11/10:

Github repo created: <https://github.com/adrianp1206/HolyFit.git>

Started brainstorming fitness app

Looking into resources for database management

11/23:

Planning meeting confirms path forward

Fitness app will be designed with Xamarin on .net core 3.1

(<https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>)

Xamarin will allow deployment to Android and iOS

Azure and MongoDB will be used to build from Github and manage backend

12/3:

Xamarin configured with IDEs and Android/iOS SDKs installed and working individually

12/5:

Problems emerge in development, individuals are not able to pull project from Github

Xamarin works on an individual’s IDE with a new project, but is not working on cloned repos

12/8:

Executive decision to abandon app development because Xamarin will not work

**New project goal** – Design a webapp, still using Azure and MongoDB, but deploy as a webpage

Using .net MVC framework:

(<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-7.0&tabs=visual-studio>)

12/10:

Problems emerge with automatic build workflow in Github. This may be a .net C# specific issue.

(MSB error 1003)

It is also noted that .NET core 3.1 will no longer be supported after Dec 13, 2022.

(<https://devblogs.microsoft.com/dotnet/net-core-3-1-will-reach-end-of-support-on-december-13-2022/>)

Executive decision made to abandon .NET core 3.1 and MVC, but to use .NET6 and razor pages

**New project goal** – Design a webapp using .NET6 razor pages, Azure, and MongoDB

(<https://learn.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/?view=aspnetcore-7.0>)

New git repository created (<https://github.com/gabrown21/HolyFit.git>)

Azure set up for hosting webpage.

Project created and merged into master branch.

12/11:

Configured database to store user accounts/data and other website information

Started user model class

12/12:

Finish user model class

Build pages that utilize the user model to add goals and log workouts

Ideal timeline would have looked like:

Graphical user interface

Description automatically generated