Feedforward App

Project Plan

Table of contents

Students: Michael Cornelisse (459466), Dylan Cornelisse (462777), Petar Mihaylovski (459725), Kaloyan Krastev (467364)
Academy: ACT
Course: Project Persistent
Teacher: René van den Nieuwenhoff

Date: 09/11/19

1. Introduction

One of the most valued parts of going to an educational institution is the ability to interact with the tutors which always have valuable knowledge and experience to share with their students.
Unfortunately this knowledge isn't that easy to ask for or access considering it is sought by many people but only provided by a couple.
"Parantion" is trying to resolve this issue by having a technological edge on the physical constraints which are implied by this interaction. In this document we will go over how the Feedforward App "Parantion" has envisioned may very well be the indefinite solution to the discrepancy between wanting and actually receiving feedback in educational institutions worldwide.

2. Unknown

Since we've never used most of the technologies we will be using before, it is safe to say we are going into uncharted territory. We will rely on the documentation that's already on the Internet and on the material that other programmers have posted both of errors and solutions.
Also we've never worked with an external client so it will also be a new experience. We are hoping we manage to establish a frequent and fulfilling back and forth.

3. Project Organization

3.1. Working Agreements

Communication
In order for the team to walk in the right direction, all of the members should be informed of the current state of the project. It is beneficial for all sides nothing regarding the project to be kept personal, so if anyone has any problems along the way or needs help with anything, the group's members will be informed so the adequate help can be given. Moreover, the team agreed that if someone thinks that he has found a reliable information/resource for resolving another team member's issue, it has to be shared.

Each one has a clear task
The team has divided four general roles stated in 3.3 Group Positions. However, every team member is free to take whichever issue in gitlab he wants or feels comfortable with as long as there are consistent deliveries.

Meet the deadlines
Every GitLab issue has a certain weight, which is voted by the team. The team will use this weight as an approximate time that has to be spent on the specific problem. However, it is possible to underrate certain issues. If a person takes too much time to finish the issue, the team members should be notified, so we can solve the problem.

Schedule weekly meetups
A good time for face-to-face meetings would be Monday or Wednesday before or after classes. Otherwise the team could do the online (WhatsApp, Discord) meetups on the evenings of any day in which all of us are free.

Give feedback how the others are doing
In order to assure that the project is of high-quality, the code is clean and the project is wrapping up as planned, a feedback on the tasks done so far has to be given. Both good and bad feedback are appreciated, as long as they are shared professionally.

Work together
Share responsibilities and help each other in order to be able to deliver as a team. Each person should be putting about 8 hours of work outside the scheduled school hours.

Tools
Google Docs/Sheets, GitLab, Visual Studio Code, Angular, Ionic, GraphQl.

Version control work
Before starting development on a new functionality every team member is expected to pull the latest project version from the master branch. Consistent commits are also necessary, for the different logical parts of the issue. This way if something

somewhere goes wrong, it could be fixed without discarding most of the finished work.

Definition of done
When the functionality of the user story is met in an efficient, error-prone and error-free way and is documented. Test the breakability of the code with either user interaction or coded tests.

Consequences
If anyone is refusing to communicate and cooperate with the group we will try to resolve it with a conversation and if that doesn't work we will consider kicking the person out or reporting to the project teacher.

3.2. Team

Name: Michael Cornelisse
Class: DHI2V.Sp
Student number: 459466
Email: 459466@student.saxion.nl
Phone number: 06 3617 0793

Name: Dylan Cornelisse
Class: DHI2V.Sp
Student number: 462777
Email: 462777@student.saxion.nl

Name: Petar Mihaylovski
Class: DHI2V.Sp
Student number: 459725
Email: 459725@student.saxion.nl

Name: Kaloyan Krastev
Class: DHI2V.Sp
Student number: 467364
Email: 467364@student.saxion.nl

3.3. Group Positions

- **Michael** - communication with the company and writing code
- **Dylan** - writing code
- **Petar** - managing the workflow, managing the database, and writing code
- **Kaloyan** - styling, writing code

4. Planning

4.1. Timesheet

In order to track the time we spend on the development of the application we will use Toggl. It will help us in creating a timesheet which in turn will enforce our work process.

4.2. Stakeholders

Parantion are stakeholders to the project since they are giving us the requirements for the application. End result is the importance for them.
Saxion are stakeholders to the project since they are providing the planning of the sprints and expect something from us. The end result is not that important for them, but the process of the application development.
The final users of the Feedforward application are stakeholders since they are the ones who are going to use the application. The final result is of importance for them, not the way it was developed.
The team developing the application is an internal stakeholder since the project directly correlates with their educational activities.

Stakeholder	**Number**	**Position**	**Internal/ External**	**Level of knowledge**	**Interest**	**Resources**
Parantion	PAR01	Owners	Internal	3	3	1
Saxion	SAX02	Planners	Internal	0	3	3
The users	US03	Users	External	0	3	0
The developers	DEV04	Developers	Internal	2	3	2

Stakeholder	**Importance**	**Influence**
Parantion	**High**	**High**

Saxion	**High**	**High**
The users	**Medium**	**Low**
The developers	**High**	**High**

4.3. Client Communication

Parantion contact information:
Kris Minkjan: kris.minkjan@parantion.nl

4.4. Storage of Documents

We will be storing the different types of files that we'll have with the help of several external applications:

- **Google Docs** the documentation
- **Mockflow** for our mockups/wireframes
- **GitLab** for our issues and code
- **Google Slides** for our presentations
- **Toggle** for our timesheet management

4.5. Deadlines

The team set deadlines for the sprints are having a Scrum Standup every Monday, and one at the end of the week. The issue weight is also considered a certain type of team set time constraint, but we are not going to look at it strictly, since the measuring may not be as accurate as we would want to.
University set deadlines are:
Sprint 0: at the end of the week 2 (15/09/2019) delivering the Project plan.
Sprint 1 is at the end of week 5 (06/10/2019) delivering the working product that the team did for the 3 week duration of Sprint 1.
Sprint 2's deadline is the end of week 8 (01/11/2019) delivering the finished product with the product's documentation.

5. Application Design

5.1. Requirements

Must

- The user must be able to give feedback by voice.
- The user must be able to give feedback through text.
- The user must be able to select who to give feedback to.
- The giver of feedback must be able to specify the context.
- The user who requests the feedback must add a context to it.
- The giver of the feedback must be able to add context to the feedback.
- The user must be able to turn off push notifications.
- The maximum upload of files must be less than or equal to 200MB at a time.
- The feedback must be stored in the portfolio.
- The student must be able to request feedback.
- The user must not be able to submit an empty request or feedback.

Should

- The sender should be able to delete their given feedback so long the recipient hasn't seen it.
- The sender should be able to complement their given feedback so long the recipient hasn't seen it.
- The users should be able to send feedback with the minimum of three click actions.
- The lecturers should be able to see their students split in classes.
- The student should be able to see their lecturers split by subjects.
- The students should see their latest feedback when they open the app.
- The lecturers should be able to see their latest feedback requests when they open the app.

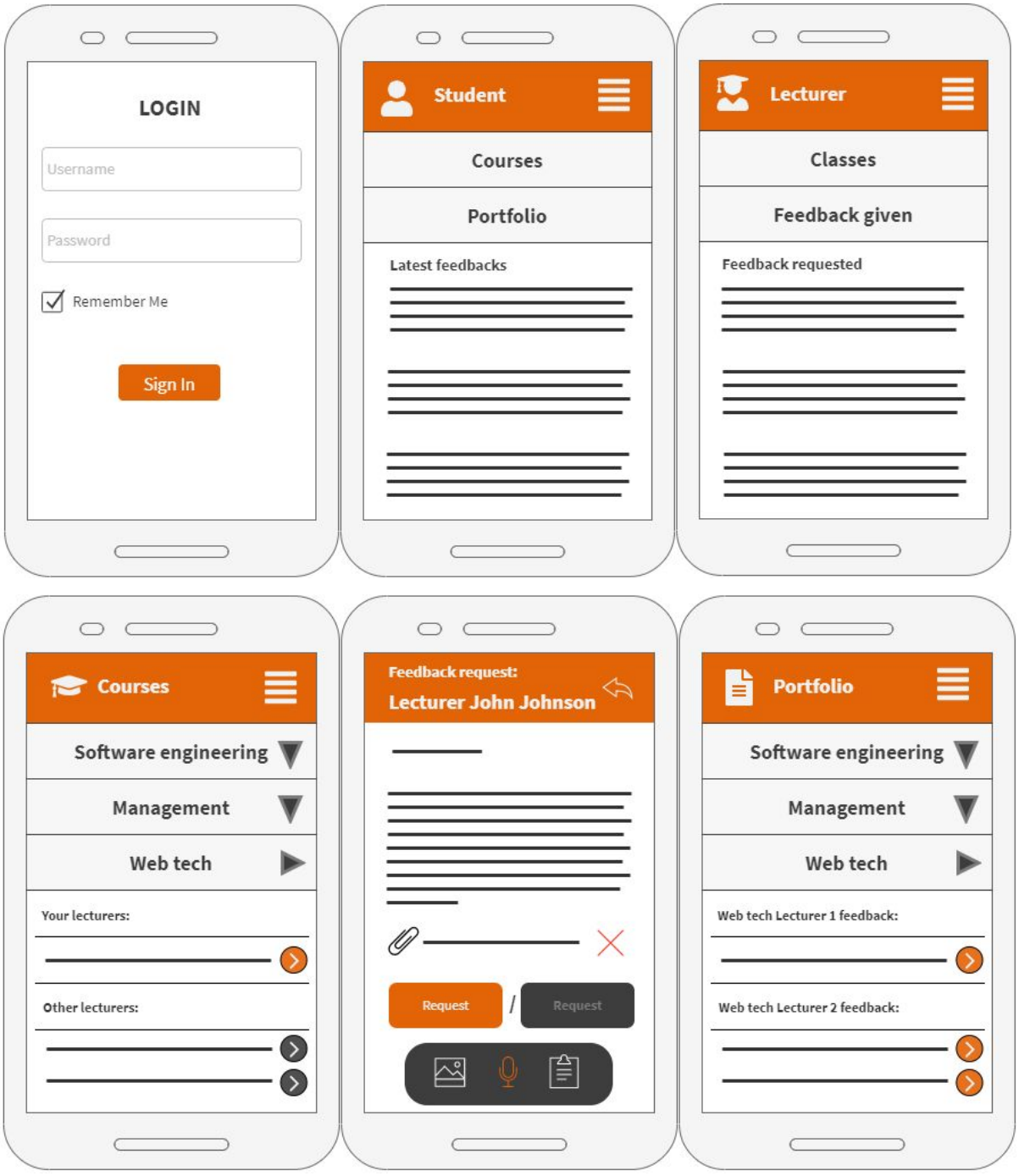
Could

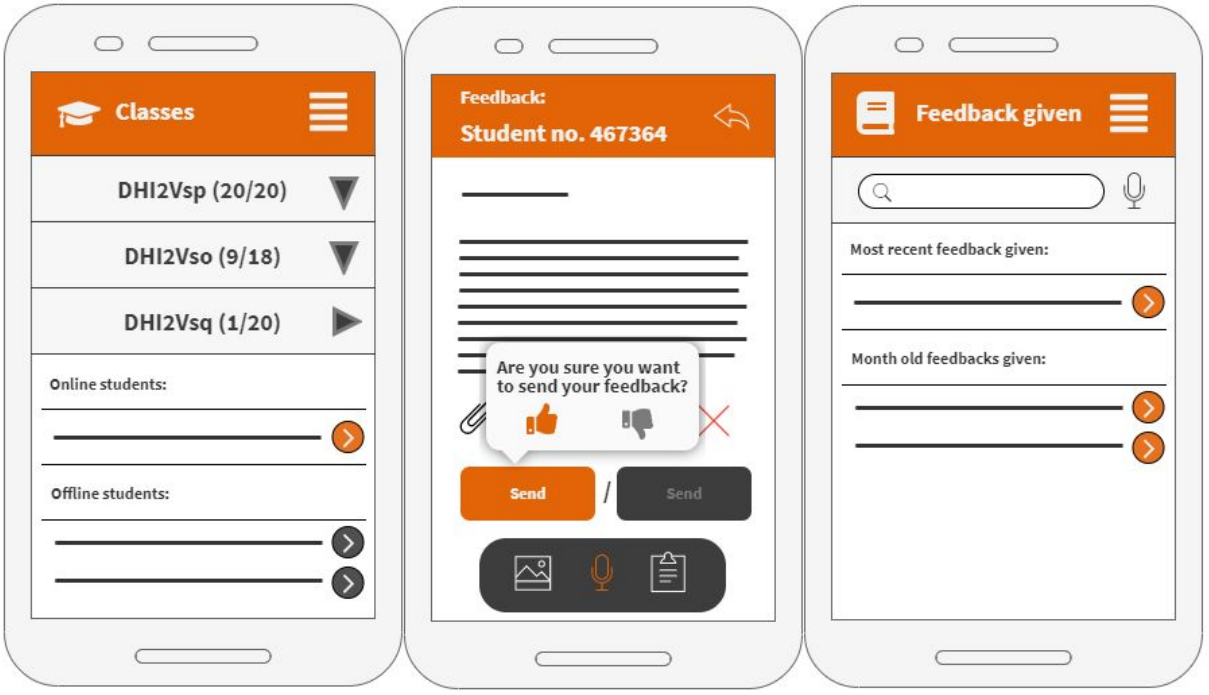
- The app could have the ability to detect lecturer and students in the same area.
- The app could have transitions when navigating between different screens.
- The app could have reminders specified by the user
  ex: If they haven't read their feedback in a certain amount of time.
- The lecturers could be able to search through their given feedback.
- The students could be able to search through their received feedback.
- The lecturer could have a custom list of frequently contacted students.

Would

- The app would be fully controllable by voice throughout speech recognition.

5.2. Mockups


LOGIN
Username
Password
Remember Me
Sign In
Student
Courses
Portfolio
Latest feedbacks
Lecturer
Classes
Feedback given
Feedback requested
Courses
Software engineering
Management
Web tech
Your lecturers:
Other lecturers:
Feedback request:
Lecturer John Johnson
Request
Request
Portfolio
Software engineering
Management
Web tech
Web tech Lecturer 1 feedback:
Web tech Lecturer 2 feedback:



5.3. Future Impediments

If we manage to create the basic layout and functionality in a reasonable amount of time then we would like to explore and possibly implement a fully voice controllable version of the app.
This means that the app can be fully navigated through with voice commands like “Send feedback to X”, “Respond to X’s feedback request”, “Open Classes”, “Open my Portfolio”, “Go Back” and so on. Since we are still unsure of how long the development of the basic functionality will take and even more unsure of how feasible the voice controllable version is it is still very much up in the air.

5.4. Resources Used

- **Google Docs** to share the documentation and edit it together
- **WhatsApp** and **Discord** are used for communication within the group
- **GitLab** for the project backlog and sharing of code
- **Toggl** for use in our timesheet updating
- **YouTube** and **Google** for help with questions we might have

6. Final product

6.1. Expected Result

After the end of the second sprint, the team has to deliver the final product. We believe that we will be able to complete all the functionalities that are must and should have. The final product will be a progressive web application suitable for both Android and iOS devices. The users will be able to use the full functionality with ease. Both lecturers and students will be able to view their classes and courses and contact a student or lecturer respectively. Feedback will be requested from students, and the lecturer will be able to give the feedback either by voice message or by typing it. Moreover, the lecturers and students will be able to attach a file, most likely a video feedback or request. After the feedback has been sent it will be automatically added to said student's portfolio.

6.2. Product Description

The application will be created with Ionic which will make it runnable in both Android and iOS and even as a web app. The purpose of the app is to create a feedback network between students and lecturers in which students can request feedback from specific lecturer and the lecturers can choose a student to whom they can give feedback or respond to a request. A lot of the communication will be possible with both audio and text in order to ease the user experience.