Pokedex App

Starting points-----------------------

- Create a database that has Pokémon names entered into it.

- table I will enter 2 image strings that I can use to identify the images

- columns within each table, Pokémon name, Primary Key number,

- table called high score, saves name with a auto incrementing pk in case there are more than one “x name”.

? Existing user and new user implementation ?

User Stories---------------------------

As a competitive player I want to be able to save my score and display it as mine so I can brag to my friends.

As a Pokémon lover I want to be able to answer questions on EVERY Pokemon ever created so I can showcase my Degree in Pokémon.

As a casual Pokemon player I want to be able to answer some questions on a variety of Pokemon so I can mix it up.

As a Pokemon beginner I want to be able to carry on with the activity even if I get an answer wrong.

As a member of the sales team I want a 2 minute video explaining our product so I can show the product to potential buyers and investors.

As A gamer I want to be able to guess the name of a pokemon and get told if I am correct or not.

---------------------------------------------------------Product Backlogs----------------------------------------

- Score Tracking

-Create Database (Pokemon information stored)

-Gui Design)

-Highscore Tracker(Store score and name in database)

- images show, one blacked out one not, if (answer == false) show silhouette, else show full image

-String Comparison Logic

-DB image functionality

-2 minute video

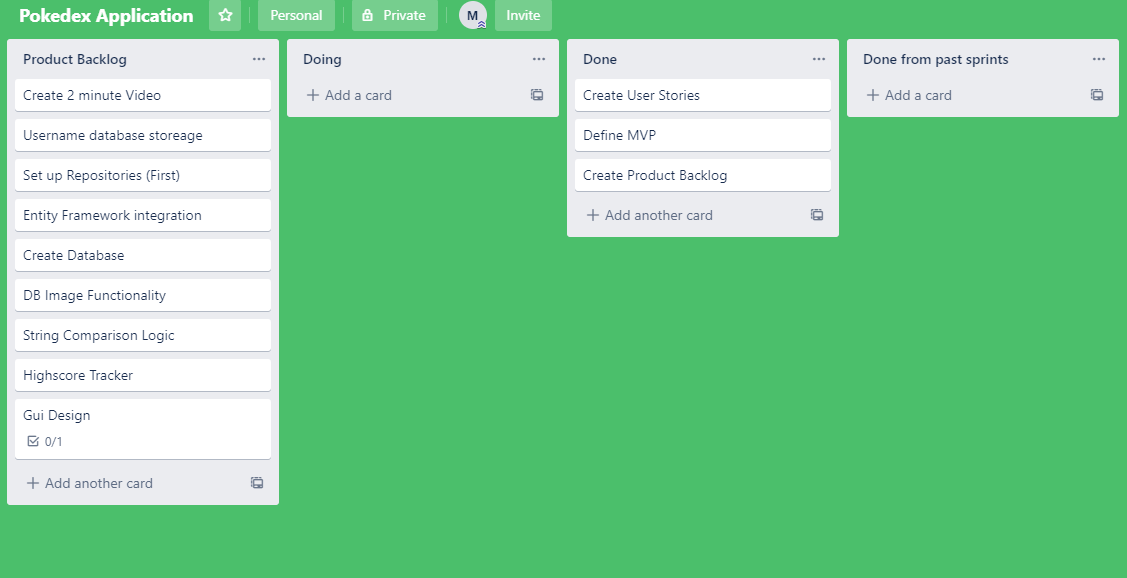
Sprint 0

Product Backlog

* Create user stories
* Create Product Backlog
* Create Kanban Board
* Set Up Repositories
* Define MVP
* Database for one Pokémon set-up
* Create 2 min Video
* Username Database
* Create Pokemon Database
* Create Highscore tracker
* Entity Framework integration

No True retrospective

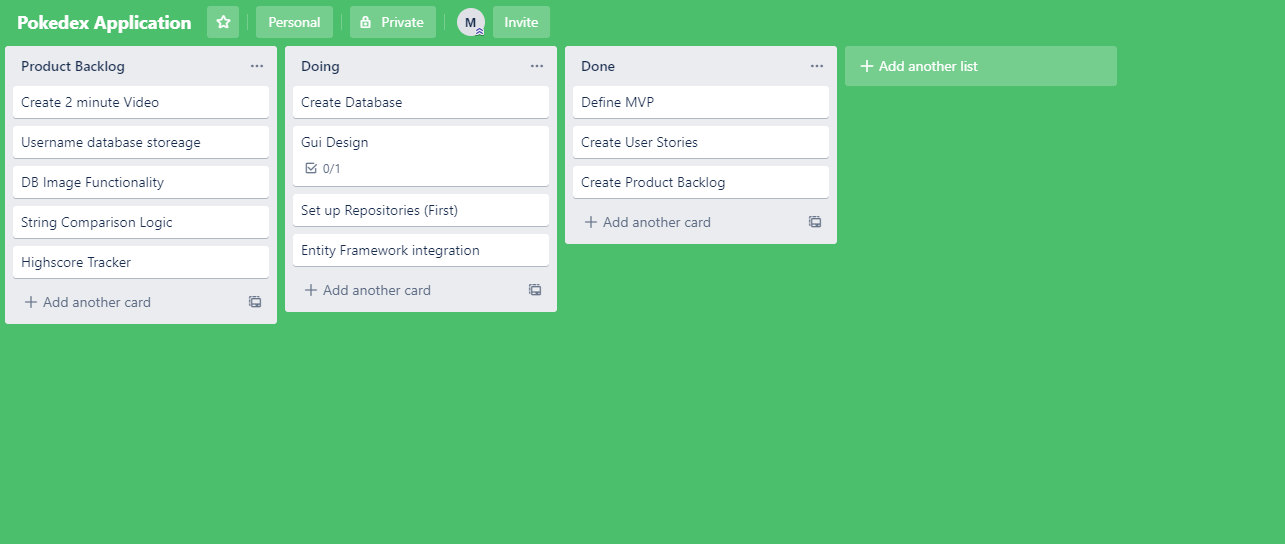
Planning feels like it’s gone well, I plan on moving on to creating the databases for my program and GUI’s tomorrow.



-Plan for sprint 1

-Set up database and functionality for 1 Pokémon

* - Create Databases for Pokémon information – DOD = Populated database(db) with one Pokémon name and two strings for file location.
* Entity Framework integration – DOD = Integrated db into visual studios and is able to perform a simple select \* from “tablename”
* Gui Design – DOD = A simple red GUI window for the games windows and the “login” window
* Set Up Repositories (first) -DOD = Setting up my repo so it is available to be committed onto when needed



Retrospective for sprint 1

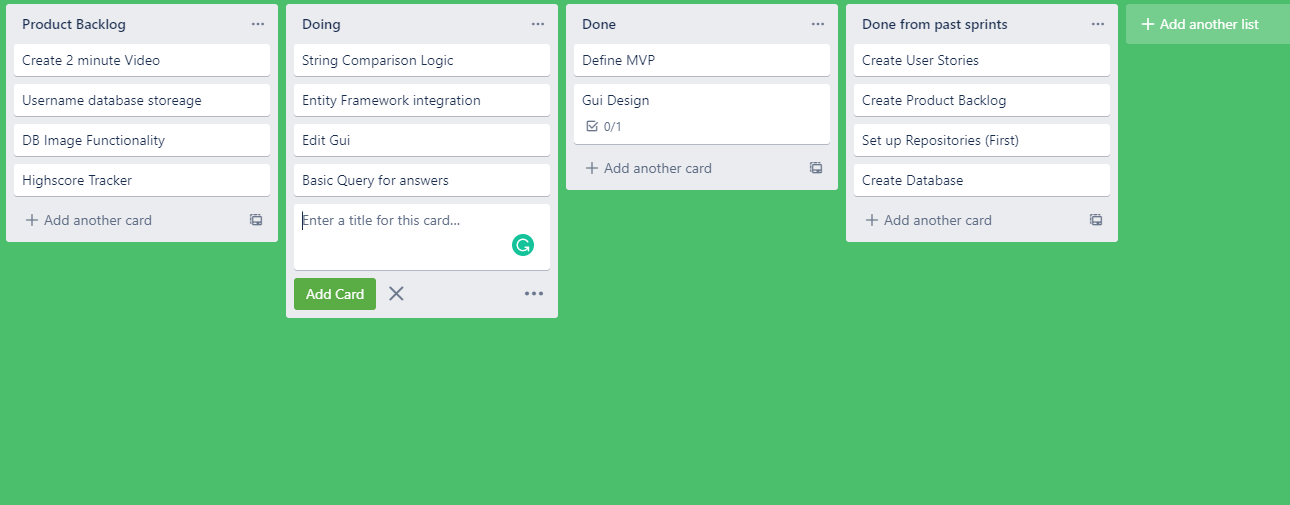
Sprint 1 Created a simple database for my Pokémon Names as well as made some of the GUI’s,

Hit multiple blockers such as not enough information on entity framework and crashes on visual studios due to image usage.

I will keep working hard and taking small breaks if I need a fresh mindset, I will start asking Catherine for help if I hit a blocker for too long, I will stop spending too much time on blockers and move onto another task to be more efficient with my time

Plan for Sprint 2

* Set up Entity framework – DOD = Entity framework is set up and db is in visual studio
* Set up basic queries for answers – DOD = receive a string in Console.WriteLine that matches the answer.
* Edit Gui to look more professional – DOD = matching colours and add a few pictures.
* Start String Comparison Logic – DOD = define what the correct answer is and match that to what a user inputs to decide if string is correct



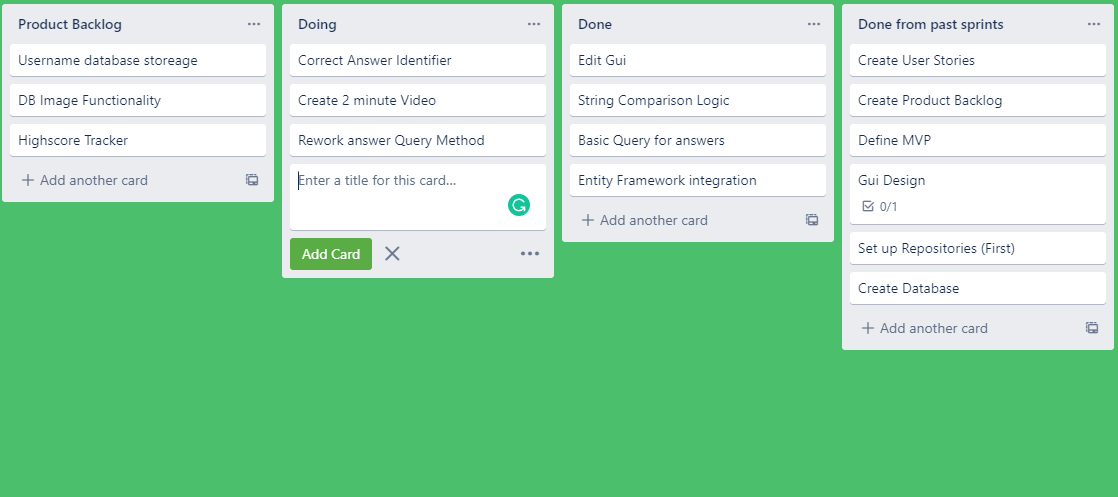
Sprint 2 retrospective

Made a lot of progress, finishing the integration of the entity framework and I made a query that allowed me to access the answers. I have also made a lot of progress in string logic. Also edited GUI so they are a higher quality.

I will keep asking Catherine for help where needed as well as keep being efficient with my time via timeboxes to prevent myself from using too much time on blockers. I will start creating more repositories to make sure my work is safe when I make a function work. I will stop over complicating an issue or function by granulating the task further and staying true to my MVP.

Plan for sprint 3

* Set up a correct answer identifier so that the game acknowledges when the user is correct – DOD = Game goes to a new window when correct
* Create a two minute program video – DOD = showcasing my program for some time to show that I have a MVP
* Rework the answer query so it is tidier and easier to understand and use in future developments -DOD = Refractor what I am doing with the query so it is more polymorphic



Sprint 3 retrospective

I created a 2 minute video showcasing my mvp after I refined and reworked some of my queries for my answer. My correct answer identifier needs more work and I have discovered today that the design decision that I have chosen does not allow me to implement some functionality that is important to improving my product past a MVP. This was the main blocker of this sprint as when a window closes and reopen, a new instance is created of the object “Game Logic” so when we assign a value to an int for integrating through the list it resets.

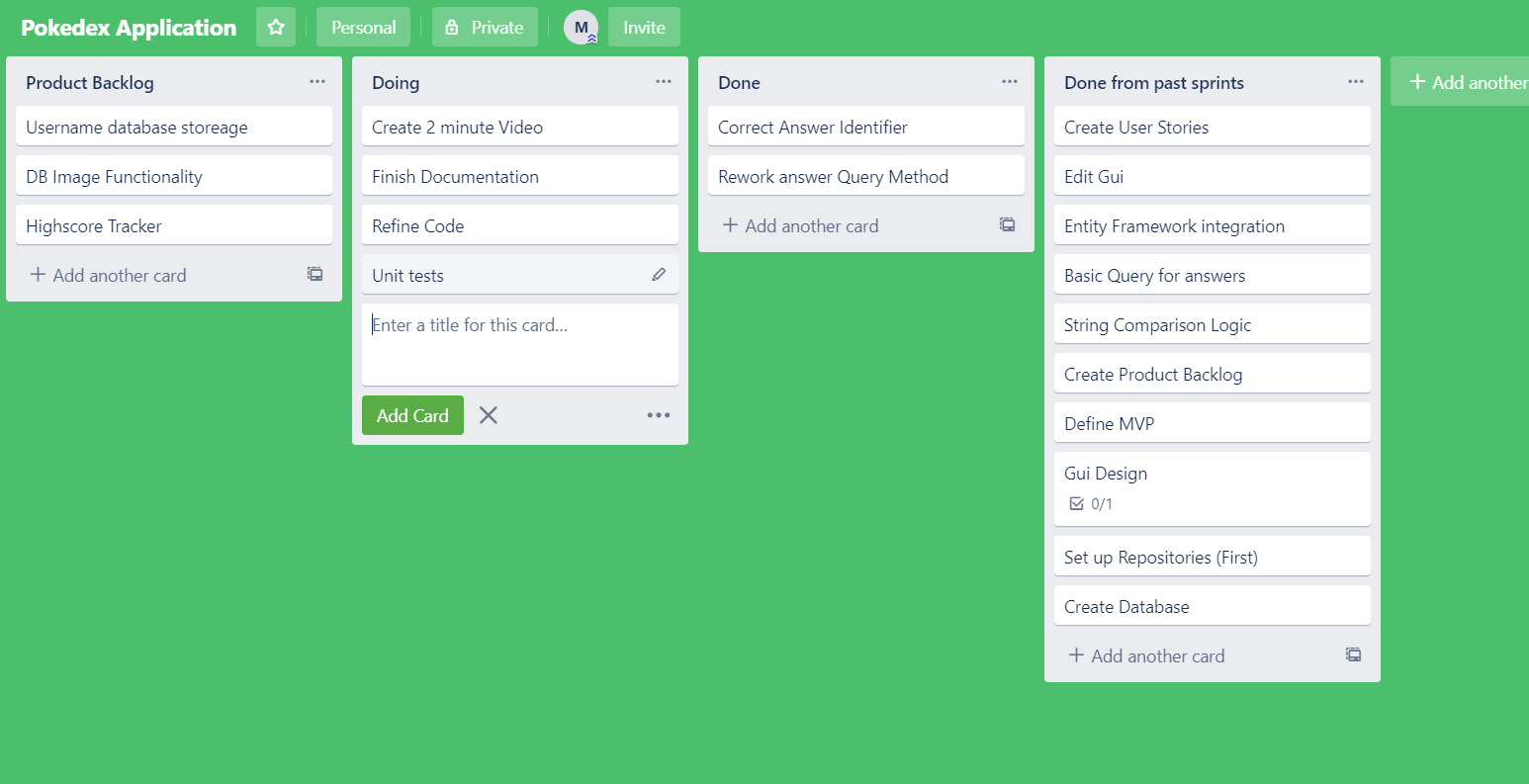
I will keep working hard through the last few issues to improve the quality of my product, I will start taking in the lessons learned from this project as a whole so that I can apply them to future projects and keep improving myself, I will stop worrying about large effort low reward tasks and take easy wins to improve my product.

Sprint 4 plan

* Finish Adding touches to MVP
* Attempt to refine Code to be more polymorphic
* Unit tests
* Finish Documentation

My plan for todays sprint is generally to refine some aspects of my program and finish my documentation. – DOD = sprint 4 documentation finished

I also intend on adding some unit tests to my work – DOD = at least one test in my project that passes.



Retrospective for sprint 4

I finished the documentation and managed to add a few quality of life changes to my code and the project as a whole by adding unit tests.

Lessons learned from this project is the hard reality of projects with strict deadlines. And what it truly means to have an mvp so that you can add to the core functionality or add others after you have your product.