

# Project specification – Syntax Error

Danilas Miscenko

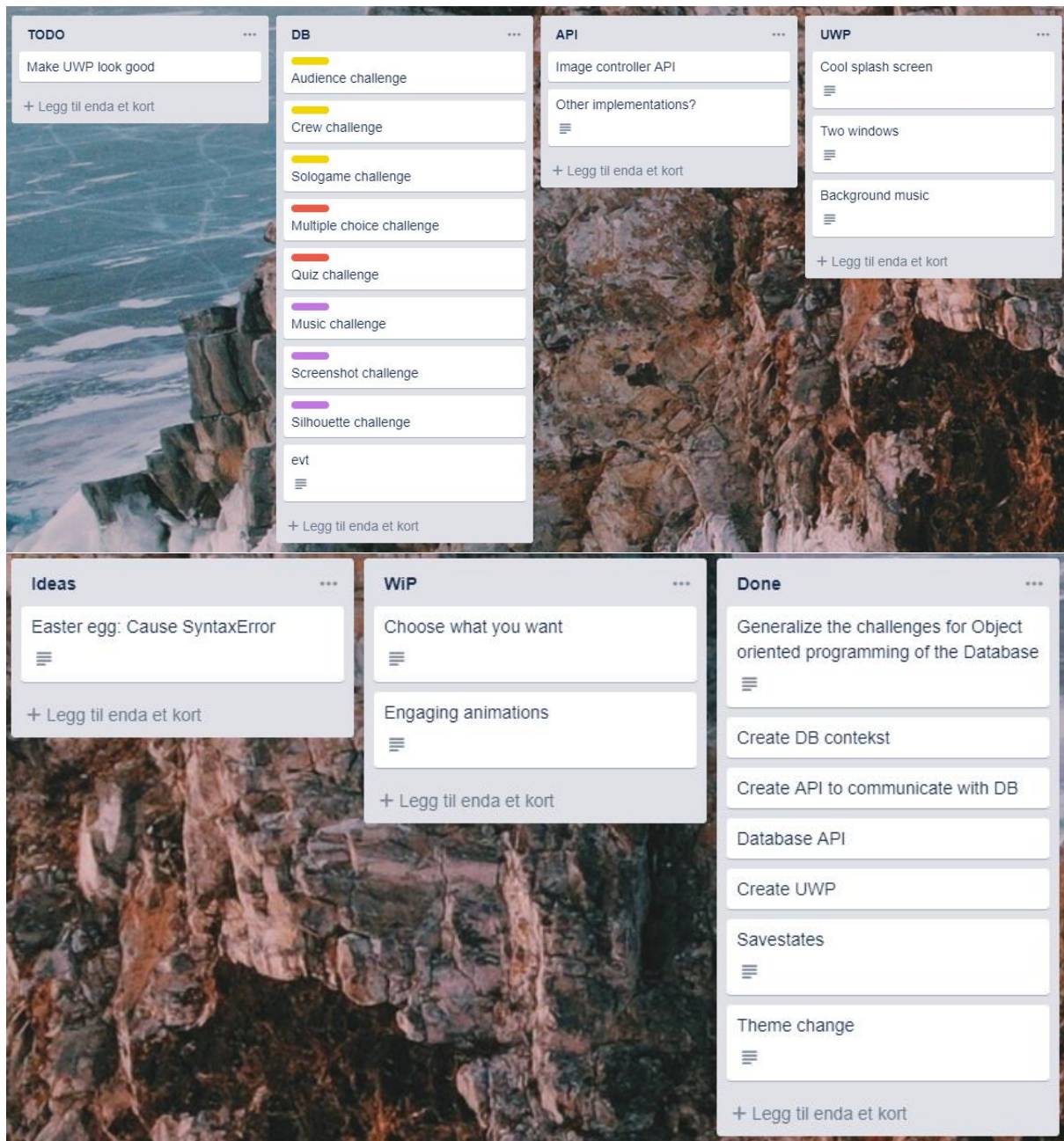


## Table of contents

Table of contents.....	1
Introduction .....	3
Functionality .....	4
Database: .....	4
API: .....	5
UWP: .....	6

## Introduction

Here is the trello page mentioned in the project specification:



As you can see, a lot of stuff has been done, and still a lot to go. The things that are done, as well as the things in progress will be discussed further down in the description with examples.

## Functionality

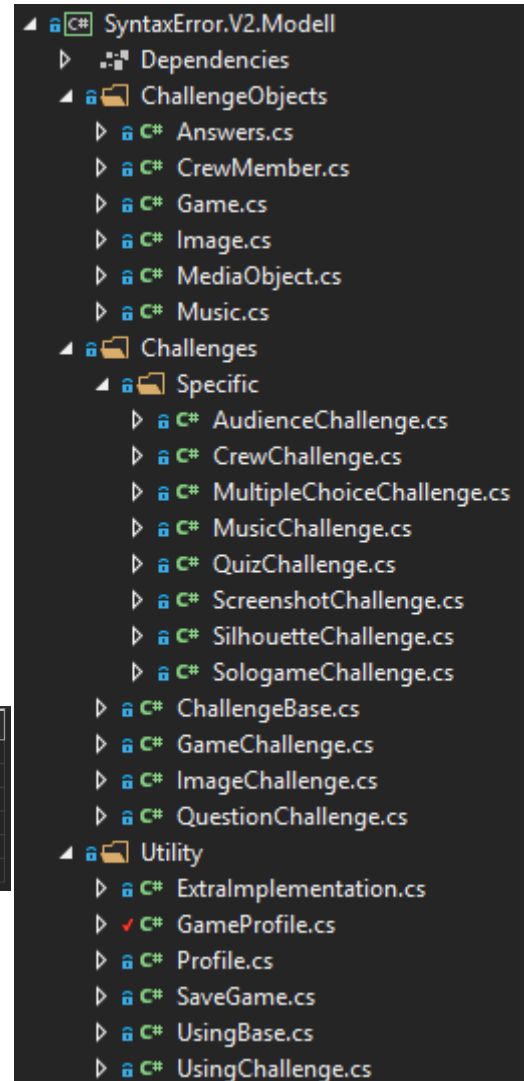
### Database:

In the picture to the side you can see my modell project. The Challenges folder is where all of the challenge classes are located. All of the specific challenges are inheriting from the classes indicating the challenge types, and those types all are inheriting from the class ChallengeBase. Challenge types are GameChallenge (that is the mother class for AudienceChallenge, CrewChallenge and SologameChallenge), ImageChallenge(that is the mother class for ScreenshotChallenge and SilhouetteChallenge) and QuestionChallenge(that is the mother class for MultipleChoiceChallenge and QuizChallenge). The only challenge that inherits directly from ChallengeBase is MusicChallenge, only because MusicChallenge does not fit any of these categories mentioned above.

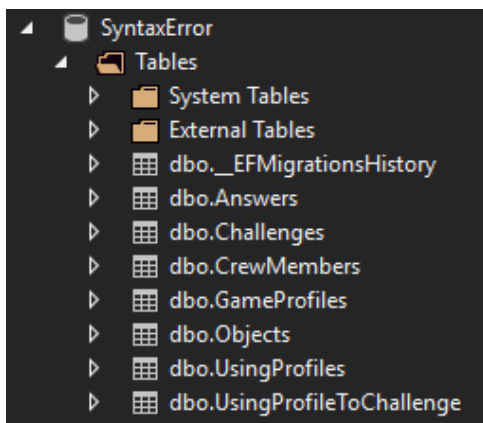
All of the challenges are located in a table called Challenges. This is how the table looks.

	ChallengeID	ChallengeTask	Discriminator	GameID	CrewMemberID	ImageID	SongID	AnswersID
▶	1004		AudienceChal...	6	NULL	NULL	NULL	NULL
	1005	Question	QuizChallenge	NULL	NULL	NULL	NULL	1002
	1006	Guess the song	MusicChallenge	NULL	NULL	NULL	5	NULL
	1007	What character ...	SilhouetteChall...	NULL	NULL	7	NULL	NULL
	1008		SologameChall...	8	NULL	NULL	NULL	NULL
⊞	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

As you can see, the specific challenge types are in the column Discriminator. This is to differentiate every challenge and cast it to the correct object programmatically when the information is retrieved from the database.



Here's the rest of the database tables:



Answers holds information on the answers for challenges that inherit from QuestionChallenge.

CrewMembers holds information on the crew members.

Objects holds information on all of the objects that inherit from the MediaObject class (these would be Game class, Image class and Music class).

UsingProfileToChallenge houses information on which challenge is used in which Profile or SaveGame (More on that later).

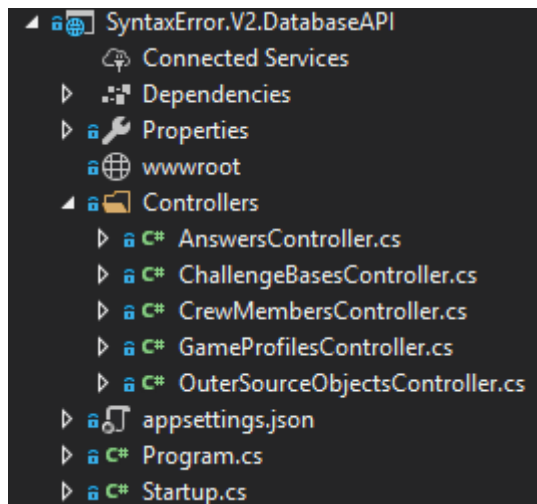
UsingProfiles houses the Profiles and SaveGames.

GameProfiles houses the information on which Profile and SaveGame is for each game created, amongst other information

The whole database is created with code first from EntityFramework. The code for entity framework is in the file SyntaxErrorContext in the project DataAccess.

## API:

The only API that's done at the moment is the DatabaseAPI. I still have to make an ImageServiceAPI.



The controllers all have appropriate operations for the information they are supposed to get from the database.

Some noteworthy stuff to mention:

AnswersController has only the GET operation for all and one specific answer. The reason for this is that a QuestionChallenge can not exist without an answer, and vice versa, and since the challenge is a bit more prevalent in the App, then deleting the challenge deletes the answer. I should add a POST operation so that an answer can be edited, but a PUT is unnecessary, since an answer gets created with the

challenge.

CrewMemberController has a big DELETE operation that deletes both the crew member and the challenges that that crew member was in, because a CrewChallenge is useless without a crew member.

GameProfilesController includes both the Profile and the SaveGame that the GameProfile has in the GET and DELETE operations.

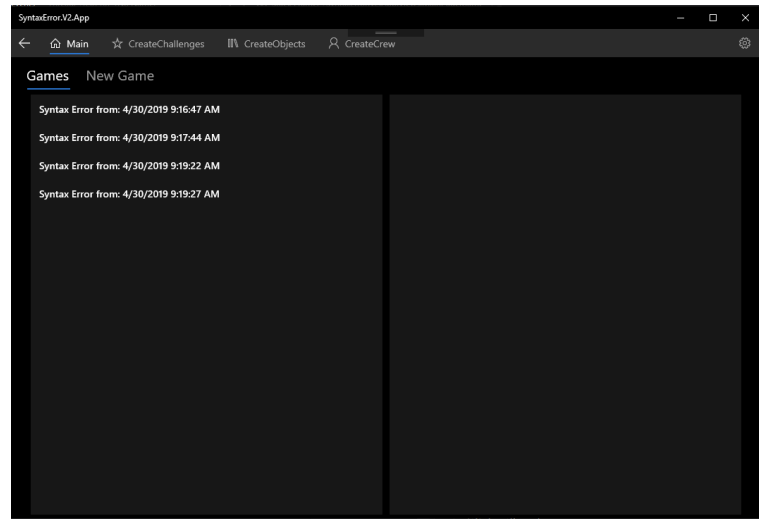
OuterSourceObjectsController DELETE operation deletes the challenges the objects are a part of for the same reason that the CrewMemberControllers DELETE operation deletes the CrewMemberChallenge the crew member is a part of.

## UWP:

At the moment the UWP does not have a lot of functionality. Here are some screenshots:

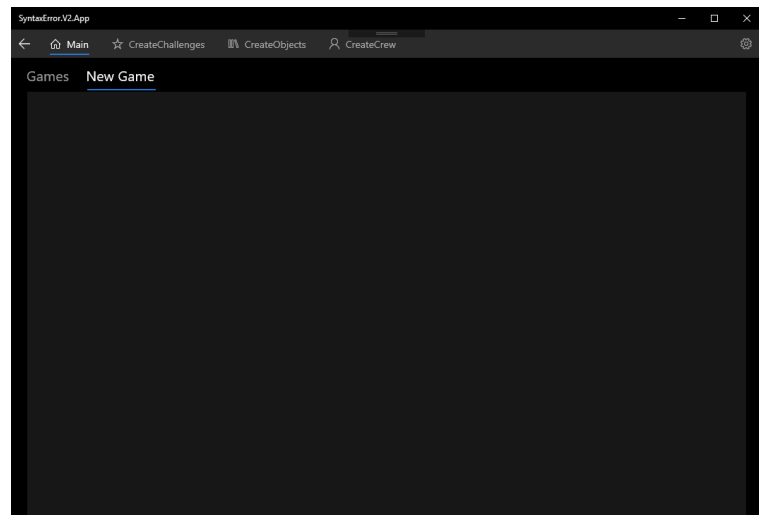
A list of all GameProfiles in the database:

I want to add buttons for deletion of the game profile and to start the game, but the only thing functioning at the moment is a delete button besides the game. It is currently commented out because of an error, but it uses a delete command found in the the MainPageViewModel. I want to also add functionality to list the challenges the game profile contains of besides in a list view when a game profile is clicked.



New game tab in the MainPage:

I want to create new GameProfiles here, but I have not yet made that functionality.



All of the other pages are basically identical to this:

Here I want to list the objects that are of the type "PageName". This should then have functionality to Create, Edit and Delete these objects.

