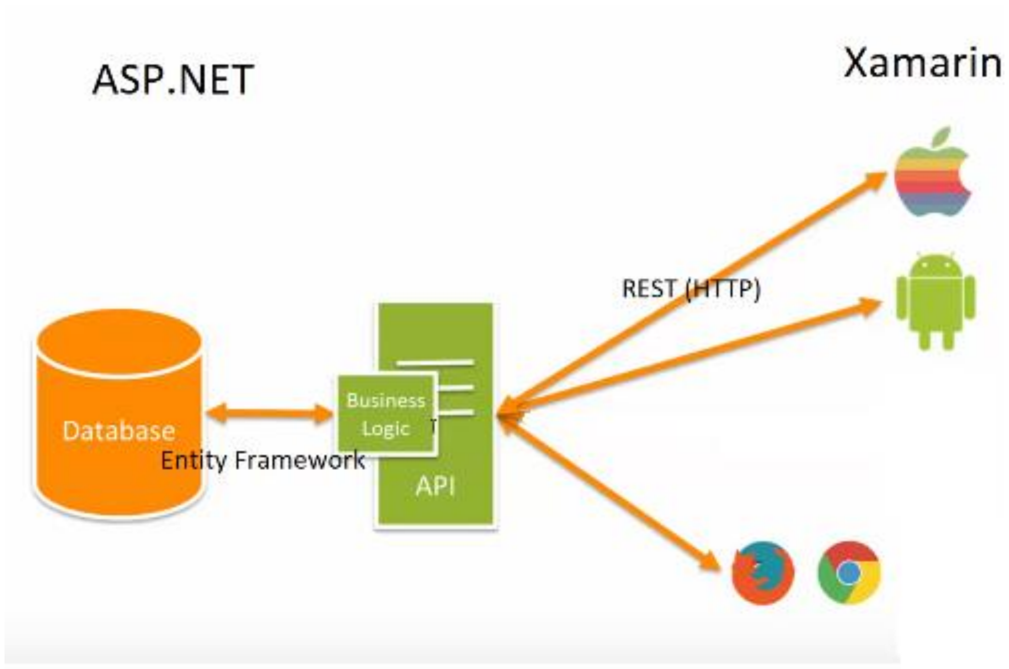
We have selected Microsoft Azure Database because our project is based on .NET and we're using Visual Studio, therefore it makes sense using Microsoft Azure. The Azure plugin allows us to easily connect to our remote database using the Server Explorer in Visual Studio which allows us to quickly view data. Our remote server is located around

Central Canada region which is relatively close to our location and runs our SQL database.

When comparing Amazon and Microsoft's database services, cost does come into play. Microsoft gives students a free 1 year subscription to use their services through the Microsoft Imagine program. Amazon also gives students a 150$ credit which was used. However, initially when we used Amazon Web Services, the configuration was slightly more complex and we realised that there are several hidden charges when using amazon servers and databases. These charges increased quickly and cost per usage time. This was terrible because there had to be further research on understanding what these charges ment and how to prevent them. Therefore, we decided to transition to Microsoft Azure and we have had no issues related to configuration, cost and service.

Webservices

In order to make our database transactions, we have made a WebServices project which communicates to our remote SQL database. It is the only gateway to our database and our application must use our WebServices in order to make requests. Our WebServices project uses Entity Framework for its Models and maps to the tables in the database. Our Core application which contains our shared business logic, uses a layer called "Repositories" where our RestClient is used. The RestClient sends CRUD operations using HTTP to our WebServices which then sends a request to our database. Moreover, a response is returned from our database to our WebService and back to the RestClient.



Name Conventions

Standard C# Coding Convention: https://msdn.microsoft.com/en-us/library/ff926074.aspx

Code

File path with clickable GitHub link	Purpose (1 line description)
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core/RestClient/RestClient.cs	Generic class to be used by repositories that implements CRUD operations.
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core/ViewModel/MainViewModel.cs	Initializes all ViewModels for our application
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core/ViewModel/VendorSignUpViewModel.cs	Uses the command design pattern to trigger the sign up of a vendor
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core/App.cs	Registers files ending with “Repository” and “Service” to IOC container, allows them to be used through interface and will only use a single instance.
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core/Services/General/Cryptography.cs	Encryption class for password

Testing and Continuous Integration

List the **5** most important test with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core.UnitTests/Tests/Repository/UserRepositoryTests.cs	When a user signs up, their credentials are stored on the database.
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core.UnitTests/Tests/Repository/VendorRepositoryTests.cs	When a vendor signs up, their credentials are stored on the database.
https://github.com/sammykaz/glimpse/blob/master/Glimpse.Core.UnitTests/Tests/Services/CryptographyTests.cs	The same password encrypted twice is not equal.
Other tests include System Tests for UI which are shown in Github.	