Tour Guide Translator

Final report

Team:
Bjartur Fannar Stefánsson, 020695-3079
Gunnar Birnir Ólafsson, 280294-2179
Ævar Þór Gunnlaugsson, 090594-3119

Teacher assistant:
Fanney Sigurðardóttir

Test examiner:
Guðlaugur Stefán Egilsson

Table of contents

1. Introduction

Tourism in Iceland has never been bigger, and last year over 2.000.000 tourists came to iceland. Tourism is responsible for nearly 30 percent of Iceland's export revenue. All of these tourist want to go on the road and explore and see the country. A popular way to do that is via bus trips around the country with a tour guide. Where the tour guide talks in one language and the tourist listen. But what if the tourist don't understand him very well. That's where we come in with our app. Tour Guide Translator is an app that allows tour guides to talk in their language and the tourist can listen to them in their language in real time. Tour guide creates a room for a certain trip and the tourist join the room. Tourists can also send questions to the tour guide in their language and the questions will be translated to the language of the tour guide on his screen to read. The app makes sure that every tourist in the trip will be able to understand the tour guide.

2. Procedures

We decided to work according to the Scrum methodology. We had a workspace at Origo and we had two days a week where we work together at Origo, on Mondays and Thursdays. We had a meeting with the teacher assistant on Mondays and there were regular meetings with our Origo representatives on Fridays after each sprint.

2.1 Scrum

2.1.1 Role and responsibility

- **Scrum master:** Bjartur Fannar Stefánsson.
- **Product owner:** Soffía Kristín Þórðardóttir.
- **The Team:** Bjartur Fannar Stefánsson, Gunnar Birnir Ólafsson and Ævar þór Gunnlaugsson.

2.1.2 Sprints

We had 7 sprints, one 3 week sprint, three 2 weeks sprints and three 1 week sprints. The reason for the 3 week sprint was because we knew that we would miss one week out of that sprint so we decided to make it 3 weeks long. We decided to have one week sprints in the final 3 weeks of the semester because then we had a lot of time and had finished all the other courses.

1. 29. January - 9. February (2 weeks)
2. 12. February - 23. February (2 weeks)
3. 26. February - 16. March (3 weeks)
4. 19. March - 30. March (2 weeks)
5. 23. April - 27. April (1 week)
6. 30. April - 4. May (1 week)
7. 7. May - 11. May (1 week)

2.1.3 Sprint planning

Each sprint started with a sprint planning meeting. All members from the team met and discussed the goals of the next sprint. They picked the requirements from the product backlog that they wanted to finish in this sprint. Each requirement was broken down into tasks and given story points. Story points are the scale of the task in time units. Then the tasks where put in the sprint backlog.

2.1.4 Daily scrum

Each day in the sprint the team met in the beginning of the day and discussed what they were doing yesterday, what they are going to do today and if anything stands in their way of being able to solve their task.

2.1.4 Sprint review and sprint retrospective

After each sprint the team had a sprint retrospective. There we discussed what we wanted to continue doing, stop doing and start doing for the next sprint. We then had a sprint review meeting with our Origo representatives and showed them what we had accomplished in the sprint.

2.2 Documentation

We used Trello to keep track of our sprint backlog, what was next in, what was on going and what was done, all the documents were kept on Google Drive where all the members of the team could access them. There we stored all our reports the sprint burndown charts and our calendar.

3. Risk analysis

In this analysis we listed all the possible risks we could encounter while working on this project. Each risk was evaluated based on probability and impact. We then prioritized the list by probability and impact score. We also came up with a prevention and reaction strategy to the risks.

Priority	Probability	Impact	Risk	Prevention	Reaction	Handler
15	5	3	Workload in other courses	Good planning	Work harder the next week	Gunnar
12	3	4	The translator does not work properly for icelandic	Look for solutions in the API	Stop speaking icelandic	Ævar
5	1	5	Google shuts down firebase or google cloud service	Nothing	Switch to another service	Bjartur
5	1	5	Origo goes bankrupt	Nothing	Finish the project on our own	Gunnar
4	2	2	Some of our computers stop working	Take good care of our computers	Buy or borrow a computer	Ævar
4	1	4	Github goes down	Nothing	Use another service	Ævar

4	2	2	Meetings with TA and project owner get canceled	Plan meetings in advance	Plan another meeting	Bjartur
3	1	3	Someone from the team gets too sick to work	Take good care of ourselves	The other members of the team have to do more work	Gunnar
3	1	3	Expo goes down	Nothing	Switch to a regular React Native project	Gunnar
2	2	1	Reykjanesbraut is closed	Nothing	Work from home	Ævar

Of these risks only two actually happened. The first one “Workload in other courses” was almost guaranteed to happen and we handled it well, by planning and managing our time wisely. The second one, “The translator does not work properly for icelandic”, became a reality in the sense that when the tour guide speaks in English the service does not understand Icelandic words like Gullfoss or Geysir. We solved this by switching over to the Google speech-to-text service, so now the tour guide can speak in Icelandic the whole time.

One problem we did not anticipate in the risk analysis was that some React Native packages don’t work with Expo. This was a big problem when implementing the speech-to-text functionality but we solved it in the end by implementing a Heroku service which converts the audio files to the correct format and makes a HTTP request to Watson, a speech-to-text service from IBM. Like mentioned above, we then later switched to a service from Google, since it offers more languages, including Icelandic.

4. Work plan

4.1 User groups

User group	Background	System usage	Environment	Main user goals
Tour guide **Importance:** Very important	**Age:** 20-70 year old **Gender:** All genders **Education:** High School and higher **Abilities/ disabilities:** Nothing special **Computer knowledge:** Bad - very good	**Usage:** All year around **Training:** Has had former training **Attitude:** Users mostly have a good attitude for the app, as it helps them communicate with tourists **Number of users:** ~200	**Technical environment:** Will be on the phone and it has to be connected to the internet **The real environment:** The system will be used in cars on the highway with slow internet connection	To make all the tourist understand what he is saying in the trip Answer questions that the tourist ask
Tourist **Importance:** Very important	**Age:** 10-90 year old **Gender:** All genders **Education:** Elementary School and higher **Abilities/ disabilities:** Nothing special **Computer knowledge:** Bad - very good	**Usage:** Once a year **Training:** None **Attitude:** Users mostly have a good attitude for the app, as it helps them understand the tour guide better **Number of users:** ~2000	**Technical environment:** Will be on the phone and it has to be connected to the internet **The real environment:** The system will be used in cars on the highway with slow internet connection	Listen to the tour guide and understand what he is saying Asking the tour guide questions

4.2 Product backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tour guide can sign in	**3**	**DONE**
2	**A**	Tour guide can create a room and name it and code will be generated for the room	**8**	**DONE**
3	**A**	Tourist can join a room by entering the room code	**3**	**DONE**
4	**A**	Tourist can join a room by scanning the room QR code	**3**	**DONE**
5	**A**	Tour guide talks in a mic in a selected language and the app displays the words	**13**	**DONE**
6	**A**	Tourist can get translated messages from the tour guide	**8**	**DONE**
7	**A**	Tour guide can write a text and send it to the room he's in	**3**	**DONE**
8	**A**	Tourist can see the text from the tour guide	**3**	**DONE**

9	**A**	Tourist can send a question to the tour guide	**2**	**DONE**
10	**A**	Tour guide can speak an answer to a question that a tourist sends	**3**	**DONE**
11	**A**	Tourist can see back in time what has been said in the trip	**3**	**DONE**
12	**A**	Tourist can choose if he wants to listen to the translation or not	**2**	**DONE**
13	**A**	Tourist can set a nickname in a room which the tour guide will see when they ask a question	**3**	**DONE**
14	**A**	Tourist can choose the language that they want listen to the tour guide in	**2**	**DONE**
15	**A**	Tour guide can push a button when the trip is over and close the room	**2**	**DONE**
16	**A**	Tourist can leave the room he's in	**2**	**DONE**
17	**A**	Tourist can change the language that he listens to the	**3**	**DONE**

		tour guide in inside the trip		
18	**A**	Tour guide can changed the language he speaks in inside the trip	**3**	**DONE**
19	**A**	Tour guide can change his name and logout	**3**	**DONE**
20	**A**	Tourist can see the QR code for the trip he's in	**2**	**DONE**
21	**A**	Tour guides can see the QR code for the trip he's in	**2**	**DONE**
22	**A**	Tourist can listen to the translated message from the tour guide	**8**	**DONE**
23	**B**	Tourist can ask a question in their language and the tour guide sees the question in his language	**3**	**DONE**
24	**B**	Tour guide can mark questions as answered if the same question comes up many times	**2**	
25	**B**	Tour guide can display and answer questions in the translation view	**3**	

26	**B**	Tourist can listen live and untranslated to the tour guide	**3**	
27	**B**	Tourist can choose a language for the user interface	**8**	
28	**B**	Tourist can see on a map where he is	**5**	**DONE**
29	**B**	Tour guide can select to speak or use a script (snippets)	**8**	
30	**B**	Tour guide can edit the generated text if he sees an error	**3**	**DONE**
31	**B**	Tour guide gets a list of words that for each word he talk which he can change to	**5**	
32	**B**	Tour guide can see translations on other languages than polly can speak.	**8**	
33	**B**	Tour Guide can see how many people are in the trip	**2**	**DONE**
34	**B**	Tour guide can select a template for the trip to get more information about it	**8**	

35	**B**	Tour guide gets notification when a new message is received from the tourist	**5**	**DONE**
36	**C**	Tour guide can select a location he wants to attach the text to and it will be visible to the tourist	**13**	
37	**C**	Tour guide can stop speaking and the text will be sent automatically	**5**	
38	**C**	When a trip is over the room will close on itself if the tour guide forgets to close the room	**3**	
39	**C**	Tourist can not go into a room if he’s not within a certain radius of the tour guide	**5**	
40	**C**	Tourist can close the room and all data in the room will be saved to a database	**8**	

5. Design

We drew up wireframes for some important screens in our app. The first section shows two screens for the guide profile. The second two sections show the trip interface for guide and tourist. We also made a tech stack picture that shows what happens in the app when certain events are executed. At the end of the section we show our database design and rules.

5.1 Wireframes

Guide Profile

When the tour guide is logged in he sees a list of all his trips and he can also create a new trip. That trip will then be displayed in the “My Trips” list.

Guide Trip

When a tour guide selects a trip from the “My Trips” list he enters in the trip room. Each trip has a Translator, Messages and Trip info page.

Tourist Trip

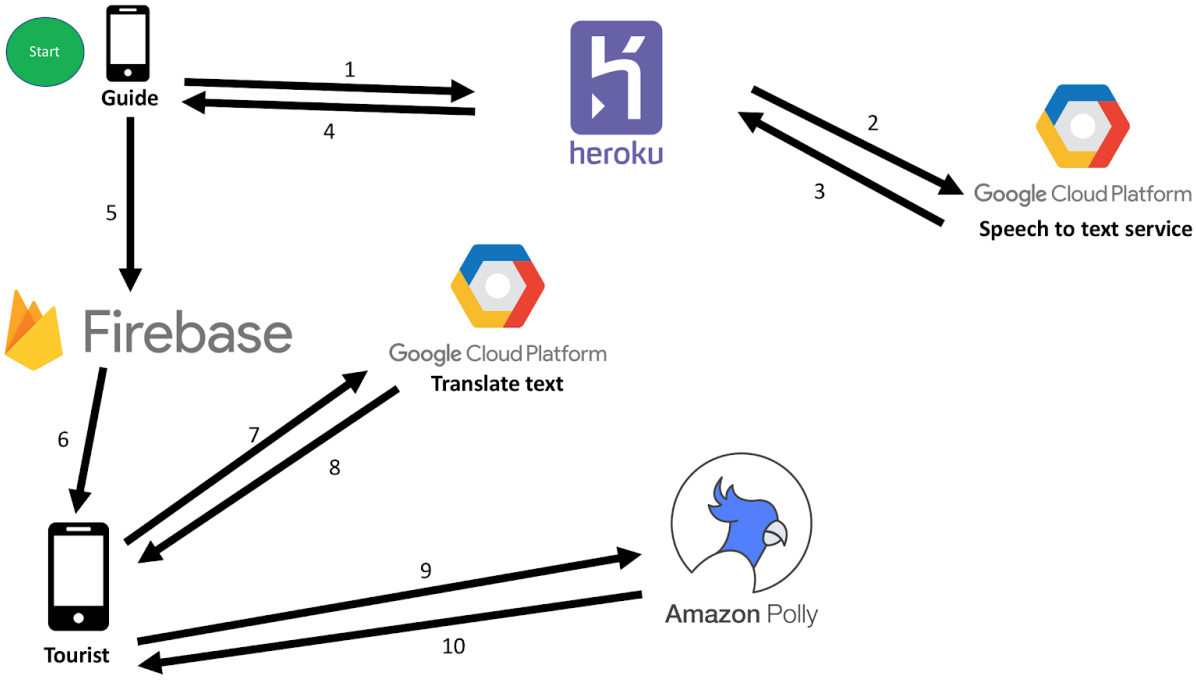
When a tourist joins a trip he has to enter the trip code for that trip. If that is successful he selects a nickname and language and enters the trip room. Translator page is the translated message from the tour guide. Messages page is where tourist can send a

question to the tour guide. Trip Info page is the same for tourist and tour guides, information about the trip.

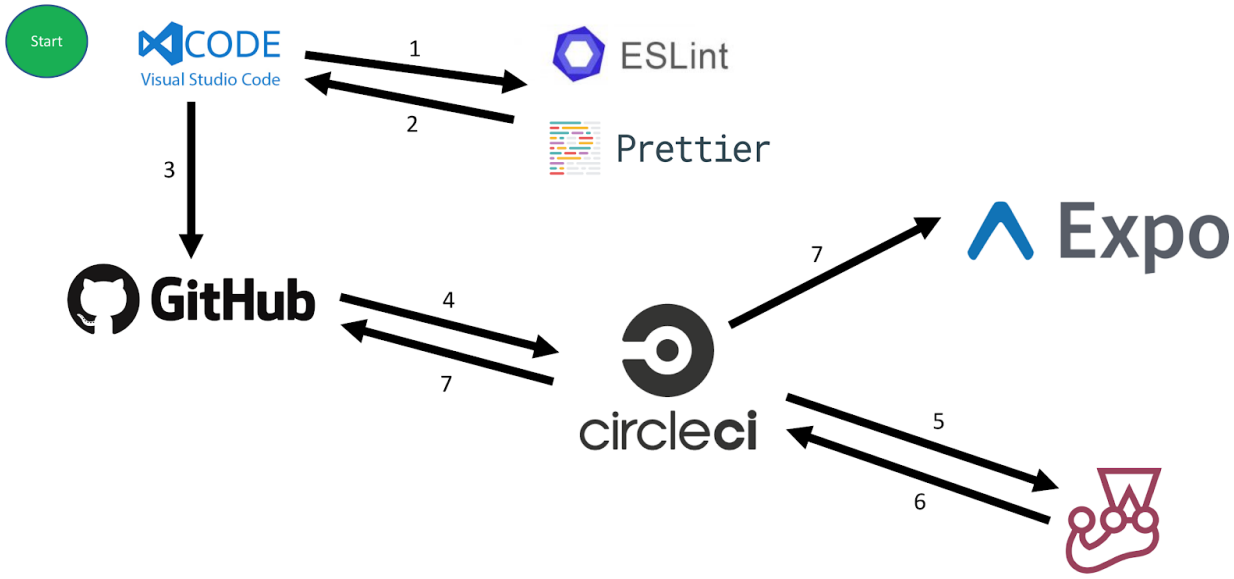
5.2 Logo

The app logo displays a tour guide holding a flag. In the background there are two letters from different alphabets to represent the translation process. The letters are A, the first letter of the Latin alphabet and the japanese letter 旅, which means travel or journey according to google translate.

5.3 Tech stack



This image shows the process when a tour guide talks in the apps and when a tourist gets his message.
Firebase: Our database (handles websocket and database connections)
Google Cloud Platform: Translates messages from and to Tour Guide and Converts speech to Text from Tourist
Heroku: Hosts our web service for converting audio formats and communicating with Google cloud platform for speech to text
Amazon Polly: Reads out loud messages for tourists



This image shows the pipeline when a change is made in the code and it gets pushed to our version control system Github.
VS code: Our code editor
ESLint: Code linter (makes sure we are following coding rules)
Prettier: Resharps our code if ESLint complain
GitHub: Hosts our code and takes care of version control
CircleCI: Our continuous integration tool. (Pulls code from Github, runs tests and publishes if tests succeed)
Jest: Tests our code
Expo: React native toolchain for easy development and testing.

5.4 Database

We use Firebase to store all our data in a JSON format for this project. Here is a photo of our database design.

Database design

See on next page

```
"tour-guide-translator": {
  "codes": {
    "trip-code": "trip-id",
    "trip-code": "trip-id",
    "trip-code": "trip-id"
  },

  "trips": {
    "trip-id": {
      "code": "trip-code",
      "language": "en",
      "name": "trip-name",
      "owner": "owner-id",
      "tourists": {
        "tourist-id": "Nickname",
        "tourist-id": "Nickname"
      }
    },
    "trip-id": {
      "code": "trip-code",
      "name": "trip-name",
      "owner": "owner-id",
      "tourists": {
        "tourist-id": "Nickname",
        "tourist-id": "Nickname"
      }
    }
  },

  "trip-translator": {
    "trip-id": {
      "message-id": {
        "trip-message": "hello velkomin"
      },
      "message-id": {
        "trip-message": "gullfoss er blaa...."
      }
    },
    "trip-id": {
      "message-id": {
        "trip-message": "geysir er blaa...."
      }
    }
  },

  "guides": {
    "guide-id": {
      "name": "guide-name",
    },
    "guide-id": {
      "name": "guide-name",
    }
  },

  "trip-messages": {
    "trip-id": {
      "message-id": {
        "language": "da",
        "message": "jeg hedder william",
        "nickname": "william",
        "read": true,
        "userid": "user-id"
      },
      "message-id": {
        "language": "en",
        "message": "how tall is gullfoss",
        "nickname": "william",
        "read": true,
        "userid": "user-id"
      }
    }

  }
```

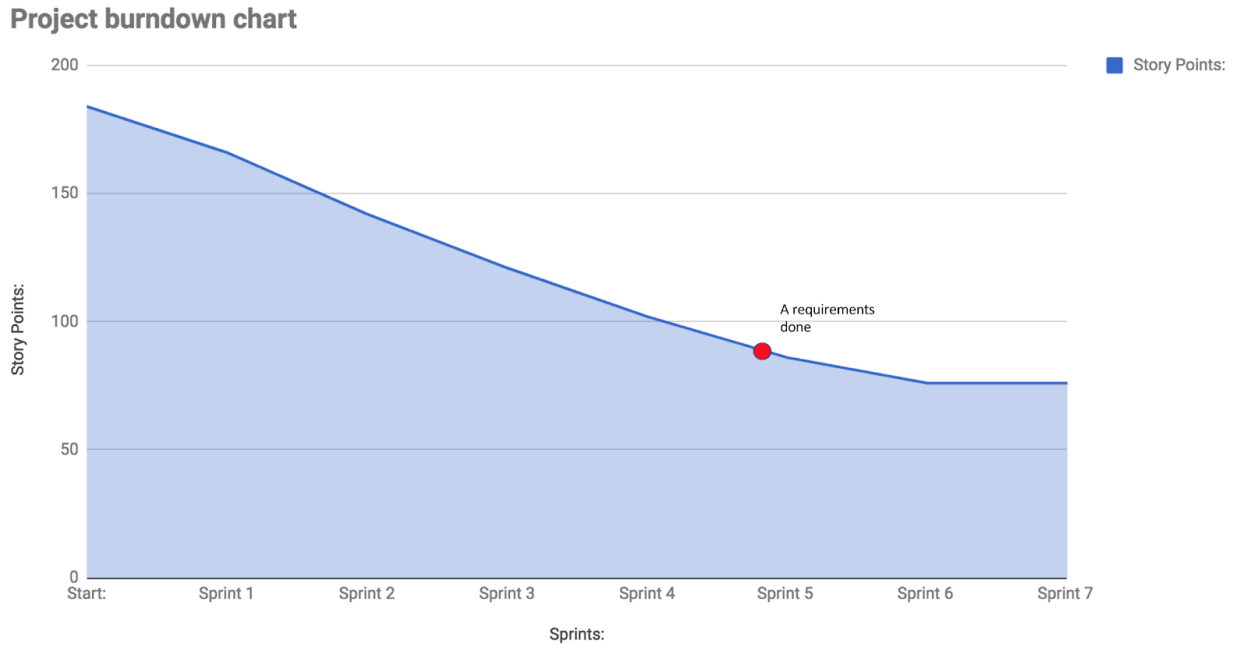
Database rules

```
{
  "rules": {
    "codes": {
      ".read": true,
      ".write": "auth != null",
      "$key": {
        ".validate": "$key.matches(/^[a-zA-Z0-9:^-]+$/) && newData.val().matches(/^-[a-zA-Z0-9-_]+$/)"
      }
    },
    "guides": {
      ".read": true,
      ".write": "auth != null",
      "$key": {
        ".validate": "$key.matches(/^[a-zA-Z0-9]+$/) && newData.hasChildren(['name'])",
        "name": {
          ".validate": "newData.isString()"
        },
        "$other": {
          ".validate": false
        }
      }
    },
    "trip-message": {
      ".read": true,
      ".write": true,
      "$key": {
        ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/)",
        "$key": {
              ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/) && newData.hasChildren(['language', 'message', 'nickname', 'userid', 'read'])",
          "language": {
                ".validate": "newData.val().matches(/^[a-z]{2}$/)"
                },
          "message": {
                ".validate": "newData.isString()"
                },
          "nickname": {
                ".validate": "newData.isString()"
                },
          "userid":{
            ".validate": "newData.val().matches(/^-[a-zA-Z0-9-_]+$/)"
          },
          "read":{
            ".validate": "newData.isBoolean()"
          },
                "$other": {
                ".validate": false
                }
        }
      }
    },
    "trip-translator": {
      ".read": true,
      ".write": "auth != null",
      "$key": {
        ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/)",
        "$key": {
              ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/) && newData.hasChildren(['original_language', 'trip_message'])",
          "original_language": {
                ".validate": "newData.val().matches(/^[a-z]{2}$/)"
                },
          "trip_message": {
                ".validate": "newData.isString()"
          },
                "$other": {
                ".validate": false
                }
        }
      }
    },
    "trips": {
      ".read": true,
      ".write": "auth != null",
      "$key": {
        ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/) && newData.hasChildren(['code', 'language', 'name', 'owner'])",
        "code": {
          ".validate": "newData.val().matches(/^[a-zA-Z0-9:^-]+$/)"
        },
        "language": {
          ".validate": "newData.val().matches(/^[a-z]{2}-[A-Z]{2}$/)"
        },
        "name": {
          ".validate": "newData.isString()"
        },
        "owner": {
          ".validate": "newData.val().matches(/^[a-zA-Z0-9]+$/)"
        },
        "tourists": {
          ".write": true,
          "$key": {
                      ".validate": "$key.matches(/^-[a-zA-Z0-9-_]+$/) && newData.isString()"
          }
        },
        "$other": {
          ".validate": false
        }
        }
    }
  }
}
```

6. Progress

In this section we kept track of the progress of our project. We recorded the time we spent at Origo and stored it in a working diary. We also kept track of the amount of story points we had left in each sprint.

6.1 Project burndown chart



The progress of this project went great and we always finished the requirements in each sprint backlog. We finished all our A requirements late into sprint 4. We did not implement any new requirements to the app in sprint 7. We held a sprint retrospective after each sprint and improved things in the next sprint we did badly in the previous one. For example stop using branches for more than one feature. Field testing our product to see how it works in its real environment and many more things that can bee seen in the details about each sprint section in the annex.

6.2 Working diary

Team members	Working hours
Bjartur Fannar Stefánsson	295.5
Gunnar Birnir Ólafsson	314.5
Ævar Þór Gunnlaugsson	303.5
Total	**913.5**

7. Usability evaluation

We tested the user interface of the app to make sure it was easy and fast to use. We made the evaluation at the time we had a up and running app so we could make the users test it on their phones. We took the results of the evaluation and improved our user interface.

7.1 Preparation

Introduction to the user

We are designing an app that translates the speech from a tour guide to the language of the tourist in real time. We have a prototype of the app that we're going to let you solve some tasks on. We are testing the interface of the app, not your ability to solve the tasks.

User tasks

1. Login as a tour guide.
2. Create a trip as a Tour guide.
3. Inside the trip you created talk the message “hello welcome to the trip” and send it.
4. Exit the trip your in as a tour guide.
5. Logout as a tour guide.
6. Login as a tourist in the trip you created as the tour guide earlier.
7. Listen to the translated text that is in the trip.
8. Send a message to the tour guide.
9. Exit the trip you're in as a tourist.

Questions after the test

1. Did you find the app easy and quick to use.
2. How satisfied are you with the app on the scale 1-5.
3. Something you want to be added or changed in the app.

7.2 Conducting the evaluation

The users that we interviewed where between the age 13 – 60 years old. The interviews took place at the home of the users, and where performed on a computer or a phone.

Description of the users

User 1.
Age: 20 years old.
Gender: Male.
Education: University Student.
Abilities/disabilities: Nothing special.
Computer knowledge: Excellent.

User 2.
Age: 13 years old.
Gender: Male.
Education: Elementary school student.
Abilities/disabilities: Nothing special.
Computer knowledge: Average.

User 3.
Age: 43 years old.
Gender: Male.
Education: Teacher.
Abilities/disabilities: Nothing special.
Computer knowledge: Average.

User 4.
Age: 24 years old.
Gender: Female.
Education: University student.
Abilities/disabilities: Nothing special.
Computer knowledge: Excellent.

User 5.
Age: 17 years old.
Gender: Male.
Education: High school student.
Abilities/disabilities: Nothing special.
Computer knowledge: Excellent.

User 6.
Age: 59 years old.
Gender: Male.
Education: Business administrator.
Abilities/disabilities: Nothing special.
Computer knowledge: Bad.

7.3 Measurable goals

Usability goals

User goal	Usability factor	Data collected	Worst case	Preferred case	Best case	Value now
Login as a tour guide	Efficiency	Time min:sec	1:00	0:30	0:20	0:32
Login as a tour guide	Effectiveness	Finished tasks	95%	100%	100%	100%
Login as a tour guide	Satisfaction	Grade 1-5	4	5	5	5

User goal	Usability factor	Data collected	Worst case	Preferred case	Best case	Value now
Create a trip as a Tour guide	Efficiency	Time min:sec	1:00	0:30	0:20	0:29
Create a trip as a Tour guide	Effectiveness	Finished tasks	90%	95%	100%	100%
Create a trip as a Tour guide	Satisfaction	Grade 1-5	4	5	5	5

User goal	Usability factor	Data collected	Worst case	Preferred case	Best case	Value now
Inside the trip you created talk the message “hello welcome to the trip” and send it	Efficiency	Time min:sec	0:30	0:15	0:10	0:16
Inside the trip you created talk the message “hello welcome to the trip” and send it	Effectiveness	Finished tasks	90%	95%	100%	100%
Inside the trip you created talk the message “hello welcome to the trip” and send it	Satisfaction	Grade 1-5	4	5	5	5

User goal	Usability factor	Data collected	Worst case	Preferred case	Best case	Value now
Exit the trip your in as a tour guide	Efficiency	Time min:sec	0:30	0:15	0:05	0:19
Exit the trip your in as a tour guide	Effectiveness	Finished tasks	90%	95%	100%	100%

Exit the trip your in as a tour guide	Satisfaction	Grade 1-5	4	5	5	4

User goal	**Usability factor**	**Data collected**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Logout as a tour guide	Efficiency	Time min:sec	0:15	0:05	0:02	0:03
Logout as a tour guide	Effectiveness	Finished tasks	90%	95%	100%	100%
Logout as a tour guide	Satisfaction	Grade 1-5	4	5	5	4

User goal	**Usability factor**	**Data collected**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Login as a tourist in the trip you created as the tour guide earlier	Efficiency	Time min:sec	0:30	0:20	0:10	0:20
Login as a tourist in the trip you created as the tour guide earlier	Effectiveness	Finished tasks	90%	95%	100%	100%
Login as a tourist in the trip you created as the tour guide earlier	Satisfaction	Grade 1-5	4	5	5	5

User goal	**Usability factor**	**Data collected**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Listen to the translated text that is in the trip	Efficiency	Time min:sec	0:10	0:05	0:3	0:04

Listen to the translated text that is in the trip	Effectiveness	Finished tasks	90%	95%	100%	100%
Listen to the translated text that is in the trip	Satisfaction	Grade 1-5	4	5	5	4

User goal	**Usability factor**	**Data collected**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Send a message to the tour guide	Efficiency	Time min:sec	0:25	0:10	0:5	0:07
Send a message to the tour guide	Effectiveness	Finished tasks	90%	95%	100%	100%
Send a message to the tour guide	Satisfaction	Grade 1-5	4	5	5	5

User goal	**Usability factor**	**Data collected**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Exit the trip you're in as a tourist	Efficiency	Time min:sec	0:10	0:05	0:02	0:04
Exit the trip you're in as a tourist	Effectiveness	Finished tasks	90%	95%	100%	100%
Exit the trip you're in as a tourist	Satisfaction	Grade 1-5	4	5	5	5

Goals about usability

Goals about usability problems	**Usability factor**	**Worst case**	**Preferred case**	**Best case**	**Value now**
Serious problems	Testing	1	0	0	0

Mild problems	Testing	2	1	0	1
Small problems	Testing	4	2	0	2

User experience goals

User experience goals	**Usability factor**	**Worst case**	**Preferred case**	**Best case**	**Value now**
User experience of the app	Testing	1	4	5	5

7.4 Results of the evaluation

The users said after the interview that they thought the app was easy and quick to use. All users but one rated the app 5 out of 5. The things they wanted to change or add in the app were: move the exit button when solving task 4, make it louder when they listen to the message in task 7, make the listen live button more clear what it does in the translator screen as a tourist. And finally they wanted to add a map to the map screen in the app.

After conducting the evaluation we see that we met almost all our goals. But the goals we did not meet like the efficiency and satisfaction in tasks 4 and 7 we will have to fix. This evaluation gave us a good indication of what is good in the UI and what we need to fix. Also just to see how users like the app and how they see it.

8. Conclusion

When we look back at the project we are very proud of the result. It was hard at times but it usually went well. We finished all our A requirements and couple of the B requirements. We started off slow mainly because we decided to program the app in React-Native. We had little former experience with it and it took some time to get the hang of it. We do not regret the decision of going with React-Native as we have gained a valuable experience that we will use in the future. The cooperation with Origo went well and we had access to programmers and designers throughout the whole project. We had a great workspace there and got free lunch each day. Our teacher assistant Fanney was very helpful throughout the whole project and very easy to reach out to her and schedule a meeting. The future vision for this project is unclear. We have finished our part and now Origo will get our app and they can decide what the want to do with it. This type of technology is developing and getting better each day. We think this type of technology is advanced enough today to be successfully used in this way. This was a fun experiment and we think that with predetermined scripts we can see better results in this technology but it still lacks some functionality as react native is still really new. Finally we want to thank Reykjavík University for teaching this course. This course gives students a chance to take a look at the software development industry and gain experience in working as a software developer. It also gives students connections and relationships with the company they work for and if they do well it makes it easier for them to get a job at that company or any other company after graduation.

Annex

Results of the evaluation for each user

Task 1 - Login as a tour guide.

Users	Completed	Time (min:sec)	Satisfaction (1-5)
User 1	Yes	0:30	5
User 2	Yes	0:20	4
User 3	Yes	0:35	5
User 4	Yes	0:16	5
User 5	Yes	0:23	5
User 6	Yes	1:05	3

Task 2 - Create a trip as a Tour guide.

Users	Completed	Time (min:sec)	Satisfaction (1-5)
User 1	Yes	0:33	5
User 2	Yes	0:25	5
User 3	Yes	0:30	5
User 4	Yes	0:27	5
User 5	Yes	0:12	5
User 6	Yes	0:45	4

Task 3 - Inside the trip you created talk the message “hello welcome to the trip” and send it.

Users	Completed	Time (min:sec)	Satisfaction (1-5)
User 1	Yes	0:15	5
User 2	Yes	0:20	5

User 3	Yes	0:25	5
User 4	Yes	0:08	5
User 5	Yes	0:19	5
User 6	Yes	0:12	4

Task 4 - Exit the trip your in as a tour guide.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:03	5
User 2	Yes	0:35	3
User 3	Yes	0:30	3
User 4	Yes	0:06	5
User 5	Yes	0:04	5
User 6	Yes	0:38	3

Task 5 - Logout as a tour guide.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:02	5
User 2	Yes	0:02	5
User 3	Yes	0:04	5
User 4	Yes	0:03	5
User 5	Yes	0:02	5
User 6	Yes	0:03	5

Task 6 - Login as a tourist in the trip you created as the tour guide earlier.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:15	5
User 2	Yes	0:20	4
User 3	Yes	0:18	5

User 4	Yes	0:14	5
User 5	Yes	0:17	5
User 6	Yes	0:36	4

Task 7 - Listen to the translated text that is in the trip.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:01	5
User 2	Yes	0:02	2
User 3	Yes	0:03	3
User 4	Yes	0:05	5
User 5	Yes	0:02	5
User 6	Yes	0:12	3

Task 8 - Send a message to the tour guide.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:10	5
User 2	Yes	0:06	5
User 3	Yes	0:08	5
User 4	Yes	0:06	5
User 5	Yes	0:03	5
User 6	Yes	0:10	5

Task 9 - Exit the trip you're in as a tourist.

Users	**Completed**	**Time (min:sec)**	**Satisfaction (1-5)**
User 1	Yes	0:02	5
User 2	Yes	0:02	5
User 3	Yes	0:04	5
User 4	Yes	0:03	5

User 5	Yes	0:03	5
User 6	Yes	0:07	5

Detail about each sprint

Sprint 1

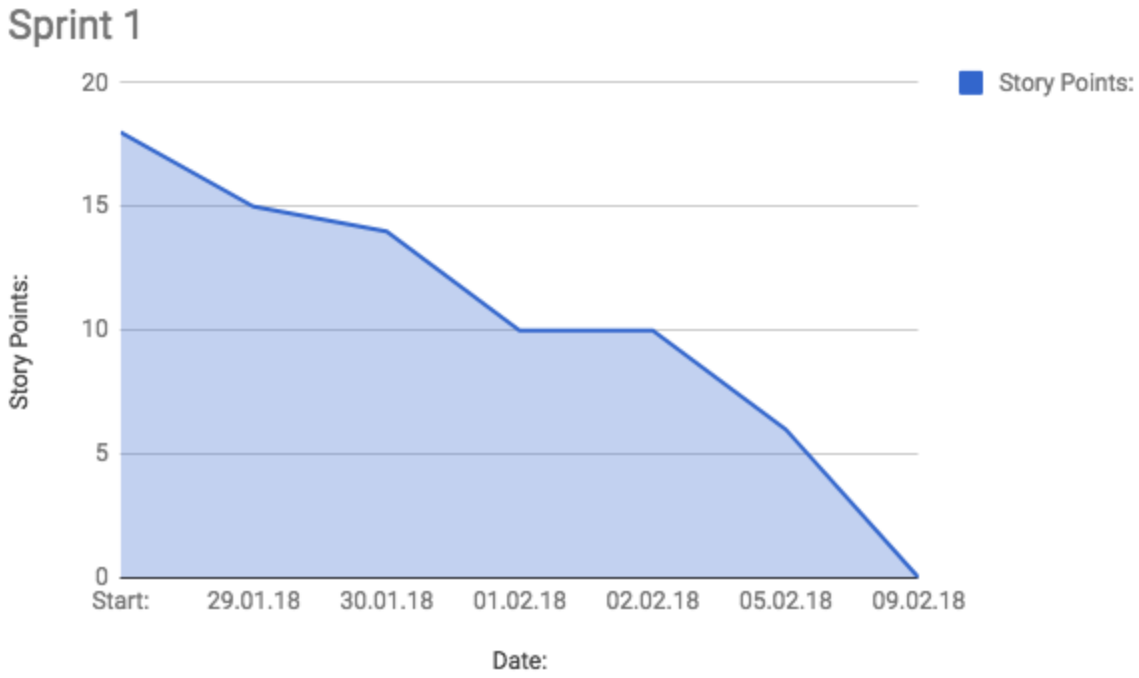
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tour guide can sign in	**3**	**DONE**
2	**A**	Tour guide can create a room and name it and code will be generated for the room	**8**	**DONE**
3	**A**	Tourist can join a room by entering the room code or scanning the room QR code	**5**	**DONE**
4	**A**	Set up CI pipeline	**2**	**DONE**

Work time

Bjartur: 48 hours
Gunnar: 49.5 hours
Ævar: 45 hours
Total: 142.5 hours.

Sprint burndown



Sprint retrospective

Start doing:

- Test before we push on branch Dev

Stop doing:

- Not testing the code

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo

Sprint 2

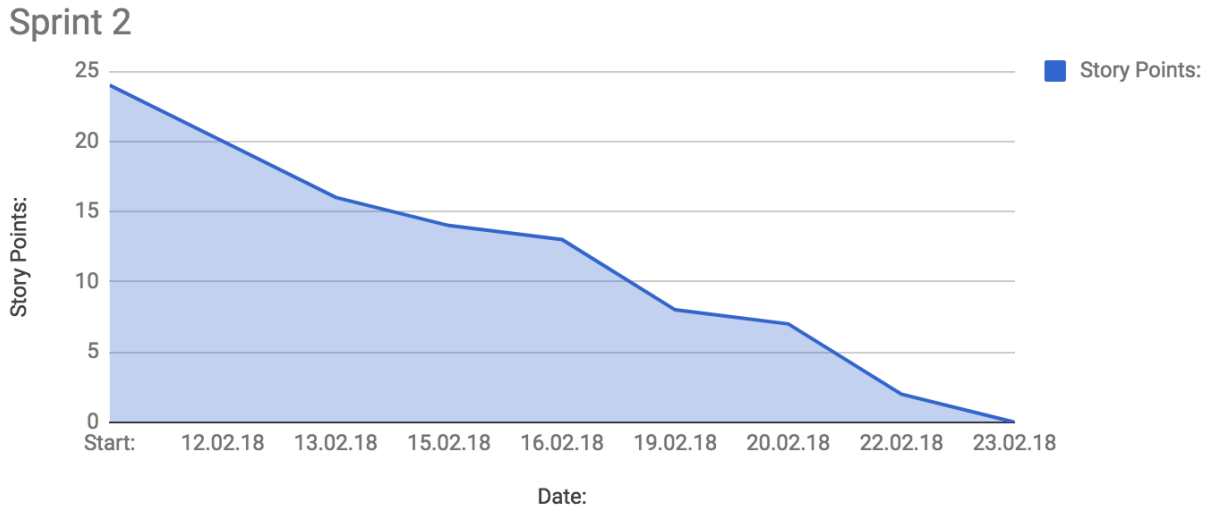
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tour guide can write a text and send it to the room he's in	**3**	**DONE**
2	**A**	Tourist can see the text from the tour guide	**3**	**DONE**
3	**A**	Tourist can get translated messages from the tour guide	**8**	**DONE**
4	**A**	Get the Jest test to work	**2**	**DONE**
5	**A**	Tourist can listen to the translated message from the tour guide	**8**	**DONE**

Work time

Bjartur: 44.5 hours.
Gunnar: 52 hours.
Ævar: 57 hours.
Total: 153.5 hours.

Sprint burndown



Sprint retrospective

Start doing:

- Implementing database rules
- Better team communication

Stop doing:

- Using branches for more than one feature

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo
- Refactoring

Sprint 3

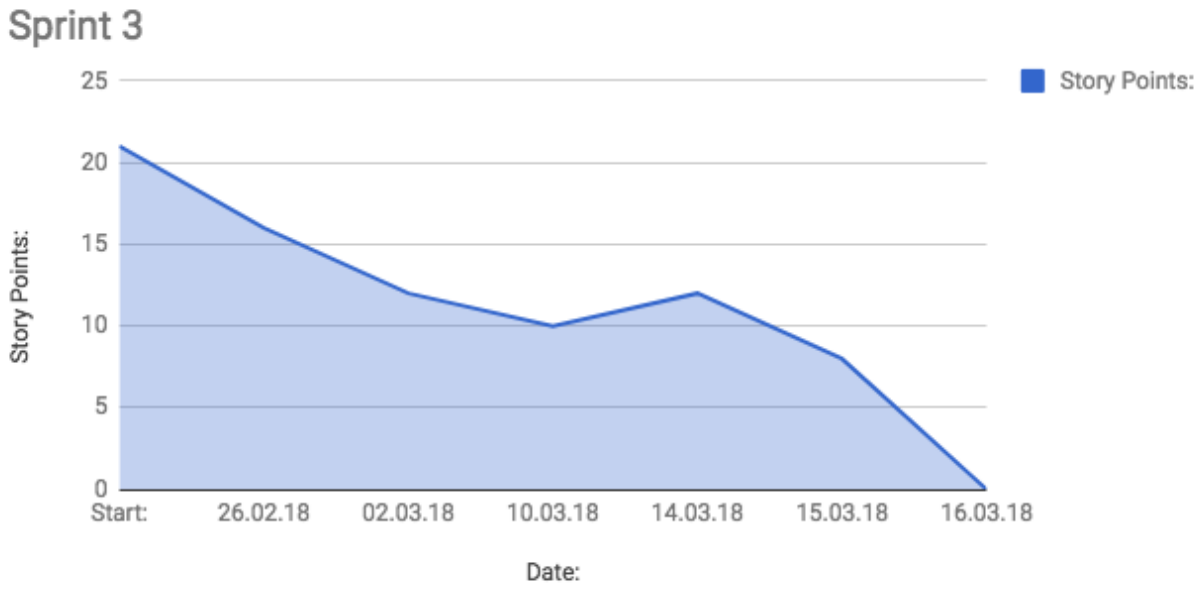
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tour guide talks in a mic in a selected language and the app displays the words	**13**	**DONE**
2	**A**	Tourist can set a nickname in a room which the tour guide will see when they ask a question	**3**	**DONE**
3	**A**	Tourist can choose the language that they want listen to the tour guide in	**2**	**DONE**
4	**A**	Test redux components	**3**	**DONE**

Work time

Bjartur: 43 hours.
Gunnar: 48.5 hours.
Ævar: 53 hours.
Total: 144.5 hours.

Sprint burndown



Sprint retrospective

Start doing:

- More frequent team meetings
- Cleaning up the database
- Implementing database rules

Stop doing:

- Using branches for more than one feature

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo
- Pair Programming
- Frequent meetings with the team instructor

Sprint 4

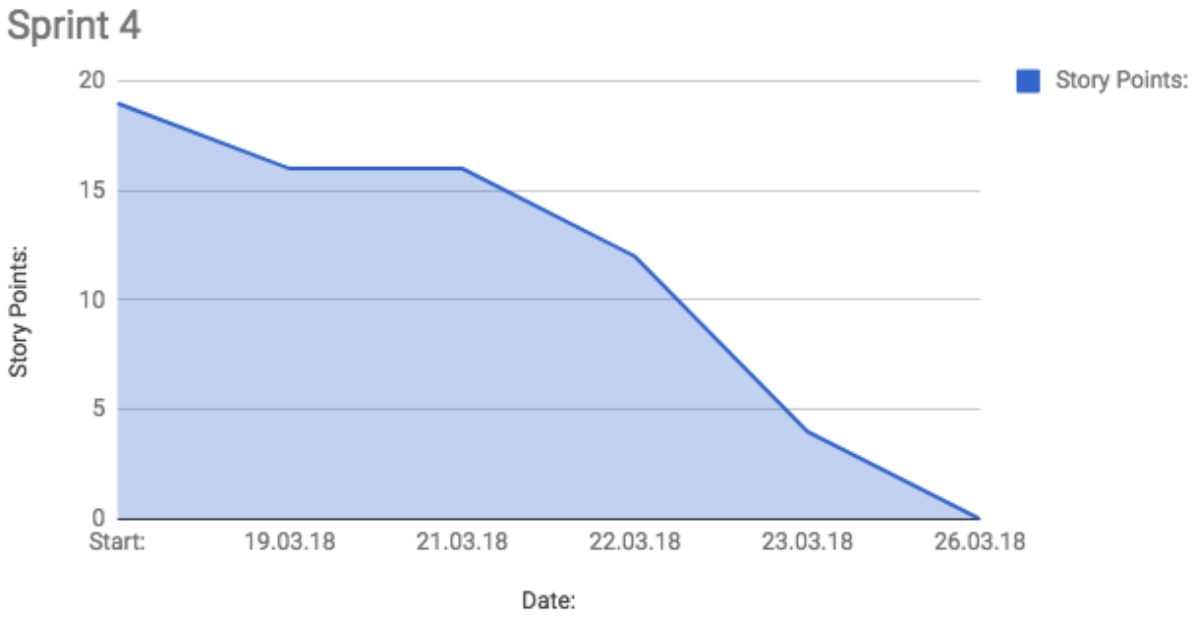
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tourist can send a question to the tour guide	**2**	**DONE**
2	**A**	Tour guide can speak an answer to a question that a tourist sends	**3**	**DONE**
3	**A**	Tour guide can push a button when the trip is over and close the room	**2**	**DONE**
4	**A**	Improve the UI Looks and bugs	**5**	**DONE**
5	**A**	Tourist can change the language that he listens to the tour guide in inside the trip	**3**	**DONE**
8	**A**	Tourist can see the QR code for the trip he's	**2**	**DONE**
9	**A**	Tour guides can see the QR code for the trip he’s in	**2**	**DONE**

Work time

Bjartur: 43.5 hours.
Gunnar: 45 hours.
Ævar: 28.5 hours.
Total: 117 hours.

Sprint burndown



Sprint retrospective

Start doing:

- Field test our product
- Implementing database rules

Stop doing:

- Nothing particular

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo
- Pair Programming
- Frequent meetings with the team instructor
- Refactoring

Sprint 5

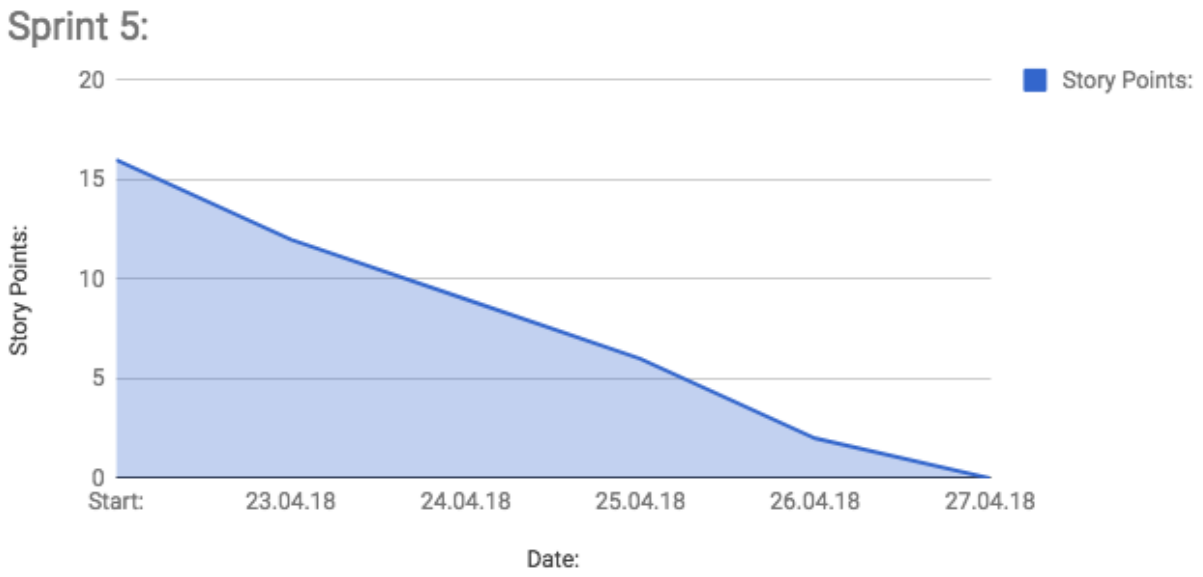
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**A**	Tour guide can change the language he speaks in inside the trip	**3**	**DONE**
2	**A**	Tour guide can change his name and logout	**3**	**DONE**
3	**B**	Tourist can ask a question in their language and the tour guide sees the question in his language	**3**	**DONE**
4	**A**	Implement database rules	**5**	**DONE**
5	**B**	Tour Guide can see how many people are in the trip	**2**	**DONE**

Work time

Bjartur: 34 hours.
Gunnar: 34 hours.
Ævar: 34.5 hours.
Total: 102.5 hours.

Sprint burndown



Sprint retrospective

Start doing:

- Testing components before we push them to Github
- Test the UI

Stop doing:

- Nothing particular

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo
- Pair Programming
- Frequent meetings with the team instructor
- Refactoring
- Field test our product

Sprint 6

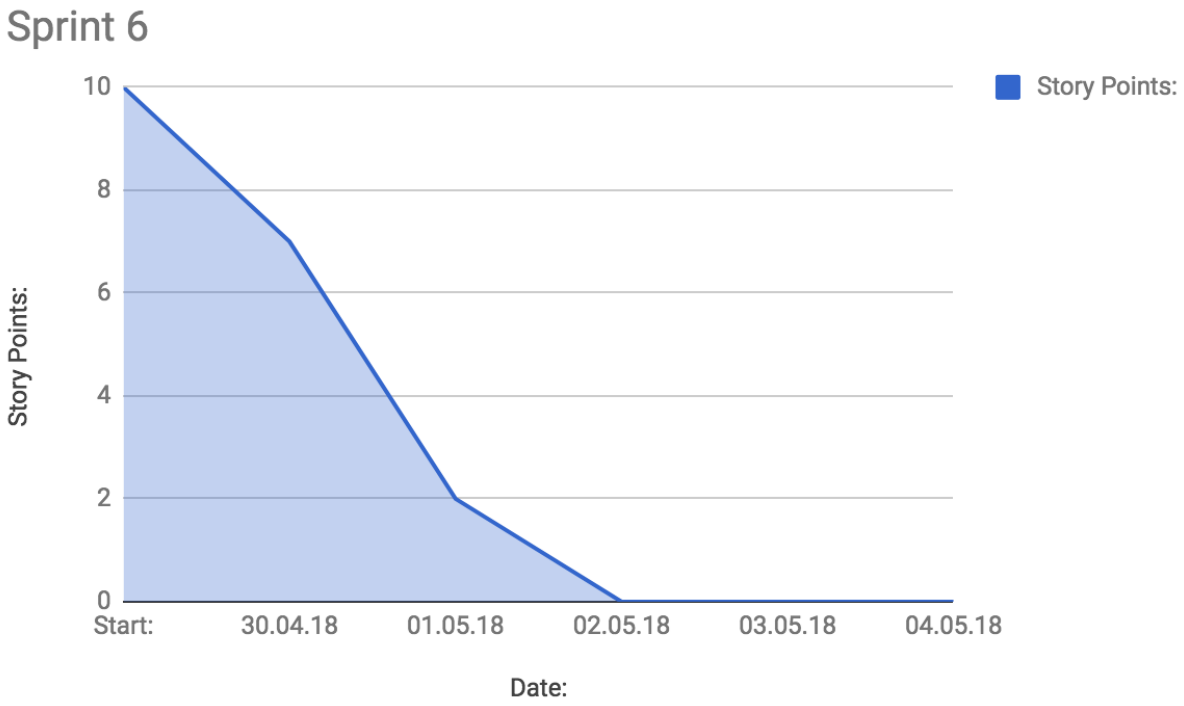
Sprint backlog

Nr	Priority	Description	Estimate	Status
1	**B**	Tour guide gets notification when a new message is received from the tourist	**5**	**DONE**
2	**B**	Tourist can see on a map where he is	**5**	**DONE**

Work time

Bjartur: 42 hours.
Gunnar: 44.5 hours.
Ævar: 44.5 hours.
Total: 131 hours.

Sprint burndown



Sprint retrospective

Start doing:

- Testing components before we push them to Github
- Test the UI

Stop doing:

- Nothing particular

Continue doing:

- Using branch by feature
- Clear division of work
- Frequent work days together at Origo
- Pair Programming
- Frequent meetings with the team instructor
- Refactoring
- Field test our product

Sprint 7

This sprint was not a typical sprint because we did not implement any new features for the app. We were just refining the app and making sure there were no bugs.

Work time

Bjartur: 39 hours.
Gunnar: 39 hours.
Ævar: 39 hours.
Total: 117 hours.