



BSc in computer science

Machine Learning Repo

ML-Repo

Team:

Gabríela Jóna Ólafsdóttir
Guðmundur Hermann Þórlaugarson
Jóhann Hinrik Jónsson
Katrín Guðmundsdóttir
Magnús Freyr Sveinsson

Supervisor:

Gunnar Sigurðsson

Examiner:

Sigurjón Ingi Garðarsson

09.05.2020

1	Introduction	5
1.1	The Team	5
2	Project Description	6
2.1	Main Milestones	6
2.2	Contacts	7
2.3	Communication with OZ and working conditions	7
3	Project Structure	7
3.1	Methodology	7
3.1.1	Time management	8
3.2	Project Schedule - Sprints	8
3.2.1	Daily Tasks	8
3.2.2	Retrospectives	9
3.3	General working methods	9
4	Tools & Rules	10
4.1	Coding rules	10
4.1.1	General rules:	10
4.2	Coding guidelines	10
4.3	Programming languages and libraries	10
4.3.1	Python/JavaScript:	10
4.3.2	React:	11
4.3.3	HTML/ CSS:	11
4.3.4	Github:	11
5	Risk Analysis	12
5.1	Risk Analysis	12
5.2	Incident Log	14
6	Analysis	15
6.1	User Interview Analysis	15
6.1.1	Questions	15
6.1.2	Answers	16
6.1.2.1	Very important	16
6.1.2.2	Quite important	16
6.1.2.3	Nice to have	16
6.1.3	Conclusions	17
6.2	Product Backlog	17
6.4	Use Cases	19
6.4.1	Use Case 1	19
6.4.2	Use Case 2	19

6.4.3 Use Case 3	20
6.4.4 Use Case 4	21
6.4.5 Use Case 5	21
6.4.6 Use Case 6	22
7 Design	23
7.1 Prototypes	23
7.1.1 Design Draft	23
7.1.2 Database Design	26
7.1.3 Updated Database Design	26
7.2 Final Design	27
7.2.1 Opening window after login	28
7.2.2 Searching: Cibao FC	28
7.2.3 Filter View	29
7.2.4 List (Toggle) View	29
7.3 System architecture	30
7.3.1 Technical stack	30
7.3.2 System Design	30
8 Progress Overview	31
8.1 Sprint schedule and events:	31
8.1.1 Sprint 0	32
8.1.2 Sprint 1	34
8.1.3 Sprint 2	36
8.1.4 Sprint 3	38
8.1.5 Sprint 4	40
8.1.6 Sprint 5	42
8.1.7 Sprint 6	43
8.1.8 Sprint 7	45
8.1.9 Sprint 8	47
8.1.10 Sprint 9	49
8.2 Summary of Sprints	50
8.2.1 Project burndown chart	51
8.2.2 Working hours per student	51
9 Product	52
9.1 Final product	52
9.2 Website features	53
9.3 Learning experience	55
10 What lies ahead	56
10.1 Problems encountered	56
10.1.1 Testing	56

10.1.2 Communication	56
10.2 Next steps	56
10.3 Vision	56
11 Conclusion	57
12 Appendix A	58
12.1 Glossary	58
13 Appendix B	59
13.1 Interviews	59
14 Appendix C	64
14.1 Code example	64
15 Appendix D	66
15.1 Time Logs	66

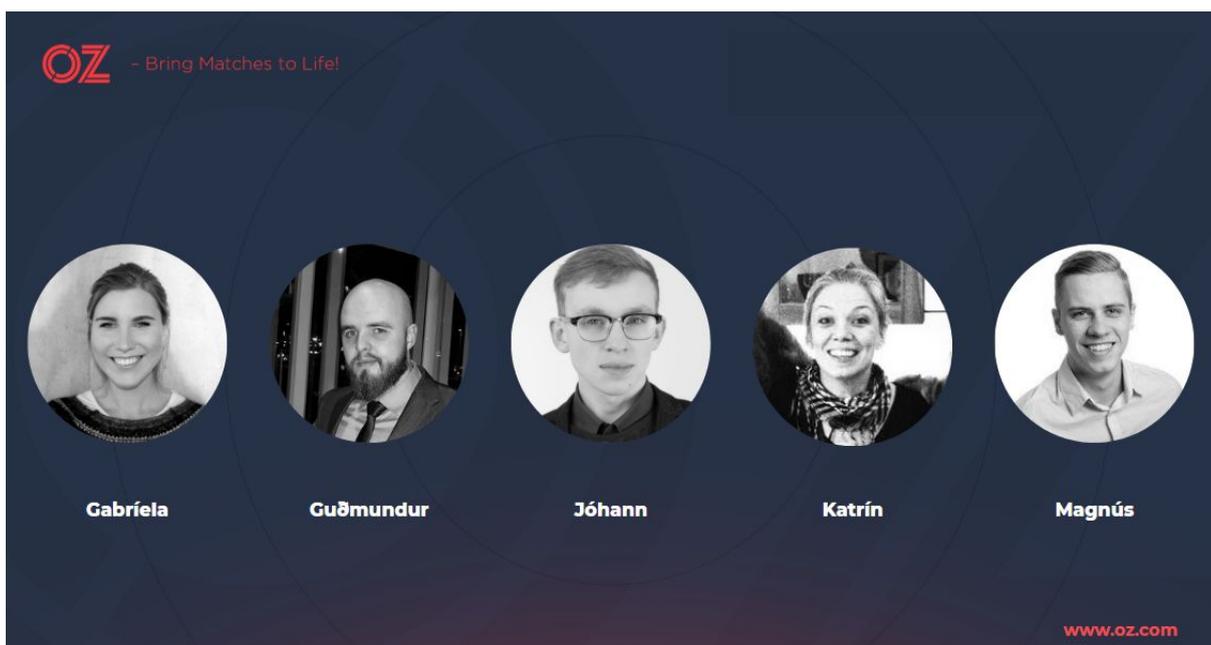
1 Introduction

This report outlines the final project for our BSc in Computer science for Reykjavík University and the process behind it. We are a group of five students in our final year and this project is in essence the accumulation of all the knowledge and skills we have acquired over the time spent at the university.

We are collaborating with OZ sports which is a sports-tech company with operations in Miami, London, Santo Domingo and the Nordics. OZ sets up multiple cameras on Football stadiums and records the matches remotely. This way they aim to give every sports league and tournaments an opportunity to broadcast to the world for a fraction of the current cost. But with this comes a lot of data and recordings that need to be accessed by current staff. Right now the process to access this is quite complicated and they need to gather the backlog with a more user friendly service.

Our team has been working on this project to set up the user friendly service that current staff of OZ sports can use to access old and new game footage more easily by creating a platform that gathers the footage that is needed. This is important since OZ is working on using AI to direct the matches and thus need to be able to easily access footage, frames or audio. This database is crucial for the ML needed in that process.

1.1 The Team



The team consists of two girls and three boys and the age difference between the youngest member and the oldest is 20 years. Two team members have children and one team member has a few prior degrees including a masters in International Relations. Even though all of the students have Computer science as their major two have Business as a minor and one student has worked for Oz Sports since 2018. The fact that we are so versatile in age, gender and background has proved to work in our favor since different eyes see different things. It has also made for a more creative workplace and helped us effectively distribute the work.

2 Project Description

When we started there was no central server where employees of OZ ehf. have access to footage produced by the company. The data was mostly the football matches that the company produces with special equipment located around the world. At the time the end product was only stored on the company website where users can watch football games, so there was a need to collect the backlog to a central server where employees can download them unencrypted and without DRM. Along with installing scripts that take the content of the video production equipment mentioned above and backing it up to the same central system, you would need to submit a final product (program feed) and a center camera (clean main camera feed) along with an overview from what is happening on all cameras (camera matrix) to the new platform.

The company also saw the need to keep track of more points that come in during the broadcast and to record them with a time-stamp when these events happen e.g. changing the camera, how the camera is moved. But that is done by connecting to an existing API and registering the following events in a database. Since the company has systems in different countries and in different places, it seemed as if it should have been possible in a simple interface to sort games by country, team and etc. along with being able to transfer the videos from Data Center A to Data Center B for processing. We also believed that if time would be on our side we could run OpenCV on the entire tunnel, or apply various free software from the ML world to the data and document it. Time was unfortunately not on our side, but a very user friendly efficient platform is available and ready for the OZ employees to use to for example to record when goals are scored by running OpenCV on the clock and or run an open ML algorithm on the dataset.

2.1 Main Milestones

We visualised some main milestones to implement:

1. Database with 5-10 videos to work with.
2. User-friendly web service to access videos.
3. Make an API to get the metadata from the videos in the database.
4. Footage can be filtered, altered and easily used for ML.

2.2 Contacts

Contacts from OZ:

Ragnar Bragi Sveinsson, Technical Manager: ragnar@oz.com

Jóhann Hinrik Jónsson, Product Owner: jonni@oz.is

2.3 Communication with OZ and working conditions

The team decided in the beginning to meet up at least three times a week in the OZ headquarters. Daily scrum meetings were scheduled to take place either in person or through messenger and Asana chose to keep eyes on tasks being worked on. We also chose to use Facebook messenger to communicate. The number of meetings at OZ quickly went up to four to five times a week, where most team members tried to, even if only for a short period, work at the headquarters on weekdays. However that all came to a halt because of COVID19 and all businesses, schools and etc. closed due to the global pandemic and then the working conditions where our homes and all meetings were conducted online.

We had a very nice access to a room at the headquarters of OZ, with desks and chairs for use as we please. We had complete freedom to rearrange and set it up in any way we pleased and we get delicious home cooked meals 3 times a week. Also to our benefit (or hindrance) there was a foosball table setup in the middle of the room. We had the opportunity to work there every weekday and in the beginning the team had fixed meetings on Tuesdays, Wednesdays and Thursdays from before noon until five, that changed to trying to come every weekday but then of course in the end nobody could go and we met online from our homes.

3 Project Structure

3.1 Methodology

The team decided to use SCRUM as a framework method around the project. SCRUM is an agile process framework and is very suitable for teamwork on a project of this size as it breaks down the project into smaller units and gives the team a clear view of what has been done, who is doing what and what needs to be done.

Scrum consists of the team, a product owner and a scrum master. In our project Jóhann Hinrik Jónsson is the product owner and Katrín Guðmundsdóttir is the scrum master, though by the end of the process and due to unforeseeable circumstances Gabriela Jóna Ólafsdóttir helped and took over some scrum master duties. The product owner is responsible for the direction of the project as a whole, the features it will have and which tasks are to be prioritized. The scrum master manages actions and oversees the process, and has to make sure at the same time that the team can work freely and needs to remove obstacles if they occur.

The team will work with the Vertical Slice architecture where we will divide the project into several small projects which slice through all components of the project, thus having a functioning prototype early on in the process which grows bigger after each sprint.

3.1.1 Time management

We use Excel to track time spent on the project. Our goal is 1500-1750 hours over the semester which we have divided between sprints not always equally but on average 136 - 159 hours which is 27-32 hours per person each sprint.

3.2 Project Schedule - Sprints

The team decided to start the project with sprints that last for two weeks, from Thursday to Thursday. By the end of the project we found out that one week sprint suited us better, so the last two weeks were a one week sprint.

At the start of each sprint we met up for sprint planning, where all members of the team discussed and decided on the goals of the upcoming sprint. During the sprint planning we came up with a Sprint Backlog which consists of stories from our Product Backlog that have been divided into tasks and each task has an estimated time frame.

At the end of each sprint we had a Backlog grooming meeting where the team checks if new stories have popped up during the sprint. If so, then they will be added to the Product Backlog. It is also used to review whether or not the importance of user stories has changed

during the sprint and or stories need to be re-edited. That will contribute greatly to a current and usable Product Backlog. Stories, we decided, should be of manageable sizes so that nobody feels overwhelmed or like they have been doing the same thing forever, this in retrospect we struggled to uphold.

3.2.1 Daily Tasks

As a part of the scrum process we decided that the team should have a short gathering daily, sometimes in person and other times online, where we plan the next 24 hours and give a short report of what we have done the past 24 hours. This is a good way to keep a clear view of what is being done and also a great motivator for the team. The tool we used for this communication started as Messenger, moved to video meetings on Whereby and ended on Microsoft Teams, as by the end we really had to rely on online and video communication.

3.2.2 Retrospectives

The Sprint Retrospective occurred after the Sprint Review and prior to the next Sprint Planning. The retrospective session is basically an “improvement” meeting held to find ways and means to identify potential pitfalls, past mistakes, and seek out new ways to avoid those mistakes, which are attended by all – the product owner, scrum master, development team members, and optionally with our supervisor, depending on the sprint before, necessity and the availability of the supervisor. We decided from looking at the since the total sprint duration is two weeks, then the duration of the Sprint Retrospective will be 90 minutes. The rule of thumb for the length of a sprint retrospective meeting is that it usually takes no more than 45 minutes per week of sprint duration, hence why we chose 90 minutes.

For conducting the sprint retrospective meetings the team will have five steps in mind:

1. **Set the stage** – Set the goal; Give people time to “arrive” and get into the right mood
2. **Gather data** – Help everyone remember; Create a shared pool of information (everybody sees the world differently)
3. **Generate insight** – Why did things happen the way they did?; Identify patterns; See the big picture
4. **Decide what to do** – Pick a few issues to work on and create concrete action plans of how you’ll address them
5. **Close the retrospective** – Clarify follow-up; Appreciations; Clear end; How could the retrospectives improve?

3.3 General working methods

1. **Have clearly defined team objectives:** To move toward a key goal in one direction, people need to have a very clear understanding of the destination.
2. **Efficient team meetings:** So according to a survey conducted by Microsoft, it is said that professionals waste up to 3.8 hours a week on unproductive meetings. We intend to turn this around with thorough preparation, because like the saying goes: if you fail to plan, you are planning to fail.
3. **Make sure individual progress is visible to the whole team:** It communicates three essential parts about every team member: biggest achievements, current goals, and major challenges.
4. **Make it fun:** Working in a team should not feel like an obligation. We intend to integrate fun and humor as much as we can, studies show it can really make the team collaboration more efficient. Socializing and getting to know each other makes the team more dynamic and connected.
5. **Show respect and be honest:** All members of the team have lives separate from the project and we all have good days and bad days. We prefer to admit to struggles rather than pretending to succeed.

4 Tools & Rules

4.1 Coding rules

In order to synchronize code the team set down some basic coding rules. This is extra important when assignments are developed in a team, both in order for an accessible code for other team members and so that class names are consistent. There is also a need to unify how comments are used in code. Common code rules were a primary source in making these rules with team member preferences a second.

4.1.1 General rules:

- KISS - keep it simple (stupid)
- DRY - don't repeat yourself
- SRR - single responsibility rule

4.2 Coding guidelines

- Safe: It can be used without causing harm.
- Secure: It can't be hacked.
- Reliable: It functions as it should, every time.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment.
- Consistent code quality — no matter who writes the code.

4.3 Programming languages and libraries

- Python
- Javascript
- Node.js
- React

4.3.1 Python/JavaScript:

- The code must be executable.
- Descriptive names of variables and functions must be used.
- The code must be consistent.
- PascalCasing used for all functions (i.e. words in functions begin with a capital letter, eg Func ()).

- CamelCasing used for variables (i.e. the first word in a variable starts with a lowercase letter and then the next one with a large character, such as myData).
- Underscore is used in naming private variables (e.g. `_privateVariable`).
- Sleeper brackets up, thus opening in the same line.
- The correct indentation is used in the code, made with a tab but not spaces.
- Space is used in loops and before and after expressions and numbers.
- Namespaces are clearly organized.
- The code will be well-written, ie. well commented since it does not explain itself.
- The code and comments are to be written in English.

4.3.2 React:

- Linters will be used.
- Prop-types are used to define props.
- DisplayName will be used.
- Error boundaries.
- Folder structure will be clear and logical.
- Component naming will be by convention.

4.3.3 HTML/ CSS:

- All HTML- and CSS code is validated.
- Tags start with a small letter.
- All tags are correctly nested and indented.
- File names start with a small letter, e.g. `styles.css`, `index.html`.
- Hyphen used in names, e.g. `.this-class {`
- Brackets come straight after rule names e.g. `.rule1 {`
- Rules are grouped together by type.
- One line between rules.

4.3.4 Github:

- Git commit in English.
- Commits should be descriptive of the task, no “blah” commits.
- Never commit a broken code.
- Use feature branches for new features.
- Nothing is merged with the master unless it works on the feature branch.
- Use pull requests.
- A person that creates a pull request can not merge it themselves.
- If the code fails, work is stopped and the system is restored.

5 Risk Analysis

We conducted a risk analysis to identify issues that could pose a threat to the project. We then decided on handlers to work on reducing the probability and/or impact of each risk. J stands for Jóhann, the product owner and K stands for Katrín, the Scrum Master. Some are assigned to the team as a whole. The risk analysis was done at the beginning of all real work but because of the rise of COVID19 we reevaluated the risk of illness midway though and changed the analysis. When we evaluate we factor in both the chance of it happening and the effect it would have on our likelihood to deliver a good project on time. The multiplication of those two were the total rate and we put them in the following order, with the highest total on top.

5.1 Risk Analysis

Risk	Prevention	Mitigation	Chance	Effect	Total	Handler
Accidents, illness and other life events	Not much to be done to prevent such things	Show patience and be helpful when other members struggle	4	7	28	Team
Project clashes with other school assignments	Plan ahead and be organized	Work extra hard and divide work according to team members schedule	5	3	15	Team
Underestimating the time project parts take to finish	Take care in evaluating the importance of tasks and rather overestimate time	Update Product backlog regularly and evaluate importance and time	2	5	10	Team
Disagreement between team members	Agree on a set of working rules and try to be patient and respectful	Listen to other members and diffuse situations that could lead to a disagreement	2	3	6	K
Product backlog does not cover the necessary requirements	Do user interviews at the beginning and test early prototypes with users during the	Update the Product backlog to the users requirements and keep an open mind on the	2	3	6	Team

	process	projects result				
Data or work loss	Be careful to take back-ups and store data and work in reliable places	Work as a team to retrieve or redo lost data and/or work	1	5	5	J
User stories are of very different size	Thorough estimation and planning poker	Let other team members know if we are struggling and be willing to help others	2	2	4	K
Technical environment problems	Choose environments carefully to fit the tasks ahead	Find new solutions and re-evaluate if necessary	1	4	4	J
Problems with working environment	All members have to be prepared for changes	Find a new working environment or fix the problems in current one	2	2	4	Team
Hardware problems	Each member is responsible for own hardware	Borrow or buy new hardware	2	2	4	Team
Product owner changes scope or requirements	Keep a very open working relationship with product owner and OZ	Integrate new requirements to the existing project	1	4	4	J
Team members are working on the same task	Do daily scrum meetings and use Asana to prevent tasks being duplicated	Make sure that this is discovered quickly, and share the work that has already been done	1	3	3	K

5.2 Incident Log

Date	Risk	Incident	Action	Handler
03.02.2020	Problems with working environment	Leak in OZ headquarters	Leak prevented and house dried	OZ
25.02.2020 and 04.03.2020	Problems with working environment	Oz uses a lot of equipment so sometimes our room will suddenly fill with new tech.	We rearranged the setup.	Team
13.03.2020	Accidents, illness and other life events	COVID19 affects the whole society and RU closed down on the 15 of March.	We are still adjusting to changed circumstances.	Team
05.04.2020	Accidents, illness and other life events	Family members of team members got COVID19.	Other team members pick up the slack.	Team
16.04.2020	Product owner changes scope or requirements	We never got the API we were promised, communication was poor, undoubtedly because of the pandemic.	We used a dummy.	OZ/J/Team
21.04.2020	Accidents, illness and other life events	One team member got sick.	Other team members pick up the slack.	Team

6 Analysis

All functional and non-functional requirements of the project were turned into user stories and then collected into a product backlog. The user stories are based on four user interviews that were conducted in Sprint 1, note it is possible to access the interviews in the appendix. The user stories are prioritized and are of different sizes. For visualization we planned on meeting up in person since ideas can usually be greatly improved with input and ideas from others, but time and situations did not allow for that so the interviews were conducted online. Our Product Backlog was and is a product of both electronic and personal meetings of the team, by the end mostly electronic. The Product Backlog also includes technical tasks, details of problems that occurred and other tasks that needed to be done, for example regarding programming.

6.1 User Interview Analysis

6.1.1 Questions

Our user group will be employees of OZ and for a clearer vision of their needs and requirements we conducted four user interviews. We chose our respondents from different positions within the company to get a diverse vision of the product they need. The list of questions asked is as follows:

1. What is your name?
2. Please tell me about yourself and your relationship to OZ and the project?
3. What is in your opinion the most important task that should be possible to perform with the software?
4. What methods do you use today to perform those tasks?
5. Which part of the current method is most important to improve?
6. How important is a user-friendly and good looking front end to you?
7. How often do you think you will use the product?
8. What have we not asked you today that you think would be important for us to know?
9. Can we be in touch with you if further questions arise?

The interviews were done remotely since not all of the employees are stationed in Iceland.

6.1.2 Answers

All of the employees answered, though some in more detail than others. We filtered their suggestions and recommendations of tasks and improvements we should undertake to three groups, varying in importance. The groups vary from very important to nice to have tasks and improvements.

6.1.2.1 Very important

Three of the four employees that partook expected to use the software on a daily basis while one around 2-4 times a month. When asked which task would be the most important to be able to perform, speed, traceability and accessibility were key factors, as was being able to view available matches in thumbnail form and see which matches one had already downloaded.

Another very important improvement they all wanted was that the files are accessible, traceable and exportable with users controlling the quality of the footage. It should be easy to create clips and video files should be uncorrupted after handling.

6.1.2.2 Quite important

Questioned about current methods used to do what our product will be able to perform it was obvious that it is a tedious and time consuming process that is done by manual labor. It is also not obvious whether or not the user has already downloaded that video, which was noted by some that would make a great additional feature.

The importance of a user-friendly and good looking front end ranged from “...prefer efficiency and speed over good looks” to “Very important...”. Overall it seemed that a good overview would be nice but the most important thing was quick access, speed and efficiency.

6.1.2.3 Nice to have

What was considered not important but nice to have by the interviewees was being able to choose different camera angles to download, as well as being able to choose resolution, frame rate or format. Being able to store generated datasets onto the platform so other employees could work with it and also having access to a decent player on the software itself was also mentioned. Other points our interviewees mentioned were the ability to attach metadata to single frames, a log-in feature for data security, the possibility to download only certain parts of the match, such as VAR incidents only and choose between different audio sources.

To an even lesser importance one respondent mentioned the possibility to download only a portion of a match in order to save bandwidth and also the possibility to download single frames from the video. Audio feed was also mentioned as data that would be important to gather.

6.1.3 Conclusions

The user interviews proved to be incredibly helpful and fundamental in all our decision making concerning Product development and is the foundation of the Product Backlog. To see the full Interviews in all their entirety go to [Appendix B](#).

6.2 Product Backlog

The following is the Product Backlog for the project that is and will be a dynamic document during the project development. The list should be clear and exhaustive and address all elements of the project to help us keep track of its progress. In addition, it helps those who take on the project to continue with its development.

The list has a color code where each color represents the status of each story and describes as follows:

- White: Story that has been added to the list but implementation has not started.
- Yellow color: Story is being worked on.
- Green color: A story is considered ready and its implementation completed.
- Red color: The story has been removed from the list and is no longer included in the project.

Product Backlog					
Nr. story	Story	Priority	Story points	Sprint	Status
1.	As a user I can log in for data security	A	2	2	Done
2.	As a user I can use existing OZ authentication to log in	A	3	7	Done
3.	As a user I can easily and comfortably navigate the site	A	5	3	Done
4.	As a user I can see all available videos in a thumbnail view	A	3	3	Done
5.	As a user I can access videos from all camera angles	B	6		Planned
6.	As a user I can download full matches quickly and easily	A	7	7	Done

7.	As a user I can choose videos from different criteria (filter)	A	3	4/5	Done
8.	As a user all videos I have downloaded are marked as such	A	5	4	Done
9.	As a user I can search with keywords.	A	6	5	Done
10.	As a user I can play a selected video in a decent player on the webpage	B	6		Planned
11.	As a user I can download only a selected portion of the video	B	8		Planned
12.	As a user I can choose different resolution (e.g. 480p, 720p, 1080p, 4K), frame rate (e.g. 25 fps, 30 fps, 60 fps and format (e.g. .mp4, .mkv, .avi)	B	6		Planned
13.	As a user I can store generated datasets onto the platform	B	5		Planned
14.	As a user I can download a video frame (e.g. .jpg, .bmp, .tif, etc.) and give it a custom frame title	B/C	4		Planned
15.	As a user I can attach Metadata to each frame in a video (e.g. manual/automatic annotation, field, cameraId, zoom level and was broadcasted or not)	C	6		Planned
16.	As a user I can download different audio feeds (e.g. pitch, referee team, VAR)	C	8		Removed Note: not possible
17.	As a user I have an oversight of all videos I have downloaded	C	4		Planned
18.	As a user I can monitor who is downloading, traceability	C	7		Planned
19.	As a user if I don't allow cookies, not registered, the system throws me out (back to the log-in page)	C	5		Planned

6.4 Use Cases

Below are examples of use cases for the system.

6.4.1 Use Case 1

Name	The user has to log in to access data.
Number	1.
Priority	A
Pre condition	The user has an account / the user is an employee of OZ.
Main course	<ol style="list-style-type: none">1. The user goes to the site and is immediately prompted for login information.2. The user enters his/hers log-in information correctly.3. The user is moved to the front page.
Alt. course	At step 2. the user enters a wrong username or password.
Post condition	The user is logged in and can access the videos.
Requirement	N/A
Actors	User
Author	Gabríela Jóna Ólafsdóttir

6.4.2 Use Case 2

Name	The user can export matches in their entirety.
Number	2.
Priority	A
Pre condition	The user is logged in and can access the videos.
Main course	<ol style="list-style-type: none">1. The user sees a list of videos to download.2. The user selects one.3. The user pushes a button that says download.
Alt. course	At step 3 user sees he already has that video.

Post condition	The user can do as he/she pleases with the video and that video will be permanently marked (as downloaded) for that user.
Requirement	1, 4, 5, 8
Actors	User
Author	Gabríela Jóna Ólafsdóttir

6.4.3 Use Case 3

Name	The user can download matches that fit a certain criteria quickly and easily
Number	3.
Priority	A
Pre condition	The user is logged in and can access the videos.
Main course	<ol style="list-style-type: none"> 1. The user enters the criteria into the search bar 2. The user gets a list of matches that fit that criteria 3. The user downloads matches they need
Alt. course	<p>At step 2 there are no matches that fit said criteria</p> <p>At step 3 the user realized they have already downloaded all matches</p>
Post condition	The user has the videos they need and those videos will be permanently marked (as downloaded) for that user.
Requirement	3, 6, 7, 8
Actors	User
Author	Katrín Guðmundsdóttir

6.4.4 Use Case 4

Name	The user can easily navigate the site and view available videos in thumbnail format and see which videos they have already downloaded..
Number	4.
Priority	A
Pre condition	The user is logged in and can access the videos.
Main course	<ol style="list-style-type: none"> 1. The user sees available videos in thumbnail format 2. The user sees which ones they have already downloaded
Alt. course	<p>In step 1. No videos are available</p> <p>In step 2. The user has not downloaded any videos</p>
Post condition	The user has a clear view of available videos and the videos they have already downloaded
Requirement	3, 4, 8
Actors	User
Author	Katrín Guðmundsdóttir

6.4.5 Use Case 5

Name	The user can download parts of the video only and choose suitable format, resolution or frame rate.
Number	5.
Priority	B
Pre condition	The user is logged in and can access the videos.
Main course	<ol style="list-style-type: none"> 1. The user finds the video they need 2. The user selects a certain part, resolution etc. they need 3. The user downloads the portion of the video with the qualities they specified
Alt. course	At step 1 the user has already downloaded said video
Post condition	The user has the part of the video they need in the resolution/frame rate/format they wanted.

Requirement	10, 11
Actors	User
Author	Katrín Guðmundsdóttir

6.4.6 Use Case 6

Name	The user can download matches that fit a certain criteria quickly and easily
Number	3.
Priority	A
Pre condition	The user is logged in and can access the videos.
Main course	<ol style="list-style-type: none"> 1. The user enters the criteria into the search bar 2. The user gets a list of matches that fit that criteria 3. The user downloads matches they need
Alt. course	<p>At step 2 there are no matches that fit said criteria</p> <p>At step 3 the user realized they have already downloaded all matches</p>
Post condition	The user has the videos they need and those videos will be permanently marked (as downloaded) for that user.
Requirement	3, 6, 7, 8
Actors	User
Author	Katrín Guðmundsdóttir

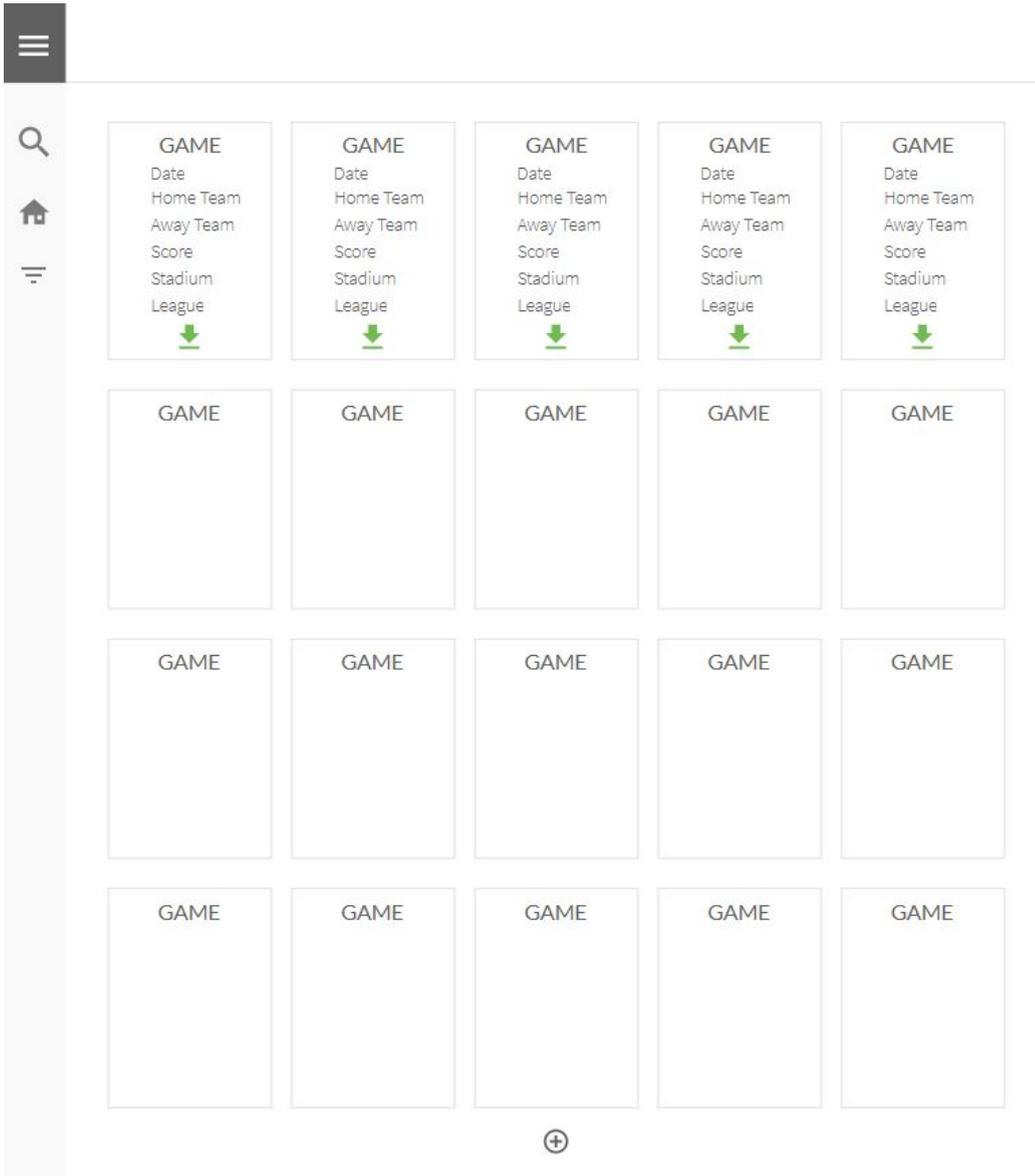
7 Design

7.1 Prototypes

We had total control over how we wanted to design the look on the front end but we are open to changes on it if OZ would like to change some features or looks on it. Decision was made on having the site simple and straightforward for employees of OZ to get the information that they need and not get some unnecessary information on the cards. Since the product will only be used on computers there was no need to adjust it to mobiles or tablets.

7.1.1 Design Draft

The focus for the main site (dashboard) was to have it simple, games are displayed as cards with useful labels in them to look at and get information about the game in each card. We designed a simple navigation bar with a simple search input, button for getting back to the main site and a filter search menu.



The filter design was made from the labels that are on the cards to make an accurate search for something specific in mind e.g. games in specific date range.

☰

Search...

DASHBOARD

FILTER

4/22/20 4/22/20

Home Team

Away Team

Stadium

League

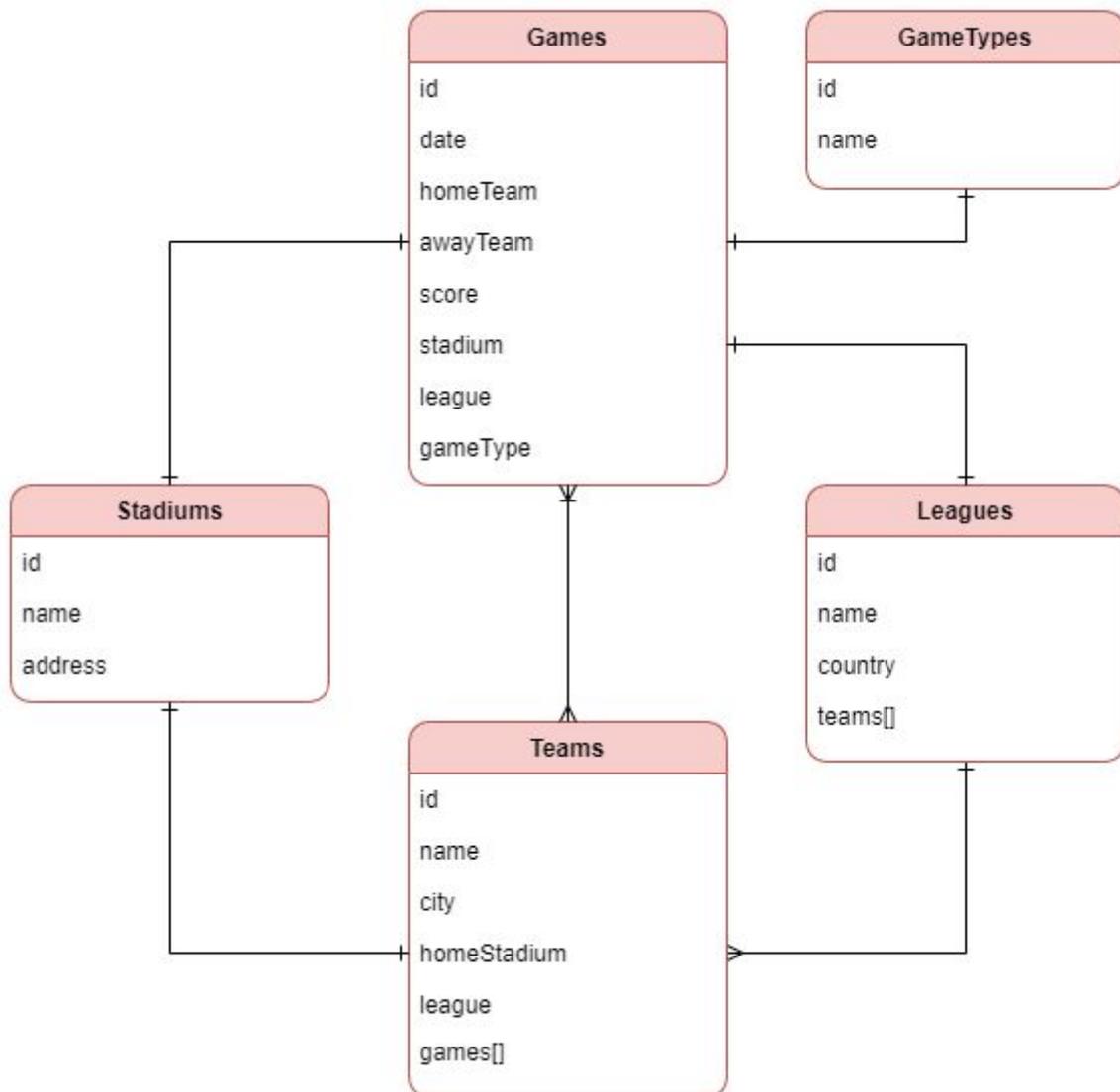
Button

GAME Date Home Team Away Team Score Stadium League 	GAME Date Home Team Away Team Score Stadium League 	GAME Date Home Team Away Team Score Stadium League 	GAME Date Home Team Away Team Score Stadium League
GAME	GAME	GAME	GAME
GAME	GAME	GAME	GAME
GAME	GAME	GAME	GAME



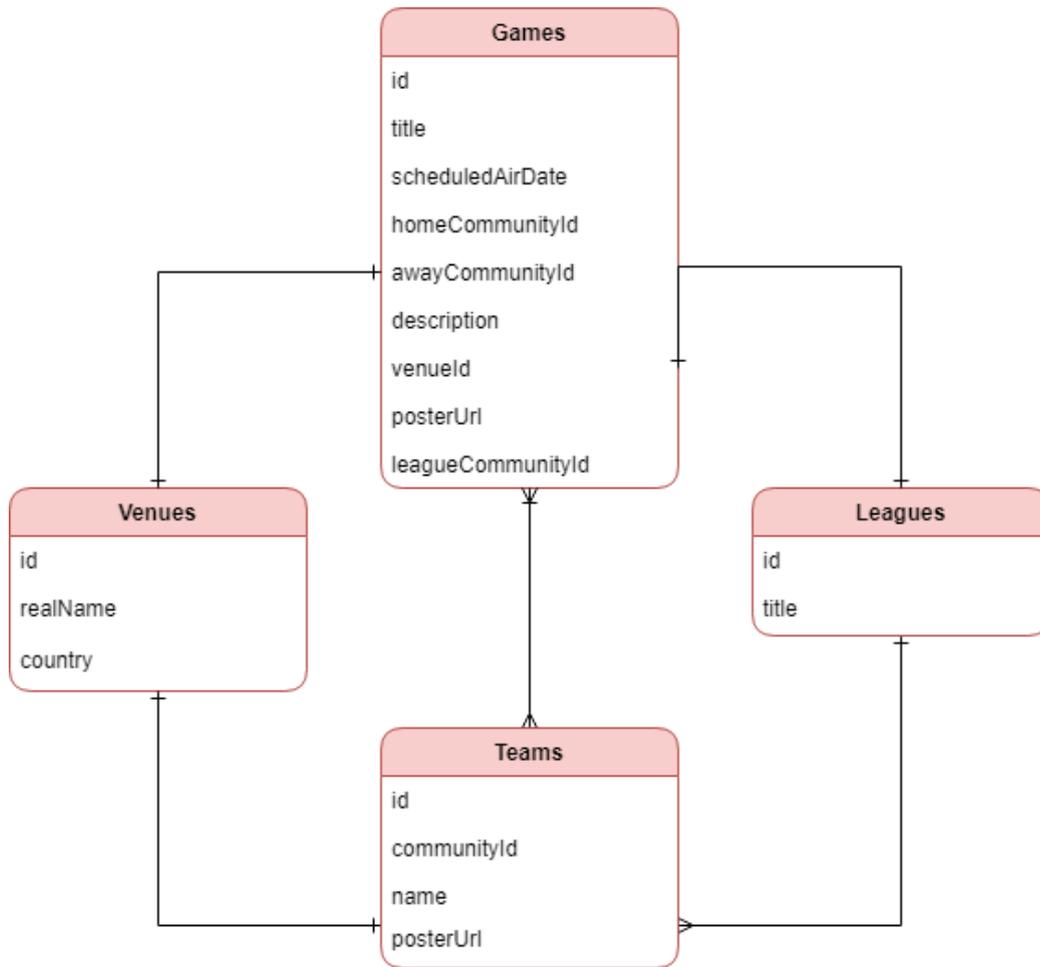
7.1.2 Database Design

We designed a database schema to use as guidance when we start designing and for the team to have a good overview of how the database will be set up, how data is stored and relations between tables in the database.



7.1.3 Updated Database Design

We mostly stuck to the original design, with only a few minor changes.



7.2 Final Design

This is the final design of what our product looks like. The final design is almost identical to how we made it first, minus a few minor changes that were made when new functions were added and new backlog items were added.

7.2.1 Opening window after login

The screenshot displays the 'All games' section of the OZ Games application. The interface includes a dark sidebar on the left with navigation options: Home, Filter, and Sign out. The main content area shows a grid of match cards. Each card features the logos of the competing teams, the match title, the date, the league name, and buttons for 'Download' and 'Details'.

Match Title	Date	League
Cibao FC vs Club 5 de Abril	2019-12-08	Liga Femenina de Fútbol
Bob Soccer School vs Delfines del Este	2019-12-07	Liga Femenina de Fútbol
FINAL VUELTA - Cibao FC vs Atlético Pantoja	2019-11-24	Liga Dominicana de Fútbol
FINAL IDA - Atlético Pantoja vs Cibao FC	2019-11-09	Liga Dominicana de Fútbol
Cibao Fútbol Club vs Atlético San Cristóbal	2019-11-02	Liga Masculina de Fútbol
Cibao Fútbol Club vs Atlético Fútbol Club	2019-10-27	Liga Masculina de Fútbol
Atlético Vega Real vs Atlético San Cristóbal	2019-10-26	Liga Masculina de Fútbol
3er. Lugar vs 2do. Lugar	2019-10-20	Liga Masculina de Fútbol

7.2.2 Searching: Cibao FC

The screenshot shows the search results for 'Cibao FC' in the OZ Games application. The search bar contains 'Cibao FC'. The results are displayed in a grid format, showing match cards for 'Cibao FC vs Club 5 de Abril' and 'FINAL VUELTA - Cibao FC vs Atlético Pantoja'. The sidebar on the left includes a search icon, a search input field, and navigation options: Home, Filter, and Sign out.

Match Title	Date	League
Cibao FC vs Club 5 de Abril	2019-12-08	Liga Femenina de Fútbol
FINAL VUELTA - Cibao FC vs Atlético Pantoja	2019-11-24	Liga Dominicana de Fútbol

7.2.3 Filter View

Home

Filter

Date from
mm/dd/yyyy

Date to
mm/dd/yyyy

Home Team

Away Team

Venue

League

Submit

7.2.4 List (Toggle) View

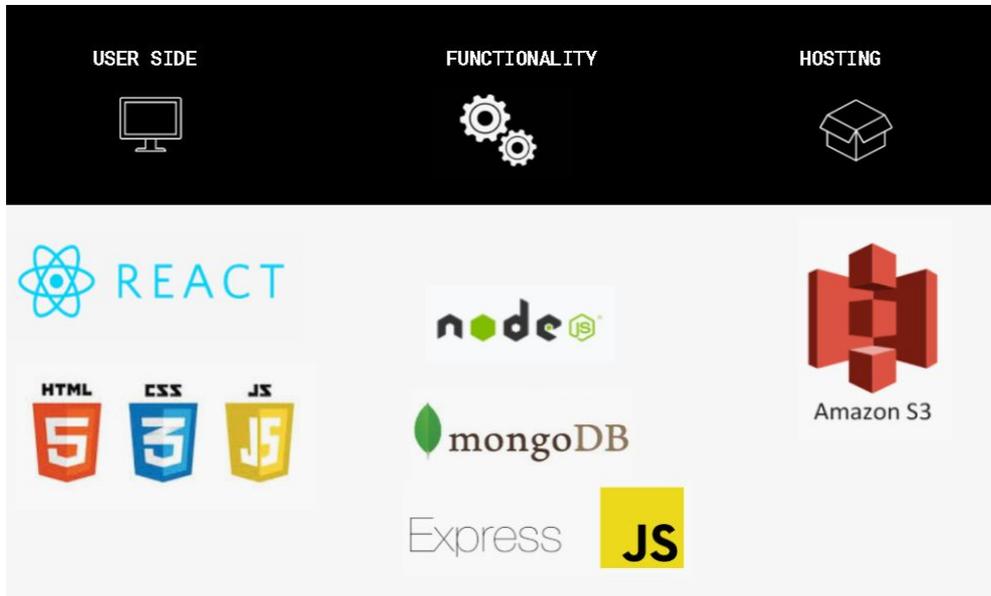
OZ Toggle View

Title	Date	Home Team	Away Team	Venue	League
Cibao FC vs Club 5 de Abril	2019-12-08	Cibao FC	Club 5 de Abril	Estadio Cibao Fútbol Club	Liga Femenina de Futbol
Bob Soccer School vs Delfines del Este	2019-12-07	Bob Soccer School	Delfines del Este FC	Estadio Olímpico Félix Sánchez	Liga Femenina de Futbol
FINAL VUELTA - Cibao FC vs Atlético Pantoja	2019-11-24	Cibao FC	Atletico Pantoja	Estadio Cibao Fútbol Club	Liga Dominicana de Fútbol
FINAL IDA - Atlético Pantoja vs Cibao FC	2019-11-09	Atletico Pantoja	Cibao FC	Estadio Olímpico Félix Sánchez	Liga Dominicana de Fútbol
Cibao Fútbol Club vs Atlético San Cristóbal	2019-11-02	Cibao FC	Atletico San Cristobal	Estadio Cibao Fútbol Club	Liga Masculina de Fútbol
Cibao Fútbol Club vs Atlántico Fútbol Club	2019-10-27	Cibao FC	Atlantico FC	Estadio Cibao Fútbol Club	Liga Masculina de Fútbol

7.3 System architecture

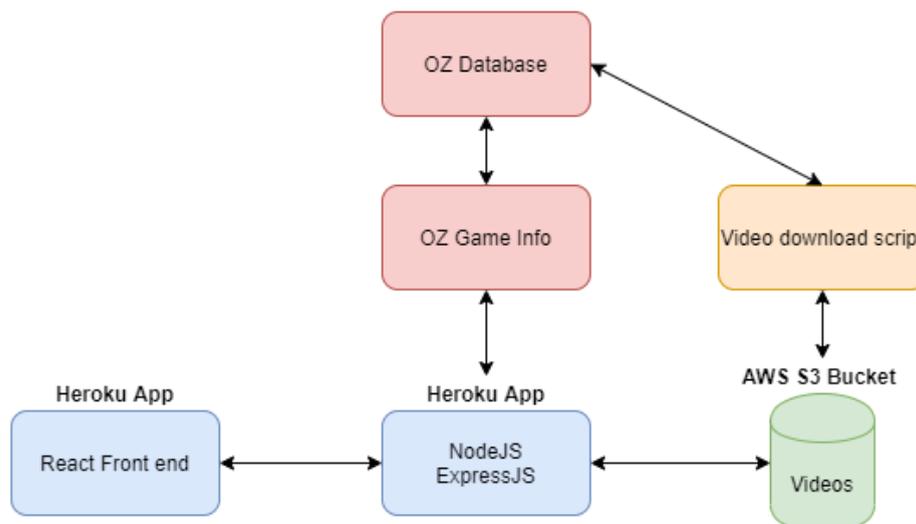
New footage is uploaded to an S3 bucket in Amazon Web Services (AWS) with scripts written in Python. MongoDB is used to structure and search the database and the website visual to the user is created within the React framework in JavaScript, CSS and HTML with NodeJs to run it.

7.3.1 Technical stack



7.3.2 System Design

The system was split up into three parts, using Heroku and AWS (Amazon Web Service). An Heroku app deploys a React frontend and a NodeJS server. The NodeJS server gets games info from an OZ database. A python script is run to download new games to the video database. The video database is hosted by an AWS S3 Bucket that has video files of games.



8 Progress Overview

8.1 Sprint schedule and events:

Sprint	Start	End	Est hours	Hand-in/Meeting	Deadline
0	09.01	23.01	80	- Detailed project description	20.01
1	24.01	06.02	120	- Methodology and Project Schedule Draft - Presentation	28.01 04.02
2	07.02	20.02	150	-Project status report -First status meeting	7-14.02
3	21.02	05.03	150		
4	06.03	19.03	150	-Project status report -Second status meeting	Week 9-10
5	20.03	02.04	150	-Project status report -Second status meeting	
6	03.04	16.04	150		
7	17.04	23.04	150		
8	24.04	30.04	200		
9	01.05	07.05	200	-Final project report -Final status meeting	Week 16-18
10	08.05	19.05	100	Final hand-in Presentation	15.05 18/19.05

8.1.1 Sprint 0

Summary

This sprint was mostly used to get to know each other, the project and the workplace. We decided on roles and work hours. Three of the team (one through Skype) met with the supervisor. Interviews with users have been scheduled for the next sprint and the Project Backlog is starting to take shape.

January 9th - January 23d		
Nr.	Task	Success
1.	Jóhann introduces the project	Yes
2.	Team is put together	Yes
3.	Meeting at OZ with the team to see the workplace	Yes
4.	Presentation at OZ about Machine Learning	Yes
5.	Meeting at OZ with the supervisor	Yes
6.	Hand in a detailed project description	Yes
7.	Create timesheets and tasks	No Note: We thought yes but this was done poorly
8.	Decide meetings and communication	Note: We created a temporary meeting schedule, but we will update later
9.	Create questions for the user interviews	Yes
10.	Decide on work ethics and environment	Yes
11.	Start the Project Schedule Draft	Yes
12.	Decide on tasks for next sprint	Note: We did this, but we didn't do it well enough
13.	Start on Project Backlog	Note: We did start, but we used none of it, changed the whole thing later

Retrospective

The team agreed that the first sprint had been successful although we could have met up a bit more often. No major problems occurred and the project vision is getting clearer. We all agreed that we had to be more active on Asana and start with the daily scrum next sprint.

Suggested improvements

Be more active on Asana and stick to the working agreement, plan meetings more efficiently and work on team building.

Sprint 0 burndown chart

Sprint 0



8.1.2 Sprint 1

Summary

We moved some tables and chairs to the room we were assigned in OZ headquarters. A lot of time went into working on the Methodology and Project Schedule Draft, work that we will benefit from later during the project process. We found 4 users to interview and did two of them. A prototype was created and the Risk analysis as well.

January 23d - February 6th		
Nr.	Task	Success
1.	Set up a working environment in OZ	Yes
2.	Work on and hand in Methodology and Project Schedule Draft	Yes
3.	Find users for interviews	Yes
4.	Do the user interviews	Note: We could not finish all the user interviews
5.	Create a prototype for the design report	Yes
6.	Set fixed working hours,	Yes
7.	Decide on suitable user stories for the Product Backlog	Yes
8.	Work on the Product Backlog	No Note: We created a Product Backlog that did not fill the requirements
9.	Conduct a Risk Analysis	Yes
10.	Work on the Progress overview	Yes
11.	Decide on tasks for Sprint 2	Yes

Retrospective

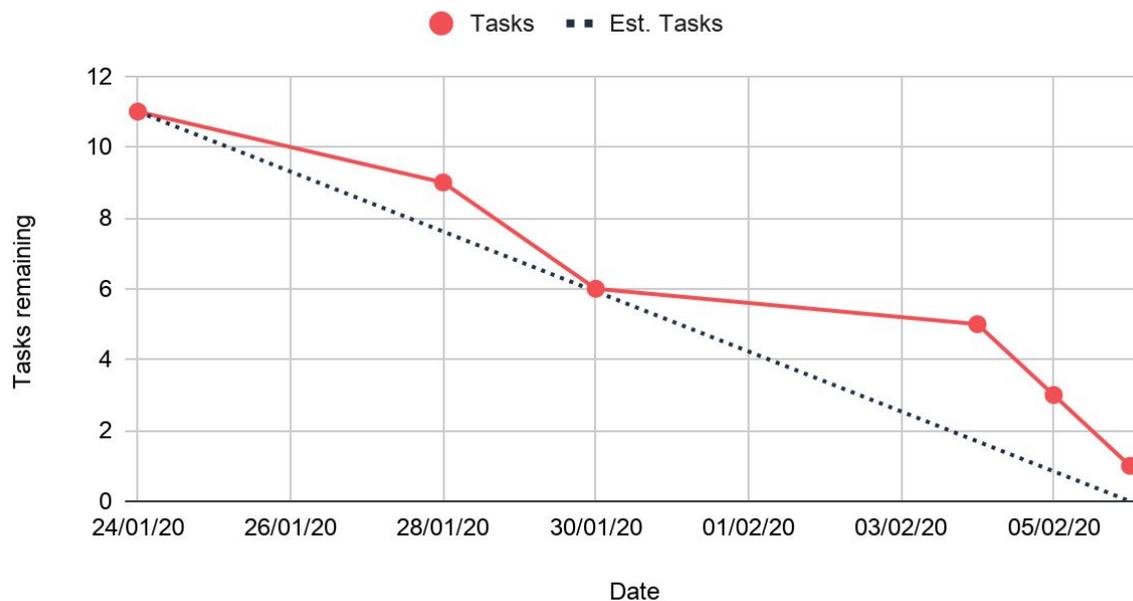
The team was quite happy with the first sprint although we would have liked to have done all the interviews and thus managed to create user stories and work more on the Product Backlog. This sprint was a bit scattered since 3 of 5 members were at some point abroad and 3 of 5 members were at some point sick.

Suggested improvements

Use the time more efficiently when we meet up and keep in touch with other team members when either sick or abroad, when possible. Pick up the slack when other members are unable to.

Sprint 1 burndown chart

Sprint 1



8.1.3 Sprint 2

Summary

The team finally got settled in well at Oz's headquarters. We finished all the user interviews, analyzed the answers and from that created the Product Backlog as well as a few use cases.

February 6th - February 20th		
Nr.	Task	Success
1.	Perform and finish interviews with selected users, workers from OZ that will be using the system	Yes
2.	Make a Product Backlog from the user interviews	Yes
3.	Start programming, set up a log-in site for the front-end webpage	Yes
4.	Make Use Cases from the Product Backlog	Note: We just started, we have two Use Cases, we will create more
5.	Prepare the presentation for the first Status-meeting (isl. Stöðufundur)	Yes
6.	Take notes from the teacher, fix what is wrong and set up a final order on the report	Yes
7.	Start setting up the database	No Note: We were not efficient enough, hence we made some changes to how we work
8.	Finish the Product Backlog	Yes
9.	Decide on tasks for Sprint 3	Yes

Retrospective

The team got a reality-check after the first Status meeting and adjusted working methods and the report according to criticism from the supervisors.

Suggested improvements

Divide tasks between team members and improve communication.

Sprint 2 burndown chart

Sprint 2



8.1.4 Sprint 3

Summary

The team started programming on the front end with good results. Suggestions were made in regards to the back end and database that needed authorisation from OZ so implementing those were moved to the next sprint. A lot of work was put into improving the Final report with points from our supervisors.

February 20th - March 5th		
Nr.	Task	Success
1.	Finish creating a functioning website (UI).	Yes
2.	A log-in page on the website.	Yes
3.	Create a NAV-bar.	Yes
4.	Add a search-bar for the website (searching by title).	Note: Still in Progress
5.	List/Overview of the videos (football games).	Yes
6.	(At first) Set up an online database.	No Note: Moved to sprint 4
7.	Connect the log in with an existing OZ authentication.	No Note: Moved to sprint 4
8.	Fill the database with at least 5 videos to work with.	No Note: Moved to sprint 4
9.	Start making an API to get the metadata from the videos in the database.	Yes
10.	Improve the final-report so that it is up to standards.	Yes
11.	Decide on tasks for Sprint 4.	Yes

Retrospective

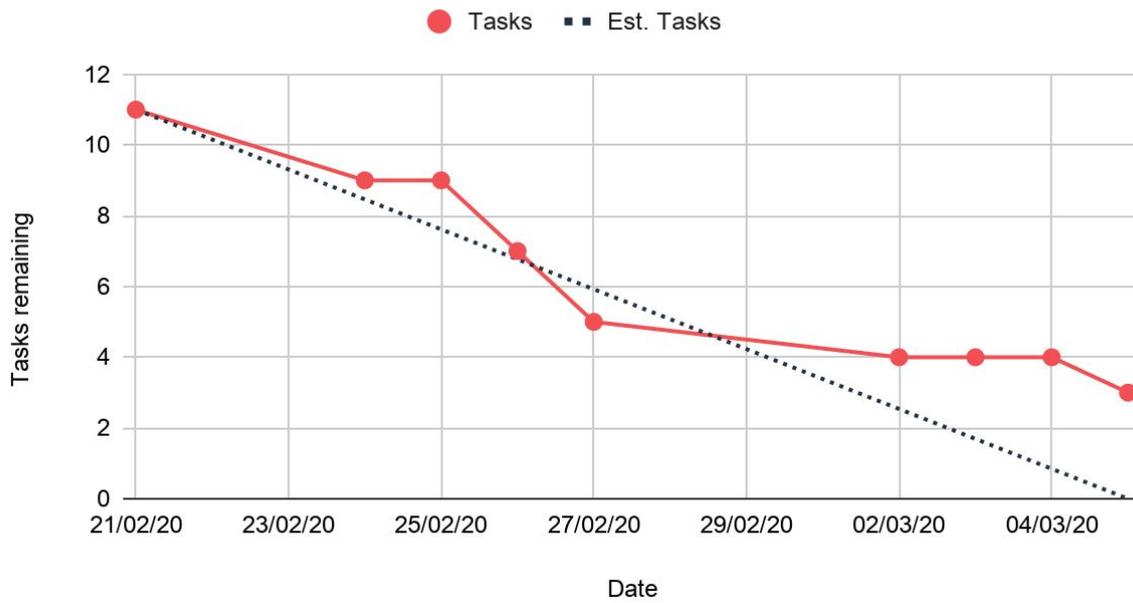
All members were happy with the look and functionality of the front end. We were disappointed that we had not managed to develop the database further.

Suggested improvements

Create a mutual vision of the project and make decisions on key elements regarding the database in collaboration with OZ.

Sprint 3 burndown chart

Sprint 3



8.1.5 Sprint 4

Summary

A lot of hours were spent stuck on a few problems during this sprint, especially setting up the connection between the database and interface. As well as the problems with our work, important meetings we had with members of Oz were postponed because of the pandemic COVID19. This was so far the most challenging sprint.

March 5th - March 19th		
Nr.	Task	Success
1.	Create a mutual vision for the final product.	Yes
2.	Work on Final report.	Yes
3.	Set up an online database.	Note: Still in progress, we had trouble with the connection (started with Django but moved to MongoDB)
4.	Connect the log in with an existing OZ authentication.	Note: Still in progress
5.	Fill the database with at least 5 videos to work with.	No Note: Not able to connect the database
6.	Meet with OZ directors to decide on a hosting database.	No Note: Meeting got cancelled
7.	Improve search filter.	Yes
8.	Improve Slide show.	Yes
9.	Practice presentation.	Yes

Retrospective

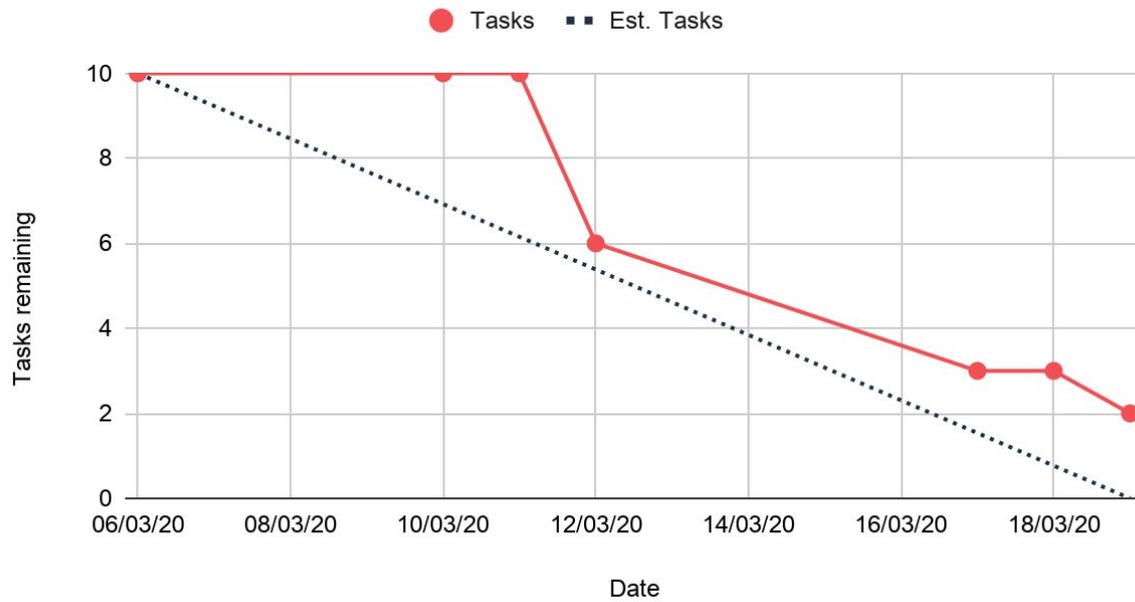
In retrospect we see that we should have been more worried and prepared about connecting the database with the interface. I do not think we could ever have prepared for the current pandemic afflicting the world, but we intend to carry on as unaffected by it as possible using positivity.

Suggested improvements

Learn to be more effective working remotely, with the development that happened to the world during the last Sprint, members need to find efficient ways to communicate and work from home.

Sprint 4 burndown chart

Sprint 4



8.1.6 Sprint 5

Summary

We spent most of this sprint trying to connect the front end to the S3 test bucket which proved quite time consuming since none of us had worked with that before. We were also still trying to decide whether or not to use MongoDB or Django to connect.

March 19th - April 2nd		
Nr.	Task	Success
1.	Set up a test bucket in AWS S3.	Done
2.	Download simple dummy files to work with.	Done
3.	Start working on the points Gunnar gave us about the report.	Done
4.	Create a login page.	Note: still in progress

Retrospective

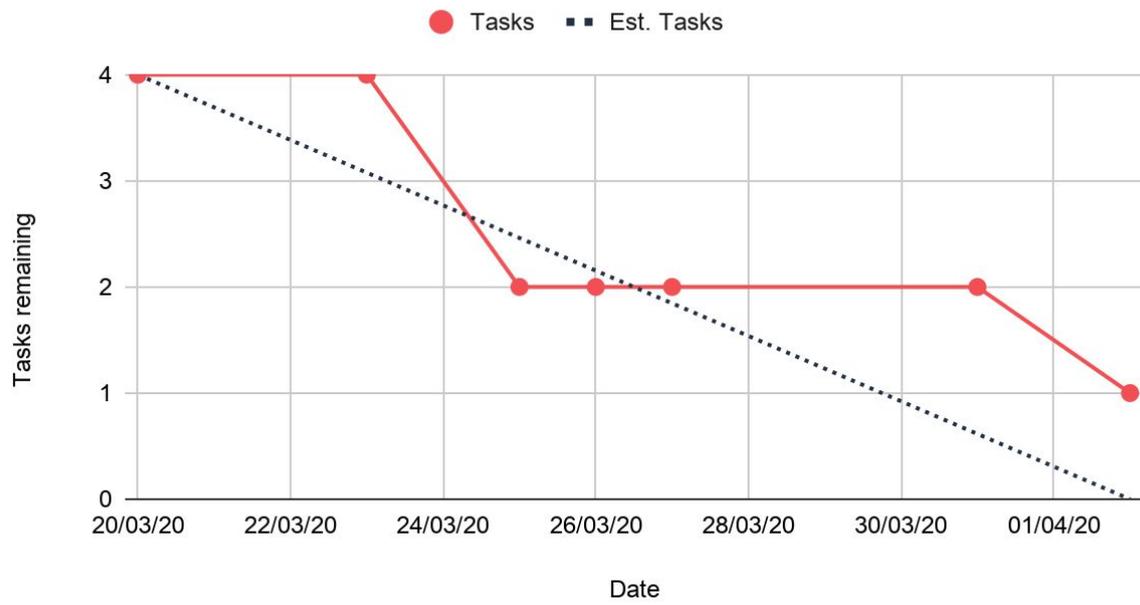
Communication could have been better and dividing assignments could have been clearer. Not being able to meet up and work together proved harder than we imagined.

Suggested improvements

Set up a weekday meeting plan and stick to it. Support each other as well and be both honest and forgiving.

Sprint 5 burndown chart

Sprint 5



8.1.7 Sprint 6

Summary

This sprint was a lot more successful than the previous one and the daily online meetings proved to be both motivational and productive. Work continued on the connection to the S3 bucket and the final report. A lot of work got done even though Easter holidays took up some time and a family member of one of us got infected and ill with the Coronavirus.

April 2nd - April 16th		
Nr.	Task	Success
1.	Design and create a login page.	Yes
2.	Set up a daily team meeting on Microsoft Teams, communicate every day.	Yes
3.	Get the download button to work, set up the connection.	Yes
4.	Make sure the login page only accepts employees from OZ.	Note: Still in progress
5.	Finish more than half of the points Gunnar gave us regarding the report.	Yes
6.	Create the actual host AWS S3 bucket.	No

Retrospective

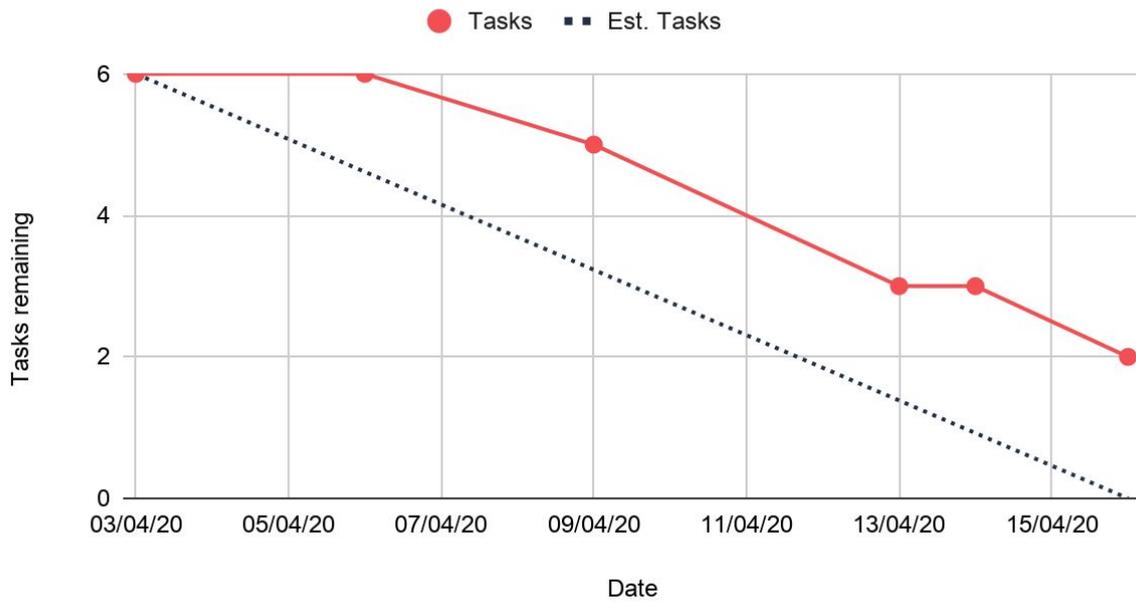
We really should have started the team meetings earlier. We are doing our best and being honest about having good and bad days.

Suggested improvements

Have two meetings per weekday. One in the morning and one at 6 o'clock in the afternoon for a better understanding of the process of the project.

Sprint 6 burndown chart

Sprint 6



8.1.8 Sprint 7

Summary

This sprint was mainly used to work on the log-in and the connection with the new actual S3 bucket. Downloading full video files was trickier than we expected.

April 16th - April 23rd		
Nr.	Task	Success
1.	Make sure the login page only accepts employees from OZ.	Yes
2.	Make the download button work for videos.	Yes
3.	Create a script that downloads all new videos to the S3 bucket.	Yes
4.	Two daily meetings.	Yes
5.	Finish all the points that Gunnar gave us regarding the report	Note: Mostly finished
6.	Create a check mark for downloaded videos.	No Note: Not started yet
7.	Create the actual host AWS S3 bucket.	Yes

Retrospective

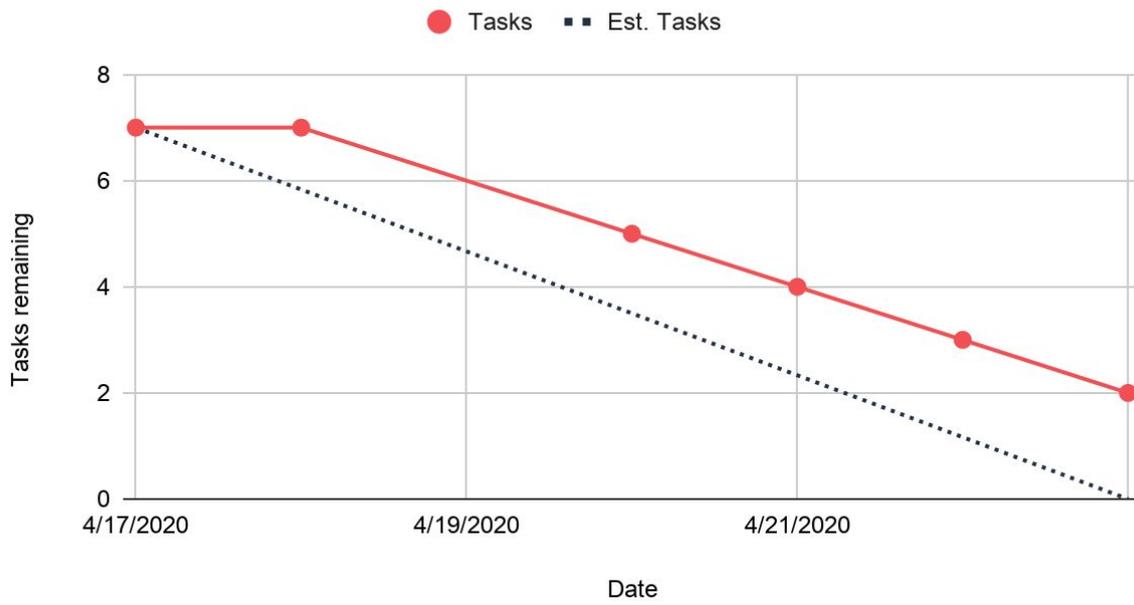
We should possibly have started work on the host bucket and connection earlier. The front end is nearly finished while a lot of work is needed in the back end.

Suggested improvements

Put a lot of focus on the back end and the connection between the front and back.

Sprint 7 burndown chart

Sprint 7



8.1.9 Sprint 8

Summary

This was possibly the most stressful and productive sprint of all. A lot of loose ends were tied up and minor bugs fixed. We refactored the code for readability and worked on the Final Report and presentation for Project Status meeting 3.

April 23rd - May 1st		
Nr.	Task	Success
1.	Work on the final project report, finish the vision chapter.	Yes
2.	Create a Final Product chapter in the report.	Yes
3.	Create a check mark for downloaded videos.	Note: Still in progress
4.	Create tests.	Note: Still in process
5.	Create a presentation for Project Status Report 3.	Yes
6.	Deployment pipeline.	No Note: Need tests to continue
7.	Create a Test chapter in the final report.	No Note: Need tests to do it
8.	Refactor (clean and comment) the code.	Yes
9.	Downloading method greatly improved.	Yes.

Retrospective

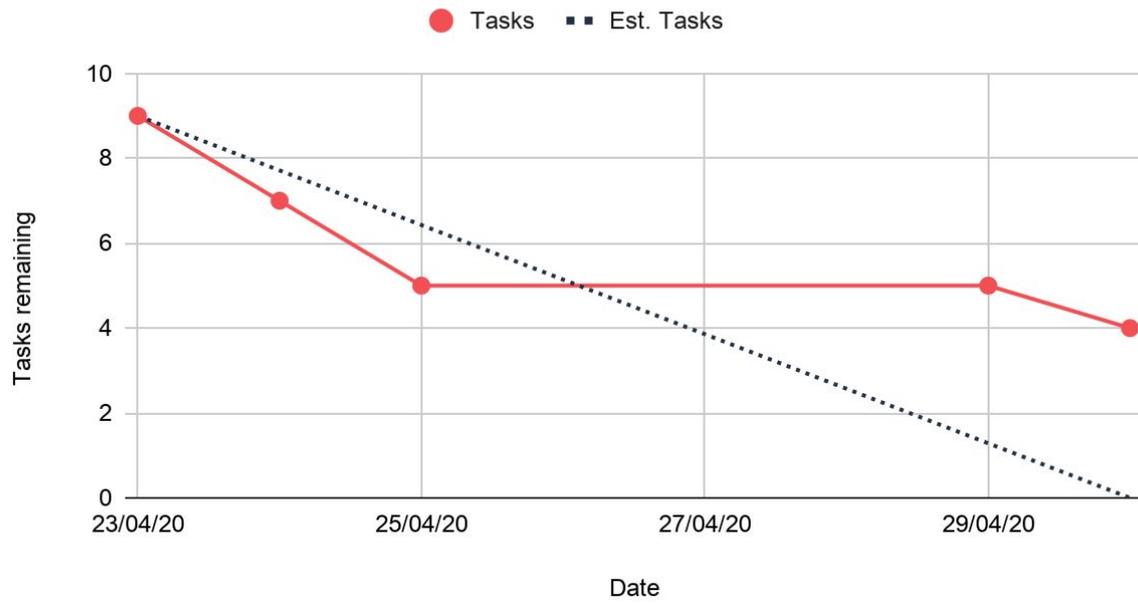
Was very productive and we worked well together. Morale was good but the days really need more hours.

Suggested improvements

We should have always worked like this. One should really always work like it is the last week before a deadline.

Sprint 8 burndown chart

Sprint 8



8.1.10 Sprint 9

Summary

Project Status meeting 3 ended and we had two weeks to go. We continued like in the week before to have two online meetings per day and in one of those meetings decided to hand in the product and the report a week early. The week went by smoothly and we look back at what we have created, thankful and with more wisdom than at the start of this project.

May 1st - May 9th		
Nr.	Task	Success
1.	Finish the final report and hand it in.	Yes
2.	Update the final presentation.	Yes
3.	Merge all branches on Github.	Yes
4.	Refactor the code.	Yes
5.	Fix the navigation bar.	Yes
6.	Hand in the product.	Yes

Retrospective

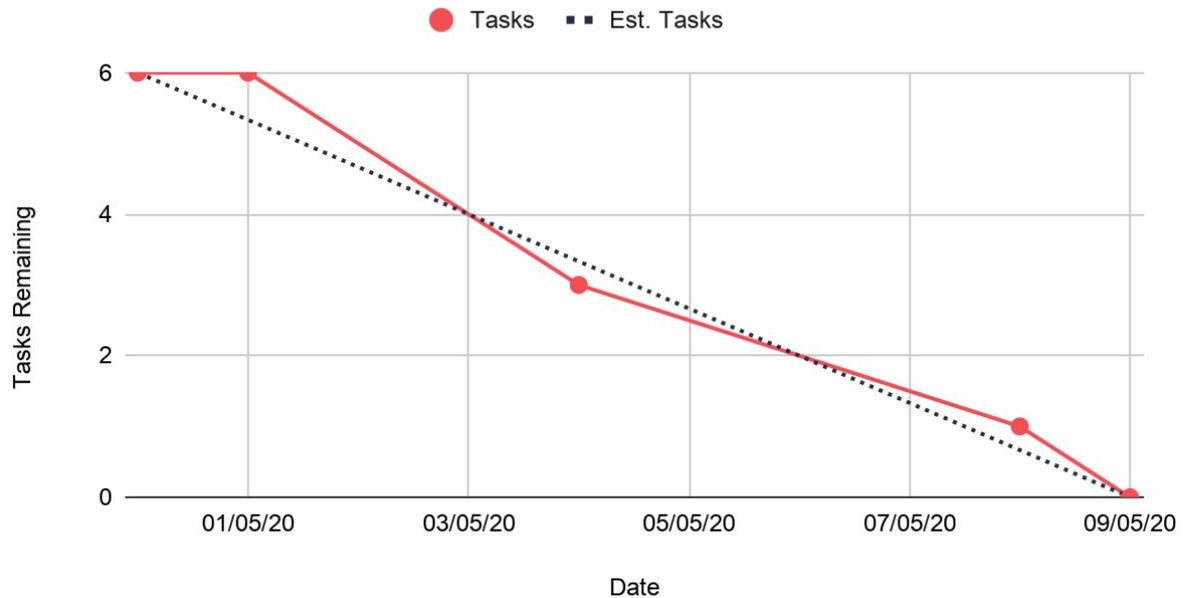
This sprint was efficient but relaxed. Since we decided to finish a week earlier, and that is one sprint before our last, we finished all the tasks we decided to finish and handed in our product and report at the end of it, working well and efficiently.

Suggested improvements

For this week (since we handed in the product and this report a week early, that is one sprint with one sprint to go) the only improvements we take with us are the ones we take in life. Upholding strong communication with future work colleagues and all team members is vital as well as

Sprint 9 burndown chart

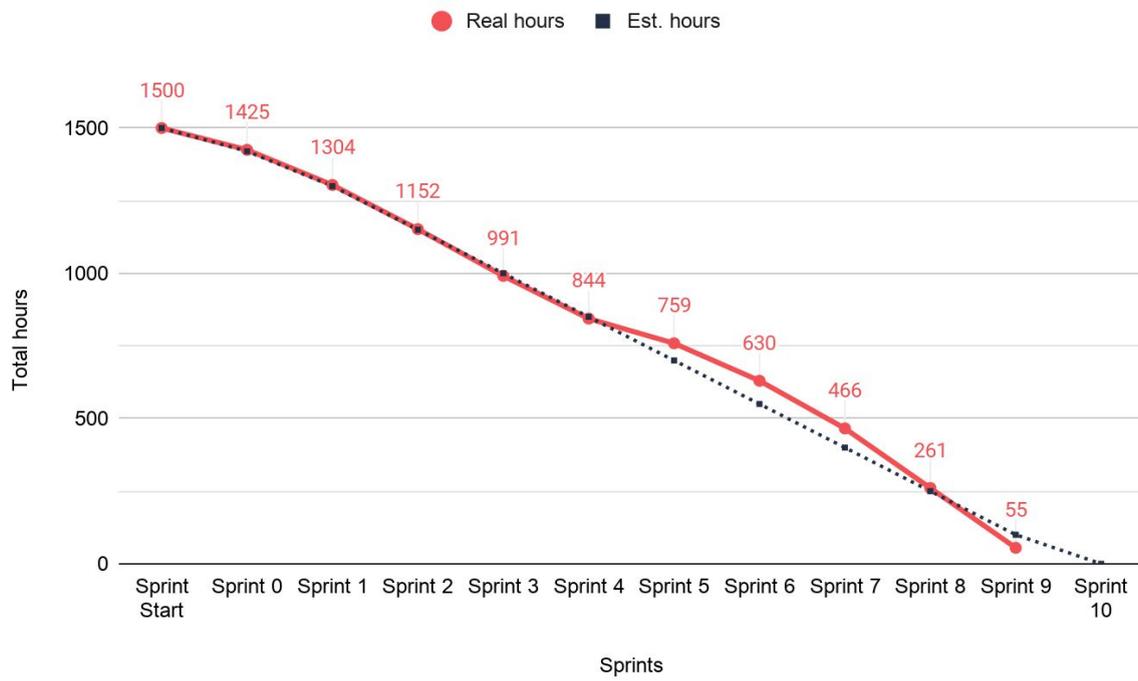
Sprint 9



8.2 Summary of Sprints

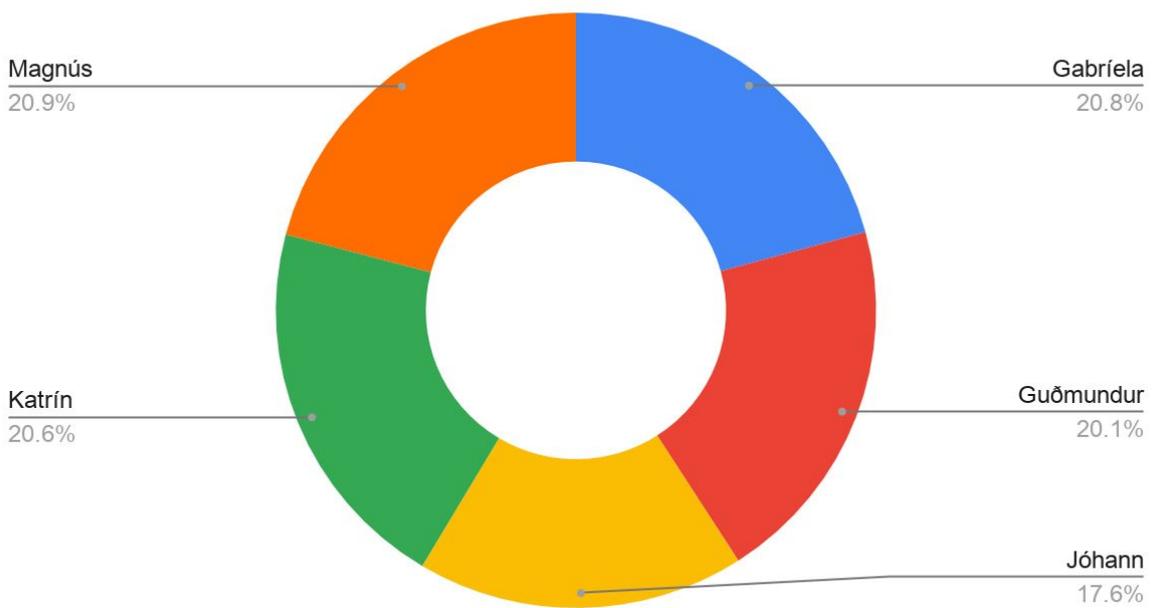
Working with a team is challenging, when we look at what we wrote most often in the suggested improvements communication was the factor that stood out the most. We are diverse and all think quite differently so that made it even more challenging, but it was a learning curve and we are very grateful for each other and this experience. We can also see that we overestimated our ability and our time, and underestimated the heaviness of some tasks. Again when looking at all the retrospects, communication (not just within the team but also with the company we were working for) really was our problem, we had a lot less contact with OZ than we had foreseen in the beginning of the project and thus had to rely on ourselves and each other.

8.2.1 Project burndown chart



8.2.2 Working hours per student

Hours spent per student



9 Product

9.1 Final product

The main goal of this project was to make sure that footage made by OZ is easily accessible to OZ employees and only them. Before we started this project anyone with the correct information could download the footage but it was scattered around different platforms and the files were all segmented and therefore took a very long time to download. There was also no way for the user to know if they had already downloaded the footage before. Users could not get a visual view on available material or search for material that fit a certain criteria.

We have managed to fix these problems and thus made it easier for Oz employees working on machine learning and other projects to access necessary material.

The footage is stored in an AWS S3 bucket and downloaded straight from there. S3 has authentication methods and without an access key ID and a secret access key it is impossible to connect to that hosting space.

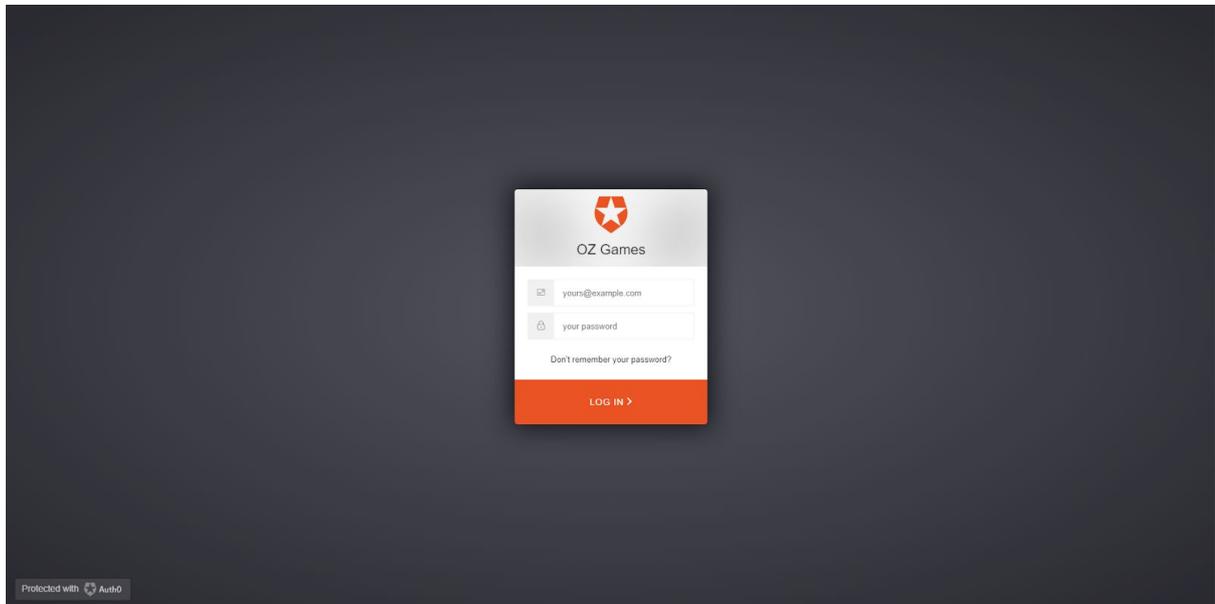
```
async function doDownload() -{
  try {
    AWS.config.setPromisesDependency();
    AWS.config.update({
      accessKeyId,
      secretAccessKey,
      region: 'eu-central-1',
    });
    const s3 = new AWS.S3();
```

All new footage manufactured by Oz is automatically downloaded to the bucket with a script written in Python so the system is always updated.

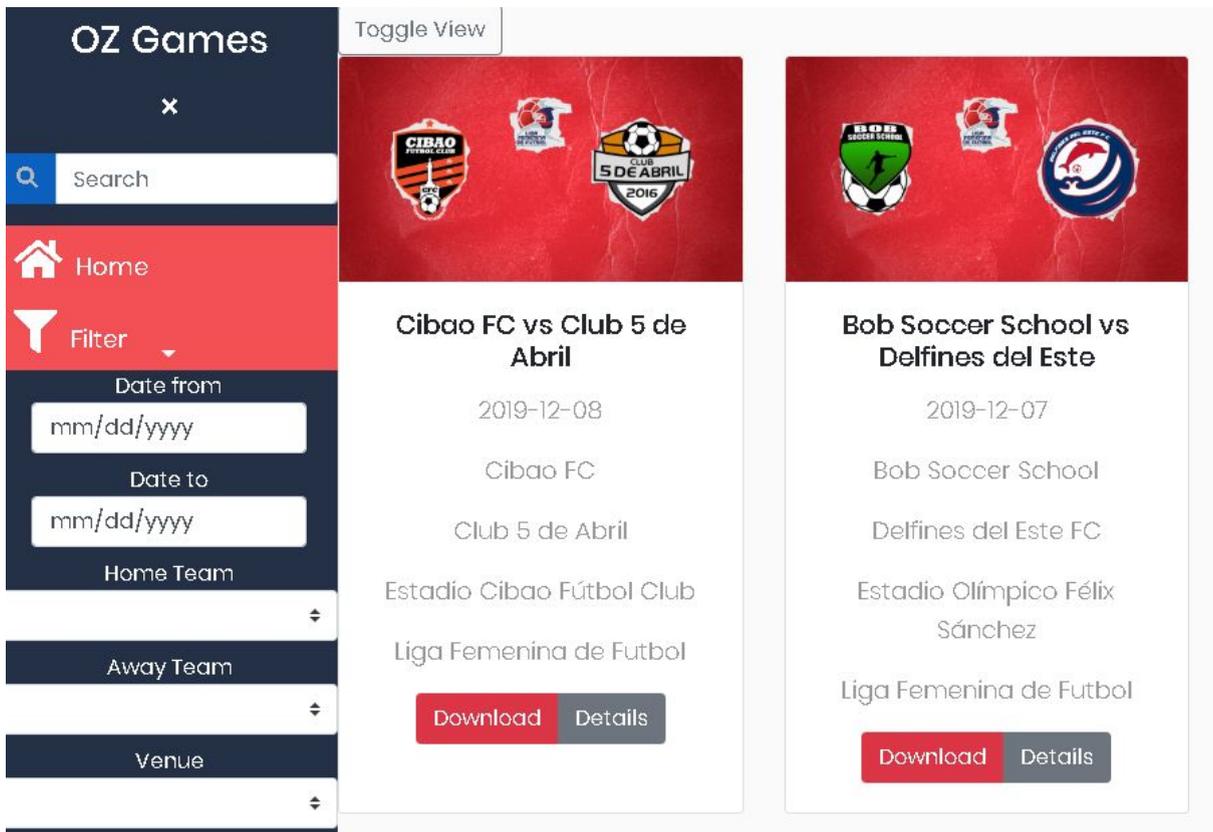
```
1 import urllib, json
2 import os
3 url = "https://core.oz.com/channels/FedofutbolGo/pastEvents"
4 response = urllib.urlopen(url)
5 data = json.loads(response.read())
6 i = 0
7 for p in range(len(data['data'])):
8     d = (data['data'][p]['id'])
9     f = 'ffmpeg -n -i https://playlist.oz.com/content/5cdbd6cd-095f-4d52-8353-b762e9cd5b30/' \
10     + d \
11     + '/1200/playlist.m3u8?ssl=true&version=1&token=REDACTED -c:a copy -c:v copy ./10min/' \
12     + d + '.mp4'
13     os.system(f)
14
15 os.system('python3 -m awscli s3 sync /home/ubuntu/10min s3://oz-johann-test/5cdbd6cd-095f-4d52-8353-b762e9cd5b30/')
```

9.2 Website features

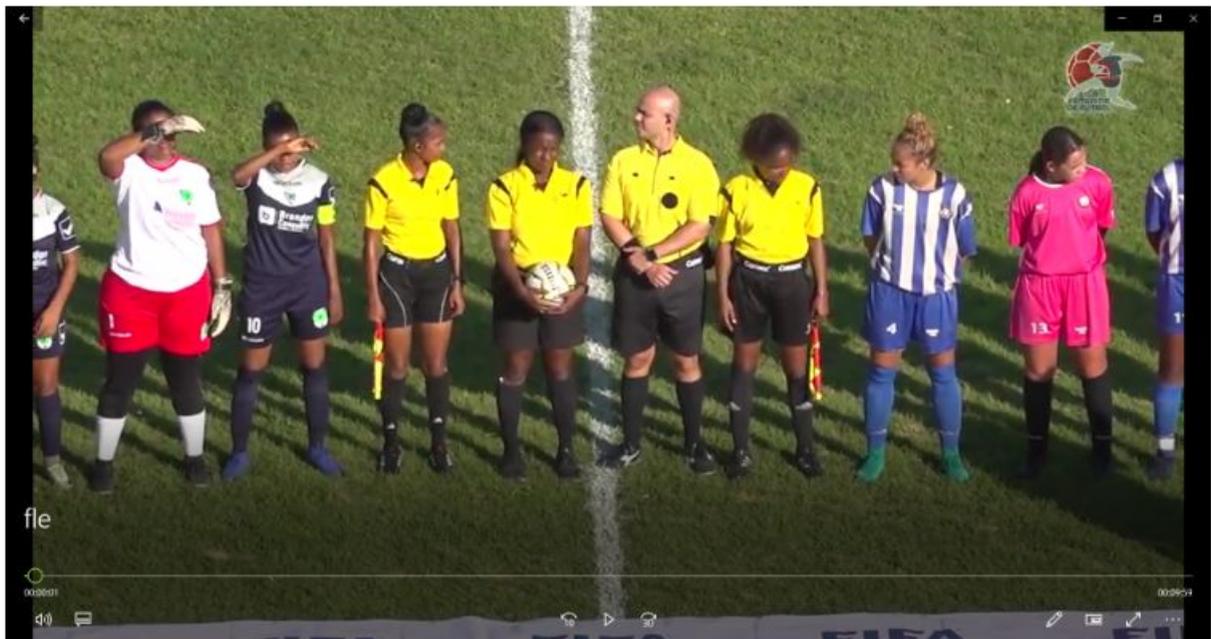
When the user enters the website they are required to log in with recognized credentials.



After the user has successfully logged in they see a page with a list of available footage. On the left sidebar there is a search bar where the user can enter relevant keywords and there is also a filter option with the criteria that was considered important as a result of our user interviews.



When the user has located the wanted material they press the download button and are prompted for location to save said material. After that the footage is ready to use.



9.3 Learning experience

The user interviews proved incredibly useful to visualise what the users want and need. We managed to diagnose the main features needed but at the same time we set out with far too many requirements and some of them proved impossible to achieve at this point.

We had a lot less contact with OZ than we had foreseen in the beginning of the project and thus had to rely on ourselves and each other. None of us had worked with AWS before so that was a steep learning curve. Most of our collaboration had to be done remotely because of Covid-19 pandemic which required self-discipline and motivation. We kept in contact with each other weekdays through meetings on Microsoft Teams to motivate and support each other.

The SCRUM methodology proved incredibly useful in these circumstances and so did having a clear mutual vision of the project. Not being able to meet up in person did prevent us from sharing our experience and knowledge and perhaps led to people taking on tasks they felt comfortable with instead of the more challenging ones. All in all this was a great learning experience in working on how to plan og execute a whole project from the beginning to the end and really appreciate the effort made by other team members.

10 What lies ahead

10.1 Problems encountered

There are a number of tasks we defined in the Product Backlog that we were unable to finish. We started setting up a deployment pipeline to have unit tests in the pipeline to prevent broken code being deployed but we encountered some issues with setting up a pipeline. But nonetheless an Heroku app is deployed when a change is pushed to GitHub.

10.1.1 Testing

Setting up unit tests would be very important for continuing the project to prevent that a broken code is released and give the person/s some security and overview of that functions are working and are correct.

10.1.2 Communication

Because of the current situation of the world today we lost most contact with OZ which proved to be fundamental to the project, especially in regards with API's and hosting. We had to provide our own solutions to problems we could have otherwise avoided.

10.2 Next steps

Now all of the employees can easily find footage and in a much more comfortable manner than before but as can be seen in the [Product Backlog](#) there are a number of steps to finish before the platform will be used for its intended purpose of serving as a comfortable ground where employees perform machine learning.

10.3 Vision

The future of the project is now completely in the hands of the employees of OZ sports. They will definitely use it and it will prove to be valuable for them in the future. They have everything they need to continue working on the product, we provided them with a technical operations manual so it should be easy for all of them to pick up where we left off. We kept our code clean and commented where we thought it would be necessary, so they shouldn't have any real issues. We also expressed eagerness to help so if any employees run into problems they have our contact information.

The B requirements in the Product backlog are doable for a more experienced programmer. Those requirements will be very beneficial for the project and the user.

11 Conclusion

We are very happy with the final product. We are satisfied with the functions although we would have liked to have developed and finished more of our requirements and we will have to keep in mind how much time can go into tasks that seem straight forward, as well as underestimating how much external factors could affect us and the development.

This has been a learning curve and we will take our newfound experience with us as we enter the unforgiving world that is the labor market. This project has helped us grow as programmers, as writers, as team members and as individuals. We are happy with the opportunity of working on this product and for OZ sports, as well as being very grateful for the support we have received from our instructor. It is always a challenging task working with and relaying on a team, especially in one where the members are so diverse and with the pandemic, but we are very grateful for each other and this experience.

12 Appendix A

12.1 Glossary

- AI:** Artificial Intelligence
- AWS:** Amazon Web Services
- DRM:** Digital rights management
- ML:** Machine Learning
- SCRUM:** An agile process framework

13 Appendix B

13.1 Interviews

Halli:

What is your name?

Haraldur Jóhannesson

Please tell me about yourself and your relationship to OZ and the project?

I'm the head of research and development.

What is in your opinion the most important task that should be possible to perform with the software?

Provide fast and easy access to all recordings to view or export.

What methods do you use today to perform those tasks?

Manual labor.

Which part of the current method is most important to improve?

Accessibility and exportability.

How important is a user-friendly and good looking front end to you?

Very important, especially fast quick access and fast loading videos and scrubbing.

How often do you think you will use the product?

Almost daily

What have we not asked you today that you think would be important for us to know?

Maybe specifics of UX.

Can we be in touch with you if further questions arise?

Sure.

Aðalsteinn:

What is your name?

Aðalsteinn Pálsson.

Please tell me about yourself and your relationship to OZ and the project?

I am a part time employee at OZ and serve as head of AI. I do this alongside a deep learning phd at RU. I will be a user of the project's end product.

What is in your opinion the most important task that should be possible to perform with the software?

In my opinion being able to easily extract content from the platform and doing it in a traceable manner, e.g. such that frames taken from the platform can be associated with their temporal context.

What methods do you use today to perform those tasks?

Currently this is done by extracting content via ffmpeg with a url taken from the platform. Usually prepared by Jóhann.

Which part of the current method is most important to improve?

The current method lacks an overview, traceability and is cumbersome to use. Also, having better control of the quality of the archived and extracted footage.

How important is a user-friendly and good looking front end to you?

An overview through a front end is definitely nice to have. The extraction could be done via command line.

How often do you think you will use the product?

On a daily basis when developing.

What have we not asked you today that you think would be important for us to know?

What would be very useful is to be able to attach metadata to each frame in a video. E.g. manual or automatic annotations, field, camera id, zoom level, was this footage broadcasted etc.

Can we be in touch with you if further questions arise?

Of course.

Paresh:

What is your name?

A: Paresh Kamble

Please tell me about yourself and your relationship to OZ and the project?

A: I have done my Ph.D. in Visual tracking of sports entities like ball and players. Currently, I am working remotely as a Machine Learning Specialist at OZ Sports from the Indian office situated in Nagpur city. I am working on the project of generating In-house datasets for OZ. Such datasets shall be used to train AI models for detection and tracking of ball, players and identification of key events during sports. Such trained AI models would also work as an assist for the remote camera operators and sometimes even let the cameras work in Auto-Pilot mode. For such a dataset we need to annotate the sequence of frames from recorded soccer matches available at the local server.

What is in your opinion the most important task that should be possible to perform with the software?

A: According to our need, we think that such a software should be able to:

- i) Show all the available match videos with a thumbnail view at a glance.
- ii) Play a selected video in a decent player on the webpage of the software itself.
- iii) Download any such video with any resolution (480p, 720p, 1080p, 4K), with any frame rate (25 fps, 30 fps, 60 fps) in any format (.mp4, .mkv, .avi).
- iv) If possible, download only a selected portion of the match video, instead of downloading the complete video and waste bandwidth.
- v) It would be great if we could directly download the video frames of any selected portion of the stored match videos in any (.jpg, .bmp, .tif, etc.) format with customised frame title (e.g. pilot_01_{0000}.jpg).
- vi) We should also be able to store the generated datasets onto the platform. Since it is the OZ's property, it should be available for employees of OZ to work upon.

What methods do you use today to perform those tasks?

A: Currently, we use the following approach for doing the above said tasks:

- i) We do not get to see the thumbnail of the match videos. We have to download it to see if we already have it or is a new one.
- ii) Play the video only after downloading or play only the partly downloaded part.
- iii) Currently, we use ffmpeg script to download such videos.
- iv) Currently, we download the whole video and separate the unwanted portion of the match manually.
- v) We manually separate out frames from them from another software.
- vi) Currently, it is stored in our Google Drive and we share the link with those who want to use it.

Which part of the current method is most important to improve?

A: Most of the time, the match video files are corrupted and not useful after downloading. A care must be taken to preserve the video files from getting corrupted.

How important is a user-friendly and good looking front end to you?

A: We would prefer efficiency and speed over good looks.

How often do you think you will use the product?

A: If such a software is available, we would use it once or twice in every 15 days.

What have we not asked you today that you think would be important for us to know?

A: I think there should be a log-in feature for the software. There should be monitoring of how many such videos are being downloaded and by whom. I think this is important in terms of data security and preventing data theft accidentally or deliberately.

Can we be in touch with you if further questions arise?

A: Sure

Villi:

What is your name?

Vilhjálmur Alvar Þórarinnsson.

Please tell me about yourself and your relationship to OZ and the project?

I am a Production Engineer and a FIFA International Referee soon to be certified VAR (Video Assistant Referee). I am the VAR Product Manager, making sure that the product is at the standard that is needed to be delivered to the market, making sure that we get market exposure and coming up with ways to improve our product.

What is in your opinion the most important task that should be possible to perform with the software?

To be able to access the full match after the games with all the camera angles, different audio feeds (pitch, referee team and VAR with RO) you can improve on previous matches by learning from previous matches. Either by improving communications or the VAR process in any given incident.

What methods do you use today to perform those tasks?

Currently we need to get each camera feed and upload it and sync it together.

Which part of the current method is most important to improve?

More easily create clips or upload full matches. Add audio functionality to the clips.

How important is a user-friendly and good looking front end to you?

Very important, persons using the software will most likely not be computer experts.

How often do you think you will use the product?

As a VAR project leader in a federation, most likely daily.

What have we not asked you today that you think would be important for us to know?

Having the option of getting different audio feeds and video feeds running.

*Recording of all audio feed

Ability to watch game with different audio feed (Match - Referee - VAR)

Recording of VAR Quad split screen

Recording of main camera

And availability of VAR Incidents with all cameras (2-4 minute clips)

Can we be in touch with you if further questions arise?

Sure!

14 Appendix C

14.1 Code example

```
1 import { data } from '../resources/teams.json';
2
3 const getTeams = (filter = { id: '' }) => data.filter((t) => (t.collection.communityId.toLowerCase().includes(filter.id.to
4
5 const getTeamById = (filter = { id: '' }) => data.filter((t) => (t.collection.communityId.toLowerCase().includes(filter.id
6
7 const getTeamName = (id) => {
8   const team = getTeamById({ id });
9   if (!team[0]) {
10    return 'Unknown';
11  }
12  return team[0].collection.name;
13 };
14
15 const getTeamImg = (id) => {
16   const team = getTeamById({ id });
17   if (!team[0]) {
18     return 'https://i2.wp.com/learn.onemonth.com/wp-content/uploads/2017/08/1-10.png?w=845&ssl=1';
19   }
20   return team[0].collection.posterUrl;
21 };
22
23 export {
24   getTeams,
25   getTeamName,
26   getTeamImg,
27 };
28
```

Above is a bit of code from the file Team services. It's quite simple and starts by getting the data from a JSON file, which will be replaced to a database. Then it has 3 functions, `getTeams`: gets all of the teams in the set, `getTeamName`: a function that returns the team name from team id. We send in a team ID from the match data and get a name to display. If there is no matching team in the data then the function returns `Unknown`. Then we have a similar function `getTeamImg` that returns the poster url for the requested team from ID. If there is no matching team in the database we return a template photo.

```

-- async function doDownload() {
--   try {
--     AWS.config.setPromisesDependency();
--     AWS.config.update({
--       accessKeyId,
--       secretAccessKey,
--       region: 'eu-central-1',
--     });

--     const s3 = new AWS.S3();

--     const urlExpire = 60 * 5;
--     const url = s3.getSignedUrl('getObject', {
--       Bucket: 'oz-johann-test',
--       Key: `${channelId}/${id}.mp4`,
--       Expires: urlExpire
--     })
  }
}

```

This is a part of the download function used to download the videos. We decided to use the method of getting a signed URL from the bucket and download with that method since it gives the user additional information and more control over the access to the content.

15 Appendix D

15.1 Time Logs

Sprint 0:

Member	Working hours	Estimated hours	Notes
Gabriela	15	16	
Guðmundur	15	16	
Jóhann	15	16	
Katrín	15	16	
Magnús	15	16	

Was a small setup sprint and was done in three 5 hour days. We managed to complete the tasks within that time

Sprint 1:

Member	Working hours	Estimated hours	Notes
Gabriela	26	24	Extra presentation prep
Guðmundur	21	24	Was out of country 1 half day
Jóhann	24	24	
Katrín	26	24	Extra presentation prep
Magnús	24	24	

Another small sprint working on presentation and reports. Was done in four 6 hour days.

Sprint 2:

Member	Working hours	Estimated hours	Notes
Gabriela	30	30	
Guðmundur	32	30	
Jóhann	29	30	
Katrín	30	30	
Magnús	31	30	Was sick 2 days

Sprint 3:

Member	Working hours	Estimated hours	Notes
Gabríela	32	30	
Guðmundur	34	30	
Jóhann	30	30	
Katrín	32	30	
Magnús	33	30	

Sprint 4:

Member	Working hours	Estimated hours	Notes
Gabríela	29	30	
Guðmundur	30	30	
Jóhann	28	30	
Katrín	29	30	
Magnús	31	30	

Sprint 5:

Member	Working hours	Estimated hours	Notes
Gabríela	19	30	
Guðmundur	18	30	
Jóhann	10	30	
Katrín	18	30	
Magnús	20	30	

Sprint 6:

Member	Working hours	Estimated hours	Notes
Gabríela	25	30	
Guðmundur	27	30	
Jóhann	24	30	
Katrín	26	30	

Magnús	27	30	
--------	----	----	--

Sprint 7:

Member	Working hours	Estimated hours	Notes
Gabríela	35	30	
Guðmundur	32	30	
Jóhann	30	30	
Katrín	33	30	
Magnús	34	30	

Sprint 8:

Member	Working hours	Estimated hours	Notes
Gabríela	46	30	
Guðmundur	41	30	
Jóhann	30	30	
Katrín	43	30	
Magnús	45	30	

Sprint 9:

Member	Working hours	Estimated hours	Notes
Gabríela	43	30	
Guðmundur	41	30	
Jóhann	35	30	
Katrín	45	30	
Magnús	42	30	

Sprint 10: Not finished

Member	Working hours	Estimated hours	Notes
Gabríela	-	20	
Guðmundur	-	20	
Jóhann	-	20	

Katrín	-	20	
Magnús	-	20	