|  |
| --- |
| IT1B |
| Project Innovate |
| A smart solution to finding friends for your dog |
| Philip, Hylke, Hemran, Muhitdin, Patrick, Nadie  28-06-2019 |

Index

Server and Database……………………………………… Page 3

AI (Artificial Intelligence)……………………………….. Page 4

App……………………………………………………………….. Page 5

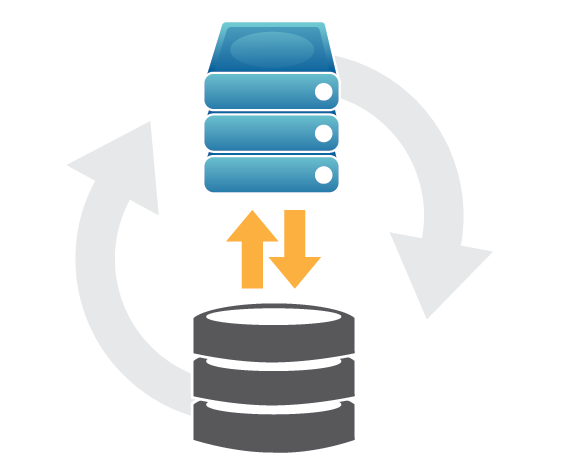
Appendix A: Functional Design

Appendix B: Storyboard

Server and Database

**Server**

The “Breedr” app runs on a server, this server is used to keep the app online 24/7, run the AI whenever required (will be explained further in the AI chapter) and to host the database to save all data created and used in the app. Currently the prototype of the app is run on a dedicated windows server.



**Database**

The “Breedr” app makes use of a database to save and store data retrieved from the app and AI. The database was created in MySQL. The database has been connected hosted on the server, to be useable in combination with the app. The database contains different tables, these tables include: User, Messages, Preferences, Breed, Swipe, Match, Dog, Userimage and Image. These tables work in tandem to create the “Breedr” database. Each table has its own function.

User: Saves user data.  
Messages: Saves all data from messages sent between users.  
Preferences: Saves data based off the preferences selected in the app by every user.  
Breed: Saves all breeds available for the AI.  
Swipe: Saves the 100 most recent swipes from each user registered.  
Match: Saves matches between users (2 users swiping right on eachother).  
Dog: Saves all dog data for each user.  
Userimage: An in between table used to save imageid’s to users and their dogs.  
Image: Saves the images and their path within server.

More information including a detailed entity relation diagram can be found within Appendix A: Breedr Functional design.

AI (Artificial Intelligence)

**AI (Artificial Intelligence)**

The AI consists of two parts, the backend C# application and the actual AI Python script. The application constantly listens to the database for new entries within the image table, when the listener event is triggered, the Python script (AI) gets called with the image path as a parameter. Once called the AI checks the image to see whether the image is a dog or not and then checks for breeds.

The AI has been pretrained with a large library of dogs of different breeds to be able to find patterns, using these patterns the AI is able to recognize whether an image is in fact a dog and its breed(s).

Once listener event gets called caused by the AI receiving an image, The AI scans through the image looking for patterns, based off the images it was trained with. First the AI checks whether the image is a dog or not, if it is the AI continues and attempts to predict the breed based off percentages of recognition, first the AI checks if the breeds match the breeds within the breed table within our database and then inserts the breed name into the dog table. After this event has finished the user will be allowed to continue forward and register their account.



App

**App (Android Application)**

The “Breedr” app was created with PHP and converted to an app using Android Studio, the current version of the app only works for Android versions above v6.0 and currently does not work on IOS.

Once you open the app you are shown the login page allowing you to register a new account or login with an already existing account, having logged in you are able to set your preferences for the type of dog you are interested in, these preferences include: breed, age and distance. After having registered an account and setting your preferences, you can start with swiping! Once you’ve swiped enough, you’ll match with someone, this “match” will give you a notification and give you the opportunity to start messaging the person you matched with. Lastly when you’re done messaging you can set up a date with your match for your dog(s).

A more detailed description including images can be found in Appendix B: Storyboard “Breedr”

