

RateMyCourse User Stories Document

George Rakshiev \wedge Andrew Lalis \wedge Koen Bolhuis \wedge Ludger Visser

September 24, 2018

1 Introduction

1.1 Overview

In this document we will discuss our product backlog and what methods/tools we are using to help track progress. On top of this, we will discuss our progress and how are product has evolved so far into the development process.

1.2 Tool selection

To help manage our product backlog we elected unanimously to use Trello. Not much time was spent on researching other tools or applications, given everyone's familiarity with Trello. Since we have somewhat strict time constraints, it would make more sense to stick with what we know best and move along with the actual development of the product.

As far as actual experience goes, our whole group has at least used Trello in our Software Engineering class last year. This mean about a semester's worth of experience. Going further, George uses Trello to manage his daily life. Andrew has had experience with GitLab's implementation of a backlog as it is used at his part-time job. Ludger has also used Trello at his workplace. Koen has used Trello on a couple past projects of his own.

2 Product Backlog

Due to the relatively small scale of the product, creating a backlog took only a matter of minutes, because of the concise requirements defined previously in the Requirements Document and Minimum Viable Product description. Working from this starting point, items were only placed in the backlog if they are included in the minimum viable product. This way, the team can focus more precisely on only those things explicitly necessary for the first iteration of the product, without any doubt about tentatively wanting to add features halfway through the development cycle.

2.1 Story Point Definition

Although the scope of this product does not necessitate strict point value assignments for each item on the backlog, such values can be estimated using what development work has already been completed before the start of the first sprint.

During the second week of the class, George, having prior experience with Python's Django framework, created a rudimentary REST API and schema for storing reviews for universities alone. Accomplished in a matter of only a few days, this first development gives an estimate to the velocity of further development of the product. At this velocity, approximately one large feature can be produced in the course of a week. Assuming two week sprint intervals, this means that it can be expected for the development team to complete about two major features per sprint.

Using the above metric, for the next sprint, it can confidently be assumed that the next two major features will be implemented: Pages for each entity stored in the database, and user interaction.

3 Reflection

3.1 The Good

As mentioned before the scope for our MVP was rather limited. Given this we were able to create tasks and execute on those tasks rather effectively. Although all of us had some experience with the tech stack, whether it be Django, REST APIs, or Bootstrap, none of us had experience with are whole stack. It was to be expected that some trouble would arise with creating a REST API within Django or integrating the API with our front end. The opposite was actually true and we believe it was because of the scope. Limiting our MVP to the bare bones as far as functionality went helped us tackle and focus on the issues mentioned above.

3.2 The Okay

Before even creating our backlog we decided to split up our team into front-end and back-end. Originally, our idea was to have George, Andrew, and Koen working on the back-end while Ludger was left with the task of creating a template web page for our reviews. It would have made more sense to first write down a preliminary to-do list before dividing ourselves, because we did not know exactly how much work/time was needed to be allocated for tasks. Thus, splitting ourselves prematurely caused a small bit of disorganization. This however, did not stop from meeting our development goals.

3.3 The Bad

There was not any outright bad aspects that we encountered for this iteration. The worst thing that did occur was that we first tried to tackle MVP with the idea that it should encompass as many schools, professors, and faculties as possible. Instead we decided to put are focus on a proof of concept where a person can review exactly one university(RUG). More will be discussed on this topic in the next section.

3.4 Changes and Future Development

As mentioned above, we decided to boil down our MVP to exactly one University. No professors or faculties were included. This let us focus on getting the fundamentals right (setting up the API and front end properly) before focusing on a more broader scope. It also alleviated the problem of finding out a way to add more things to review. We believe that although it is important to have this feature, it is not needed for the MVP. We decided that it was best to tackle this problem after we had our proof of concept. For the future, we to wish expand our product to accommodate more than the just students at the RUG. On top of this we will look to add other important features such as having a way to actually discern helpful reviews from spam and unhelpful reviews.

4 User Stories

Note that for these stories, it is assumed that a major feature has 1.0 story points, as elaborated on earlier in this document, and all other story point values are relative to this. All story point values shown here are the result of a modified story point poker in which the developers each submitted a floating point value, and the average of these values was rounded to the nearest tenth.

1. As a reviewer, I want to be able to write reviews for universities, courses, and professors because I want to give others insight into what it is like to work with that entity.

- Story Points: 1.5

Acceptance Criteria: This story is done when a person who wants to review a university can write up a review in a text field provided, click a submit button, and the review is displayed on the page for everyone viewing the page to see. For the MVP this is only going to be available for one university(RUG).

2. As a reviewer, I want to be able to rate universities, courses, and professors because it is a simple way to summarize one of those entities' reviews.

- Story Points: 0.9

Acceptance Criteria: This story is done when a reviewer is able to select a rating(1-10) and click a submit button. After which, the new rating is displayed on the page. For the MVP this is only going to be available for one university(RUG).

3. As a reader, I want to be able to mark certain reviews as helpful, or not helpful, because this helps the community in filtering out reviews that don't contribute to a meaningful conversation.

- Story Points: 0.8

Acceptance Criteria: This story is done when a reader can click a button to toggle whether or not the review was helpful. Once the reader has toggled the button, it will add to the ratio of people who thought the review was helpful or not.

4. As a reader, I want to be able to view the reviews for a university, course, or professor, because that is a logical thing to do.

- Story Points: 0.8

Acceptance Criteria: This story is done when a reader can click or load up a page for a particular topic and reviews are displayed for that topic.

5. As a reader, I want to see an overview of the different ratings that people have given something, just like on Amazon or EBay, because this makes it quicker to know the general sentiment about something.

- Story Points: 0.4

Acceptance Criteria: This story is done when a reader can see the average rating of a particular topic on that topics page.

6. As a reader, I want to be able to read **every** review given for a particular university, course, or professor, because that can help me make an informed decision. (This means using pagination or script-based page lazy loading.)

- Story Points: 0.4

Acceptance Criteria: This story is done when a reader can scroll down on a web page until they have seen all the reviews for that particular topic and all the reviews have been loaded.

7. As a reviewer, I want to be able to delete my review at any time, because I would like to have more freedom regarding data that I generate.

- Story Points: 0.3

Acceptance Criteria: This story is done when an author of a particular review can click a button, and the review page in question doesn't not show the review and the review is removed from the database.