Glimpse Release Summary

Our focus for the first release consisted on several areas that we wanted to explore starting with the high risk high value. The vendor verification process was one of the high risk user stories in iteration 3 because we weren't sure on how to verify a vendor who signs up. Several solutions were proposed for this user story and we concluded that we could not automate this process so we proceeded with manual verification solution. Other than the high risk, high value user stories, we wanted to get the team comfortable working in C# with Xamarin and the MvvmCross framework. Also, we started creating several views in order to get familiar with data binding . Furthermore, we also set up a remote database which is communicated through a web service.

At this point, many of our configuration issues related to Xamarin and Portable Projects that we had in the first and second iteration were resolved. For example, we had issues where some portable class library references were not supported for our Profile in Xamarin. Therefore, we had to change our Profile and research in order to verify what references we want to support. We realised that selecting a Profile determines which portable class library references (plugins) we can use. These references are plugins that are added to our project which allows us to easily add features.

Team members

Name and Student id	GitHub id	Number of story points that member was an **author** on.
Sam Kazerouni 26658415	sammykaz	29
Omer Deljanin 27064438	bosniak47	8
Jonathan Phillips 26699596	PhillipsJP	5
Eric Leon-Rodas 26648754	RiceAndBeanz	11
Joseph Daaboul 27077890	jdaaboul	26
Asma Laaribi 29326197	asmalaaribi	11

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	56	56	

Average Velocity = 19

Plan up to next release

ID	Name	Priority	USP
Iteration 4			

4 Stories - 2 weeks (42 USP)			
#32	As a user, I want to able to see myself on the map	High	13
#61	As a user, I want to be able to create a promotion for my store	High	8
#33	As a user, I want to have a specific layout and styling for the map	Medium	13
#62	As a user, I want to login to my account	Medium	8
Total			42
Iteration 5 X Stories - 2 weeks (USP)			
#36	As a user, I want to be able to zoom in & zoom out of the map	Medium	13
#20	As a user, I want to be able to filter promotions by category in my settings.	Medium	13
#40	As a user, I want to able to view settings so that I can modify parameters in the application.	Medium	3
#64	As a user, I want to able to view promotion pins on the map.	Medium	5
Total			34
Release 2 X Stories - 2 weeks (USP)			
UI		High	34
Total			34

Overall Arch and Class diagram

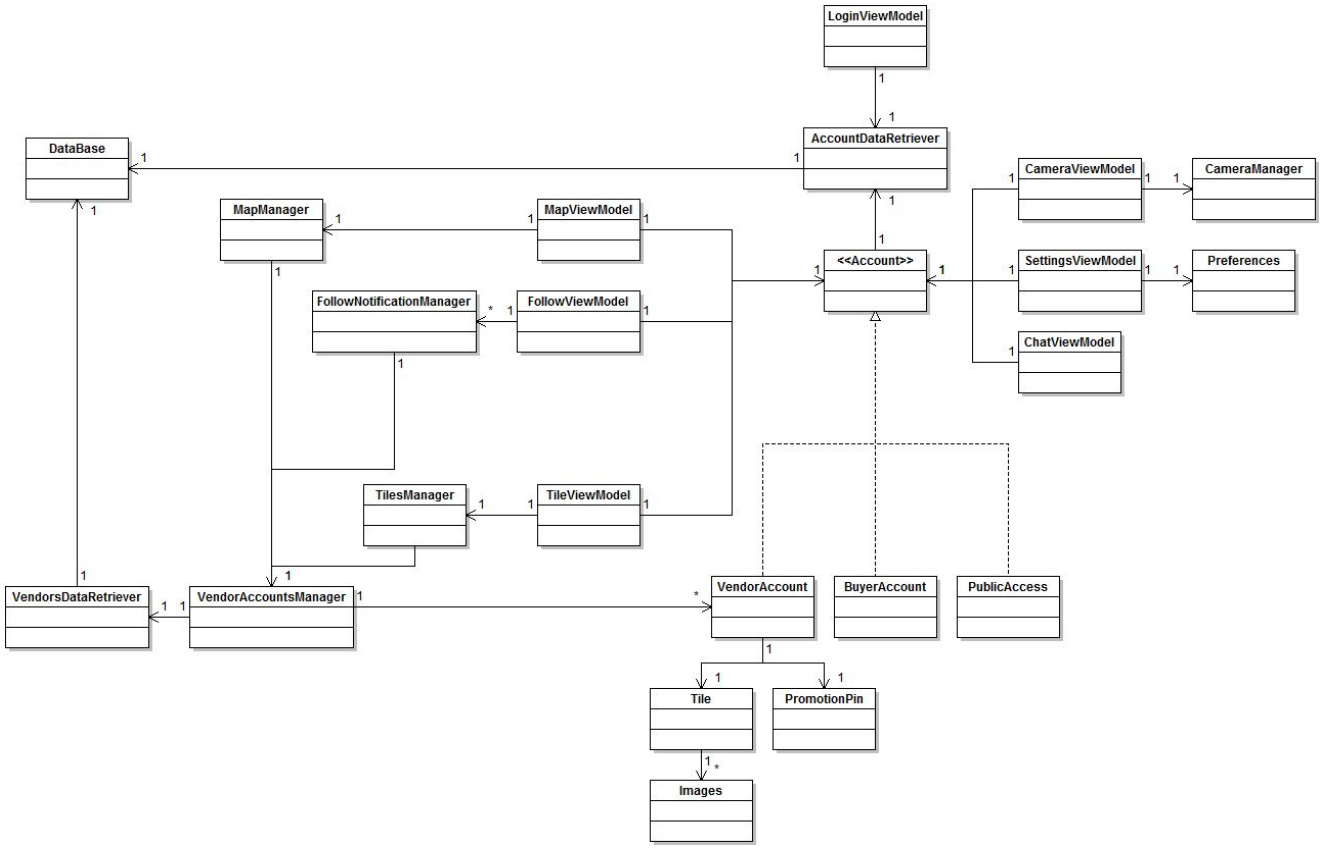
The business logic of the system will revolve around the account type used in the login phase. Each system component will behave differently depending on the type of account logged in. The components include the following: login, map, tile, notification, settings, and chat. The user interface views will be coded differently using mono and mono for android. However, each component will reference a view model which will contain the properties that will bind to different control elements of the view. The

properties of the view models will perform business logic that will be common to both views coded in mono and mono for android.

The different account types will be the *VendorAccount*, the *BuyerAccount* and the *PublicAccess*. Vendor accounts contain a the *PromotionPin* and the *Tile* objects.

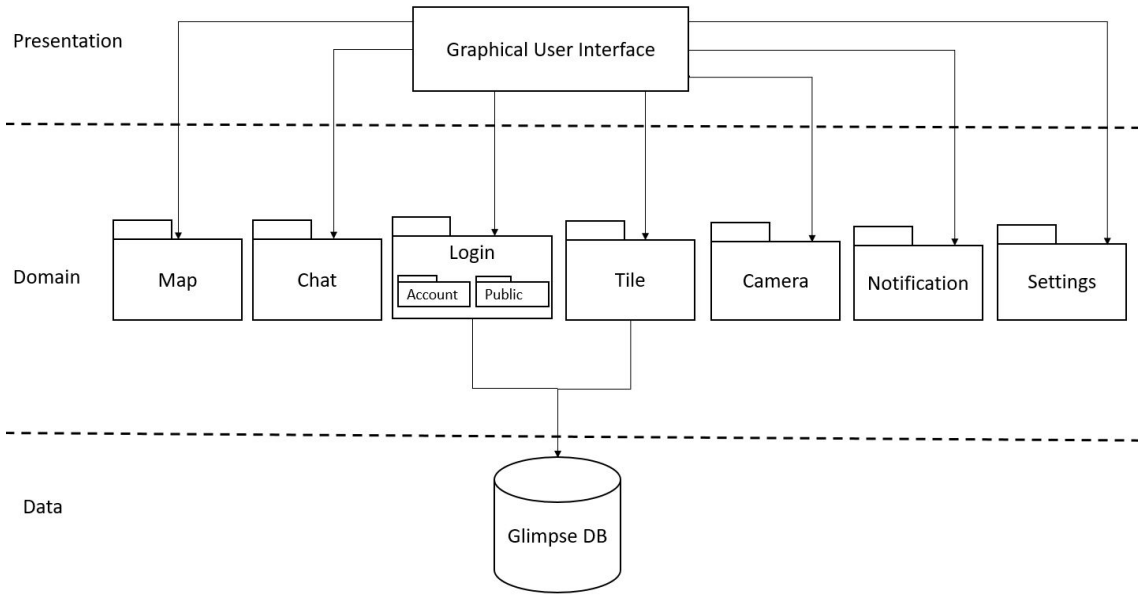
The map, follow notification and tile components will use a *VendorDataRetriever* to access the vendor's account information from the database.

Moreover, the camera, settings, and chat components will not use the database to retrieve data. The camera will be able to access the photo library of the user's phone. The settings will store the user's settings locally on their phone and finally the chat will simply manipulate data loaded on the app.



The layer diagram of our application is represented below and separates Glimpse into 3 tiers: user interface, domain logic, and data tier. The user will interact with the application through the presentation layer. Information related to the Account will be

stored and retrieved through the database. The logic tier coordinates the application and is composed of 7 main concepts: the map, the chat feature, the tile display, the camera, notifications, and the login process. Moreover it processes information between the data layer and the presentation layer.



Infrastructure

MvvmCross Framework (Application framework)

https://mvvmcross.com/

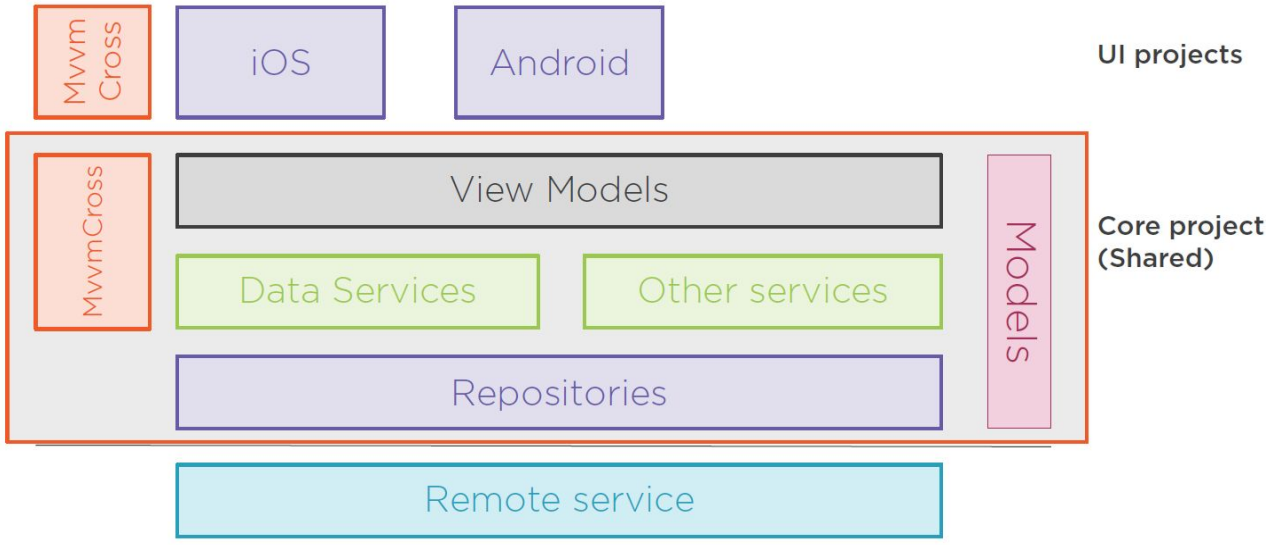
MvvmCross is a cross-platform mvvm framework that allows developers to create cross platform apps. Model-View-ViewModel is an software architectural pattern. MVVM is a variation of Martin Fowler's Presentation Model design pattern. Like Fowler's Presentation Model, MVVM abstracts a view's state and behavior. 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/

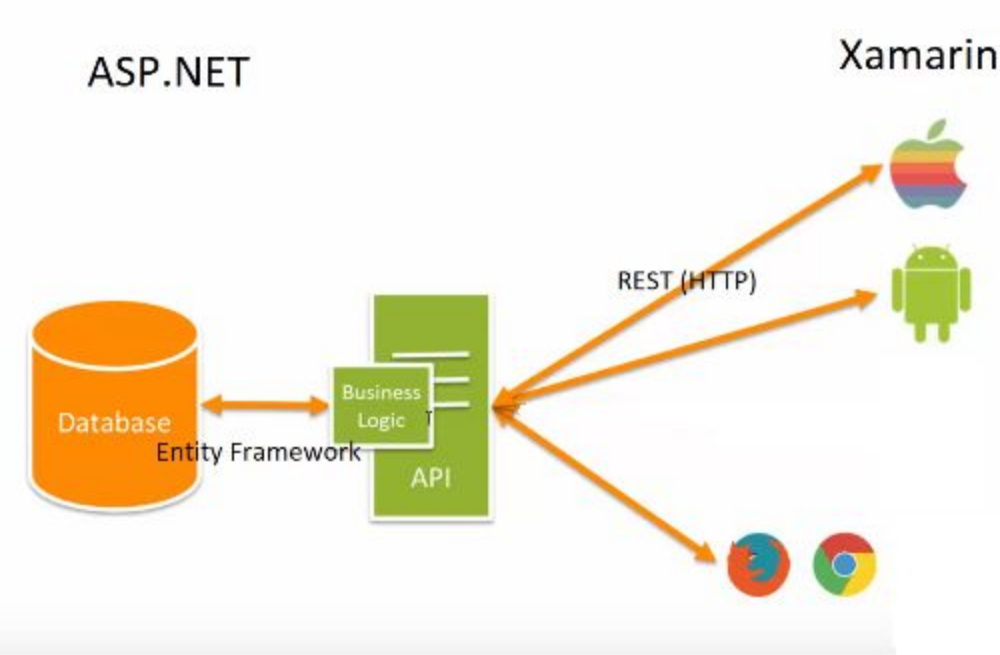
We have selected Microsoft Azure Database because our project is based on .NET and we're using Visual Studio, therefore it makes sense using Microsoft Azure. The Azure plugin allows us to easily connect to our remote database using the Server Explorer in Visual Studio which allows us to quickly view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