

CM2020: Agile Software Projects  
Final Project Report, Group 6 (Tutor Group 1)

B. Currey, B. Lazaruk, A. Muralidharan, A. Milosevic & H.C. Ishida

4th March 2021

## **Contents**

### **List Of Figures**

### **List Of Tables**

# Reference To Rubric

The table below provides references to the relevant sections where specific grading criteria can be found in this report. A live version of the application is hosted at <https://tbd>.

Criteria	Location in Report
<b>Technical</b>	
Code solves challenges presented in the aims/objectives	
Iterative Design	??, ??
Approach is fit for purpose without drastic oversights	??
Elegance/aesthetics/readability of code	??
Sensible approach to structure	??
Evidence of collaboration and team work	??
Evidence of milestones and reflective updates	??
Difficulty of technical challenge	??
Novelty of technical challenge	??
<b>Testing</b>	
All parts of the system working	??
UI evaluation	??
Well documented tests	??
Effective error handling	
Systematic testing regime including design of appropriate test cases	??
Justification of testing methods	??
User evaluation involving representative stakeholders formative	??
User evaluation involving representative stakeholders summative	??
<b>Report</b>	
Good introduction and fair discussion of literature	??
Clear statement of problem and effective problem analysis	??
Clear structure	see Table of Contents
Justification of design decisions	??
Good argumentation and justification of claims/problem analysis	??
Clear documentation and user guide	??
Sufficient and appropriate references, and good citing method	
Good layout and formatting, especially of tables, figures, formulae and code examples	
Well written with systematic analysis	
Insightful discussion of results	
Evaluation of own work in relation to original proposal and plan	
Conclusion and discussion of future work	

# 1 Background, Aims And Objectives

## 1.1 Introduction And Literature

This project aims to develop a grades leaderboard for students to share grades. Before diving further into the context and objective of our project, we will briefly describe the most important literature used in developing this work.

The academic foundations of this project are drawn from [?] as a primary source for design methodology. This text describes the double diamond design approach, which was at the heart of our approach. It allowed us to widen our scope at the start of the project, yet letting us quickly select a high-potential idea for further development. We critically considered this approach alongside others, as will be discussed further in ?. Our testing methodology during development was based on the UMUX-Lite whitepaper [?], which enabled collecting user feedback on the user experience with little effort on behalf of test subjects, yet still allowing us to assemble a quantitative view on user feedback during summative assessment, as described further in ?. Our user interface has been largely designed using principles from [?], which outlines effective communication of functionality and usability of applications. We used this extensively in designing buttons, UI elements and descriptions as outlined in ??.

## 1.2 Context And Problem Statement

Motivation to study is a scarce resource for university students and one that can easily get depleted. Today's connected and highly technological environment provides the opportunity to use gamification to increase motivation and provide to students and to the educational institutions that they partake in [?] [?].

In addition, students with a competitive mindset might draw additional motivation from the ability to compare themselves to peers and identify opportunities to improve their own skills.

Every module in a degree program is different and may require a tailored strategy for studying. Reflecting on a completed module provides "hindsight 20/20" notes on what worked well and what can be improved. In a collaborative environment however, students are not limited to their own experiences - they can benefit from the wisdom and experience of fellow students.

The **problem** that we aim to solve is that students normally have no way to easily compare their own grades to those of their peers, and cannot assess if their grade is within an expected range for any given module. For prospective students of a module, it is hard to judge if a certain module usually provides high grades or not, and what effort might be associated with achieving such a grade. For students who have received grades, there is little opportunity to find out whether a received grade is reasonable given the self-perceived effort and completed coursework.

## 1.3 Opportunity

We aim to motivate the students to be their best selves and adopt the life-long growth mindset.

Our review of academic literature found evidence for positive effect of use of leaderboards on student motivation [?] [?].

To realise this long-term aim, we propose development of a **grades leaderboard** that allows students to share and compare grades, while managing their own grades for a given university program.

This report outlines how we decided to pursue this opportunity, what we aim to achieve, how we aim to achieve it and how we will measure outcomes.

## 1.4 Goals

With this project, we are pursuing the following goals. Details of critical success factors that measure achievement of these goals are provided in section 6.

### Business Goals

- Launch a grades leaderboard in one program with real students
- Gain traction so that the solution becomes an established part of student life

### Usability Goals

- Make entering and sharing feel extremely fast and frictionless
- Make students feel safe in sharing their grades online

### Technical Goals

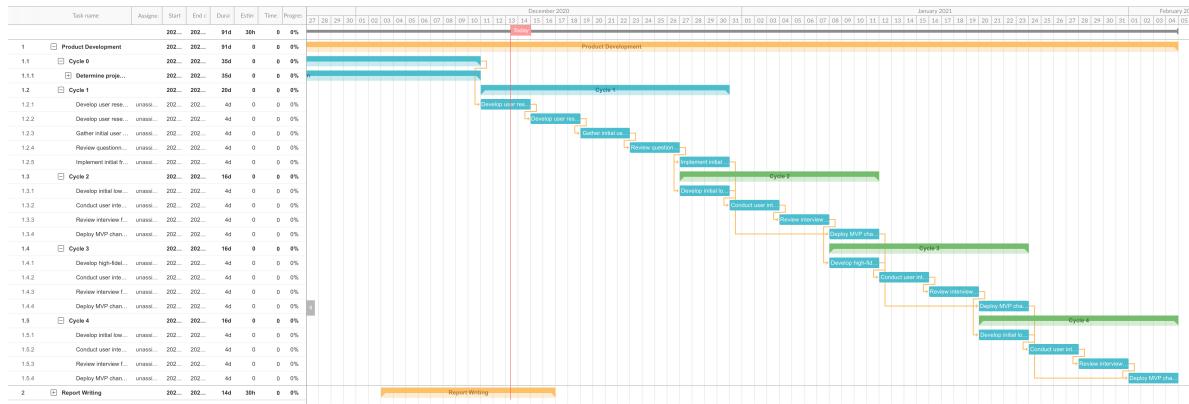
- Ensure the security of the platform to ensure integrity of the data provided
- Make the application easily accessible as web application

## 2 Planning And Research

### 2.1 Planning

We chose an iterative implementation approach that fits multiple cycles of design, development and user testing from the start of the year until submission of the project by mid-March. This approach envisions that based on completed initial requirements analysis by the end of the year, we can start with short, 2-week cycles of developing the solution. This phase will be closer to a Google Design Sprint in which the team spends focused, full days in developing solutions, implementing prototypes and iterating on the product.

Our timeline plan is shown in ?? illustrating the cycle-based approach until March of 2021.



**Figure 1.** Illustration of the cycle-based project plan

### 2.1.1 [BRAD] Milestones And Plan Updates

The main update to our working approach after starting the project was that we switched from two-week sprints to one-week sprints to ensure closer control and ensure delivery by the deadline, given the comparatively short project timeline. We decided to plan our sprints accordingly.

[BRAD] Can you create some kind of update to the Gantt chart?

## 2.2 Team Roles

We used SCRUM as a methodology to assign basic team roles to work in an agile manner. The key roles involved a product owner, a scrum master, individual contributors and a QA lead.

In addition, all team members took on the role of representing personas for testing purposes to ensure that we were not developing for ourselves, but rather for our users.

In terms of strict SCRUM roles, the team was allocated as follows (however, our responsibilities were treated fluidly as per the project's needs and in adherence to agile principles):

1. **Aleksandar:** SCRUM master
2. **Arjun:** Product owner
3. **Blair:** Individual contributor
4. **Brad:** QA lead
5. **Hayato:** Individual contributor

Within the team, our contributions were as follows: **Aleksandar** led definition of the technical architecture and software approach, and also led the implementation of user feedback specifically around the visual design. **Brad** led development of all tests, including the implementation of automated testing,

and also contributed with developing parts of the application. **Blair** led much of the core application development including the Slack authentication layer. **Hayato** worked on developing core parts of the application such as the grade leaderboards. **Arjun** supported with facilitating design workshops at the start, and later with some application development such as the grade calculation algorithm and documentation for the project.

## 2.3 Market Benchmarking

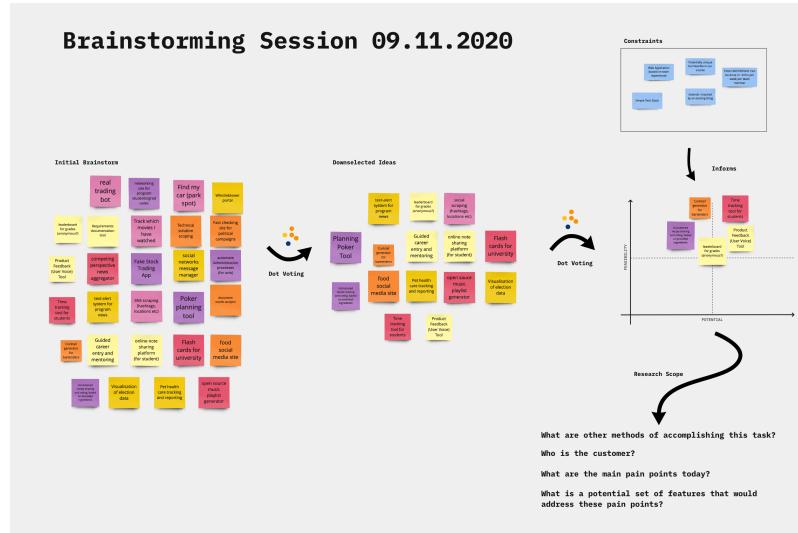
Initial desk research indicates that the selected project, a grades leaderboard, does not have a direct competitor. There are some utilities which partially solve the problem we are tackling - namely, tracking grades - but none of these enable the user to compare themselves to their peers, nor aggregates grade data. As such, we must cast a wide net to analyze our place in the market and consider *alternative solutions*. We have identified utilities which overlap with our project in terms of possible user needs as well as tools which solve analogous problems in different domains in order to get a more comprehensive picture of where our product stands in the market.

Our solution offers two mains functions: **tracking grades** and **sharing/comparing** grades. For each feature, we have documented the alternative solutions, their specific functionality and notable design choices.

### 2.3.1 Grade Tracking

We surveyed various existing tools that enable students to track grades. Some peer-created utilities and University of London portal were discovered through our teammates' participation in this course, while `gpacalculator.io` [?] is an example of many similar utilities that were found by searching 'grade tracking' and 'grade calculator' in a web search.

**Peer Student's Python Command Line Grade Calculator** A peer student within the BSc Computer Science program at University of London developed a command line grades calculator [?]. Users record their grades in a JSON file and run the script from the command line to see their grades and other relevant information, as shown in ??.



**Figure 2.** Python Calculator Sample Output

**Notable Features** The tool allows tracking course-level and cumulative grades and converts to ECTS (Europe) and US scales. It also supports conversion from US to UK GPA scores and determines which classification your grades fall into. Further, it allows tracking how many credits out of 360 were completed.

**Notable Design Choices** The design is lightweight as it can be installed locally and run from a command line. The input is provided via a JSON file, which also allows easily storing, versioning and sharing a set of grades.

**Peer's Google Sheets Degree Planner** Another student (and co-author of this paper) has published a spreadsheet-based tool to track and calculate grades [?]. Users record their grades in a Google sheet to track their credits and grades, as shown in ??.

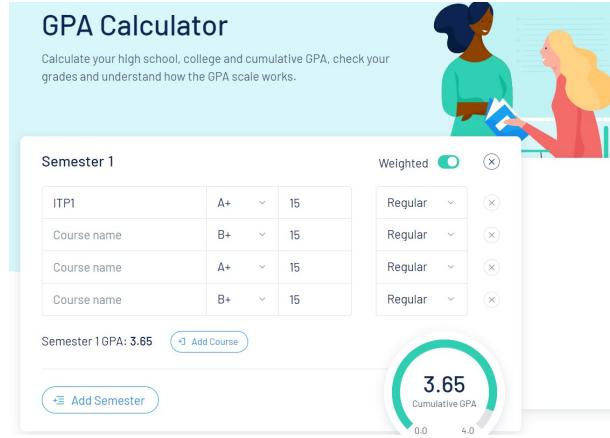
Module	Weighting	Weighted Credits	Status	Grade	Weighted Grade	% of Final Grade
Introduction to Programming I	1	15	Done	100%	100.00%	1.41%
Introduction to Programming II	1	15	Done	87%	87.00%	1.41%
Numerical Mathematics	1	15	Done	88%	88.00%	1.41%
Discrete Mathematics	1	15	Done	97%	97.00%	1.41%
					<b>90.31%</b>	<b>100.00%</b>

**Figure 3.** Google Sheets Degree Planner Excerpt

**Notable Features** The tool allows tracking course-level and cumulative grades, allows tracking how many credits out of 360 were completed and also highlights which modules are optional. It allows planning modules across semesters in order to distribute the course load.

**Notable Design Choices** This tool is implemented with Google Sheets, which makes it easier to use for many people, and allows easy sharing of the tool as anyone has free access to the software without any local installation.

**gpacalculator.io** A publicly available website, GPA Calculator [?], allows users to enter their grades in a web form and see cumulative results, converted to various other grading standards.

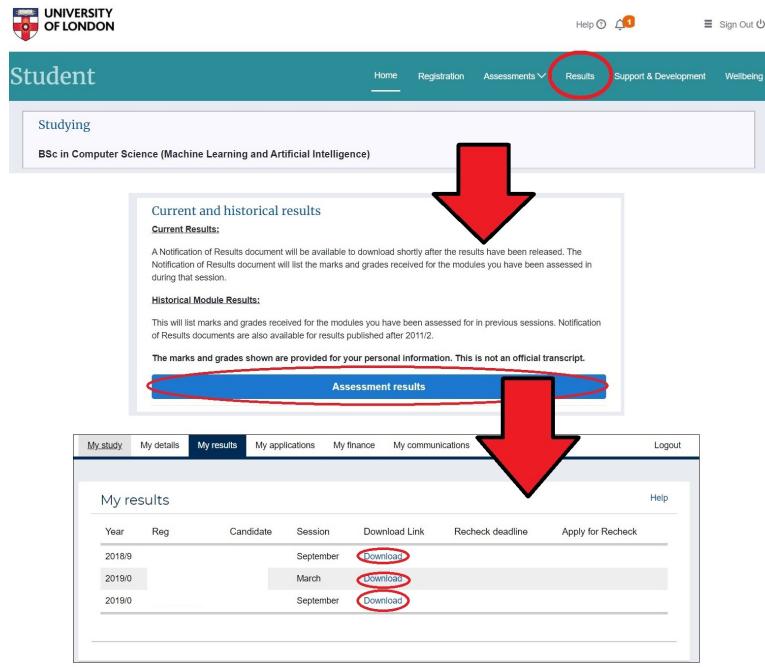


**Figure 4.** Landing Page Form on *gpacalculator.io*

**Notable Features** The tool calculates semester or total cumulative GPA from individual courses, offers different calculators for high school and college and also provides general information about GPA, grades, and different honors.

**Notable Design Choices** Placement of the input form is very prominent and users are immediately confronted with it when visiting the page. There is no login or data persistence available, and the tool is US-specific.

**University Of London Portal** The University of London itself provides access to grades for viewing [?]. Users can view grades from each semester, broken down by source (coursework vs. exam), as shown in ??



**Figure 5.** Path to find grade download links on University portal

**Notable Features** The student portal displays official grade results for each semester and shows granular grades for each exam portion separately.

**Notable Design Choices** Grades are not accessible easily, as they are buried in menus, and there is no centralized display of all grades or cumulative grades. A PDF download is required for all but the most recent grades. The results have the highest level of trustworthiness since they are official.

### 2.3.2 Grade Sharing

We searched the web for 'grade sharing,' 'grade comparing,' 'compare grades to other students,' and other derivatives but found no relevant tools. Glassdoor was suggested as an analogous tool during a brainstorming session and the University of London Slack workspace is known to us through our participation in the degree program.

**University Of London Slack Workspace** Students at the University of London's online BSc in Computer Science degree use a Slack workspace to collaborate [?]. Within this workspace, students enthusiastically share and query grade information following release of results, as shown in ???. This workspace is not publicly available.



**Figure 6.** Students sharing midterm results on Slack

This is the Slack space open to all University of London online CS students. There are channels for each course. Grades are sometimes shared in threads after they are released.

**Notable Features** Slack allows very quick impromptu sharing of grades, however on an irregular basis in bursts during grades season, and the grades are not solicited. Other students can weigh in and react or respond to grades and associated grade queries.

**Notable Design Choices** Grades reported are usually biased towards students with high grades, or people with low grades who suspect an error in their results.

**glassdoor.com** Glassdoor is a review and reporting website for employers [?]. It collects textual reviews and salary data from users. Salary reports are analogous to grade reports in that they are personal information about professional/academic performance that have a similar level of sensitivity.



**Figure 7.** Example of data and graph shown on glassdoor

**Notable Features** The site allows users to anonymously submit company reviews and salary information (including job title, locations of position, and years of experience). It contains job postings and company profiles with generic information. Users can also submit and collect job interview questions, and the site displays aggregate salary information for specific job titles.

**Notable Design Choices** Users must submit a salary entry in order to view detailed salary information from other users. The salary collection form is not very prominent and not highlighted on the main site. Users can define their job title, often creating close duplicates of job titles.

### 2.3.3 Summary Of Market Benchmarks

We abstracted a common set of functionality across these solutions as shown in ?? for grade tracking solutions, and ?? for grade sharing solutions.

	Python Grade Calc.	Degree Planner	gpacalculator.io	UoL
Track Course Grades	✓	✓	✓	✓
Calculate Cumulative Grades	✓	✓	✓	
Track Credits	✓	✓	✓	✓
Granular Grades				✓
Convert Scales	✓			
Determine Honors Level	✓			
Persistence	✓	✓		✓
Shareable		✓		

**Table 1.** *Grades Tracking Comparison*

	UoL	Slack	glassdoor.com
Incentivized to submit data		✓	
Can compare to others	✓		
Easy to reference		✓	
Aggregates data		✓	
Includes profile on data source		✓	

**Table 2.** *Grades Sharing Comparison*

## 2.4 Market Positioning

In order to understand our solution's place in the market we need to examine the external factors which might affect the growth and performance of our application.

In the following section we perform both a STEEPLE [?] analysis as well as a SWOT [?] analysis to support identifying the correct market position.

#### **2.4.1 STEEPLE Analysis**

The STEEPLE analysis will allow us to analyse our problem space based on environmental factors and outlines some of the contextual consequences of our proposal.

##### **Social**

- People tend to be competitive [?]
- People may be hesitant to share if their grades are low or they feel they are unfairly graded

##### **Technological**

- Student's participation in the course ensures they will have the hardware necessary to access the website (computer or mobile and internet)

##### **Economic**

- Global audience means any potential efforts to monetize users would be subject to currency risk and local purchasing power
- Usage costs and any potential revenue would be related to enrollment in University of London program and size of other Universities

##### **Environmental**

- Server has energy requirements that will grow with usage
  - An increasing number of cloud providers offer solutions that use green energy

##### **Political**

- Regulation of data handling is increasing everywhere. This continued trend is exemplified by recent regulations enacted over the past two years:
  - California, USA's CCPA
  - EU's GDPR
  - Canada's PIPEDA
  - Japan's APPI
  - Brazil LGPD

Given the global nature of this degree program our application is sensitive to all of these policies and future data protection measures implemented worldwide

##### **Legal**

- Must comply with local data protection regulations listed above

### **Ethical**

- Must handle sensitive data responsibly and transparently
- Obligated to maintain a healthy sense of competition and steer away from toxicity and envy
- Honest reporting of grades and integrity of reported data is critical

#### **2.4.2 SWOT Analysis**

The SWOT analysis supports the identification of our solution's underlying potential and associated risks.

#### **Strengths**

- Knowing you have a relatively high grade could be motivating / encouraging / validating.
- Validation if your grade was low but many people report similar grades.
- Highlights which courses may be more difficult and require additional focus.
- Helps students understand when their grades may be unjustifiably low, which may help in organizing a more detailed and thoughtful request to the institution to review the grades or rubric.
- Comparison aspect provides motivation for maintaining one's standing at or above a certain percentile.
- The leaderboard aspect "gamifies" grades and gives incentive to submit grades.
- Ability to see module "reviews" and retrospectives might be an additional motivator for students

#### **Weaknesses**

- Data reliability. We have no good way to validate data so we must dis-incentivize dishonesty and not assume 100% accuracy. This might be solved with a voluntary "random check" participation where students need to prove that they actually received these grades
- A small sample size of participants will limit usefulness of averages.
- Those with poor marks or marks they feel are unjustified will likely be less willing to share.
- Those who fail or drop out may be unable (depending on integration) or unwilling to participate, which could impact grade averages.
- Without a representative sample of students the averaged data, graphs, etc will be skewed and could lead to reduced acceptance or usage of the tool.
- If we report anonymously, do we also store anonymously? If so then do we provide for the ability to remove data on request?

## Opportunities

- This project has few real competitors. There are some student created utilities but grade sharing/comparing is outside the scope of those utilities.
- The program size will continue to grow for a long time. We haven't reached the maximum number of cohorts and the recent cohorts have grown. The need exists, there just wasn't a reason to make it until now.
- If we are the first to make this, we can capture the interested user-base which should be a large moat.
- Slack integration allows us to naturally link to the community of students that already exists, leveraging the University's policing of that population to being actual students.
- Retrospective tips can be used by future cohorts to gain a studying advantage for the course
- Average grades per course, per semester, etc can similarly help future students know what to expect and perhaps how much they need to prepare

## Threats

- Time. Other student utilities show recognition of value for this type of product. Probably only a matter of time until someone else extends the grade idea in a similar fashion.
- Leak or theft of personally identifiable grade and contact information.
- People could poison the data with dishonest grade reporting.
- Insecure design could compromise data integrity and give people access to sensitive information.
- Loss of integration / changes to any APIs we're using (such as Slack).
- Data loss.

## 2.5 Customer Survey

As of this report, the team has designed the high-level questions that should be answered by data recording. Our key research questions are listed below. Note that these are not the actual survey questions, but rather the questions we aim to answer through our data recording and analysis.

- How do students currently keep track of grades?
- How comfortable are students with sharing their grades?
- How honest will people be with reporting their grades, anonymously or not?
- Why would students want to compare their grades with other students?
- Why would students want to see retrospective tips left by other students?

- Why is this different from just getting the (presumably anonymized) values from the University?  
Why is this better?
- Would there be value in providing automated conversion of grades from the UK standard to standards from other countries?
- Does anonymity encourage or hinder contribution to the leaderboards? Does it affect accuracy of reporting?

Data gathering will be conducted via an online survey among university students, and interviews will be conducted via online video calls with selected university students.

### 2.5.1 Online Survey - Design Choices

We used the online survey to gather quantitative data. Not all of our key research questions can easily be answered quantitatively so we will seek to answer these in the student interviews.

We created the survey by converting key research question to a survey question designed to elicit the required answers where able. We did not require any questions to submit the survey because we felt it was important to ensure reliable answers and wanted to ensure the highest possible response rate. The survey was only 8 questions and took approximately 2 minutes to fill out in order to encourage a higher response rate. We provided context of our project before asking the following questions.

We posted the survey in this course's *general* slack channel. The survey respondents are the potential users so the responses can be considered representative of our real users.

### 2.5.2 Online Survey - Questions And Results

Following are the questions we asked to start collecting data on our key research questions, followed by the results.

**1. Key Research Question:** How do students currently keep track of grades?

**1a. Survey Question:** What tools do you use to keep track of grades?

**Results:**

Responses	
<b>University of London Portal</b>	68.4%
<b>Outside Tool</b>	47.3%
<b>None</b>	15.8

**Table 3.** *What tools do you use to keep track of grades?*

**2. Key Research Question:** How comfortable are students with sharing their grades?

**2a. Survey Question:** Have you ever shared your overall course or individual assessment grades in slack?

Responses	
Yes	52.6%
No	47.4%

**Table 4.** *Have you ever shared your overall course or individual assessment grades in slack?*

This question helps us establish a behavioral baseline for student comfort in sharing grades. We ask about past behavior to get the most objective answers possible. We also surveyed students for their sentiment on this key research question.

**2b. Survey Question:** How comfortable are you with sharing grades with other students?

Responses	
At least somewhat comfortable	79.9%
Not comfortable	21.1%

**Table 5.** *How comfortable are you with sharing grades with other students?*

**2c. Survey Question:** What would make you more comfortable sharing grades?

Results:

Responses	
Anonymity	73.7%
Evidence of Privacy Protections	52.6%
Seeing other students using tool	47.4%
Knowing grades can only be viewed by students	36.8%
Open source with transparent developers	31.6%
University endorsement	21.1%

**Table 6.** *What would make you more comfortable sharing grades?*

**3. Key Research Question:** How honest will people be with reporting their grades, anonymously or not?

**3a. Survey Question:** How would you feel about submitting a low grade anonymously?

Responses	
Indifferent	73.7%
Good	26.3%

**Table 7.** *How would you feel about submitting a low grade anonymously?*

**3b. Survey Question:** How would you feel about submitting a low grade publicly?

Responses	
Indifferent	78.9%
Bad	21.1%

**Table 8.** *How would you feel about submitting a low grade publicly?*

**4. Key Research Question:** Why would students want to compare their grades with other students?

**4a. Survey Question:** Please rate to what degree each of the following are important reasons for contributing grades. 1 is not important, 5 is very important.

**Results:** To determine the most important reasons, we consider a 4 or 5 an "endorsement." The endorsements ranked from high to low:

Endorsements	
See Where I Rank Against Other Students	13
Track Grades	10
Keep University Grading Curve Transparent	9
Help Upcoming Cohorts	6
Recognize High Achievers	5

**Table 9.** *Most important reasons for contributing grades*

**5. Key Research Question:** Why would students want to see retrospective tips left by other students?

**5a. Survey Question:** Please rate to what degree each of the following are important reasons for leaving retrospective tips for courses. 1 is not important, 5 is very important.

**Results:** Like the previous question, we considered a 4 or 5 an "endorsement." The endorsements ranked from high to low:

Endorsements	
<b>Learn from wisdom of others</b>	15
<b>Assessing course expectations</b>	15
<b>Assessing effort course</b>	13
<b>Give back to community</b>	12
<b>Choosing courses to take</b>	12

**Table 10.** *Most important reasons for leaving retrospective tips for courses*

### 2.5.3 Online Survey Insights

?? shows a majority of respondents (52.6%) have shared their grades in slack. Slack is not anonymous and there is no explicit place to share so these students who shared can be considered highly motivated to do so. In addition, a large majority (79.9%) report feeling comfortable sharing their grades in an unqualified manner (neither anonymous or public specified), as shown in ???. In addition, ?? shows nearly 50% of respondents already use an external tool to calculate their grades. These responses strongly confirm that there is demand for a tool that allows people to share and compare grades.

Responses also reveal that anonymity is key. We see in ?? that a large majority (73.7%) of respondents said anonymity would make them more comfortable sharing grades. This was the single largest factor for making students more comfortable in sharing grades. In addition, 73.7% of respondents said they felt indifferent about sharing low grades anonymously and 0% reported feeling bad about it in ???. Whereas 78.9% reported that they would feel bad about sharing a low grade publicly and only 21.1% said they would feel indifferent in ???. Therefor anonymity is crucial for increasing participation and integrity of grades.

## 3 Prototyping And Iteration

### 3.1 Techniques

Our strategy towards prototyping will focus on creating a number of low-fidelity prototypes to quickly iterate on, while high-fidelity prototypes will consist of actual software that we can test. We will not create detailed up-front designs or clickable prototypes for the reason that our application's core functionality can easily be communicated with low-fidelity prototypes.

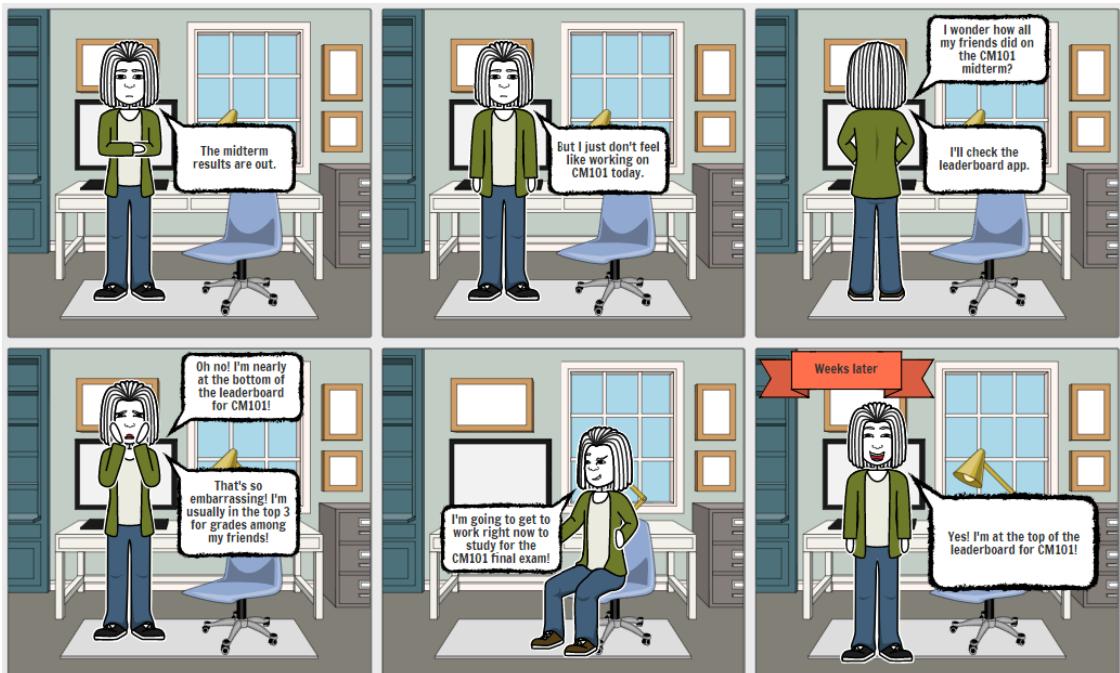
The reasoning for this approach lies in the economic principle of prototyping, which states that the best prototype is one that can demonstrate the possibilities and limitations of a solution in the most efficient manner.

### 3.2 Storyboards & Sketching

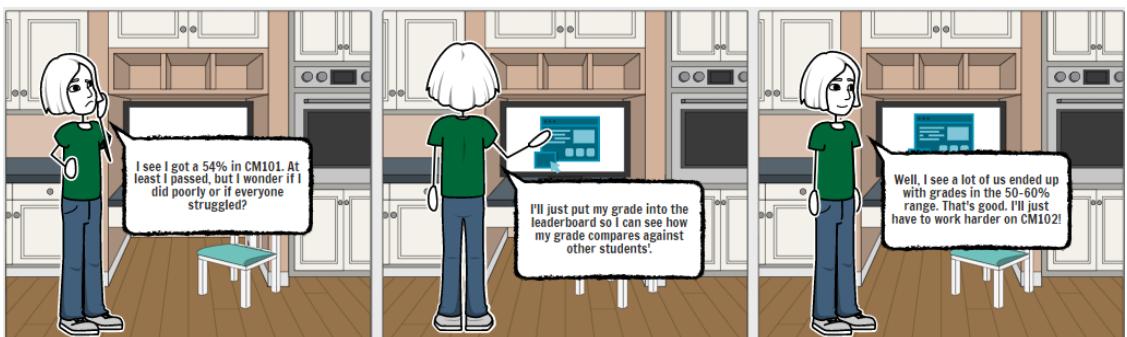
The team developed initial storyboards to illustrate the user journey and make the value proposition tangible. A series of sketches is embedded for the identified personas in ?? and ???. Based on these storyboards, we learned the following:

- The main motivation to log on is likely the need to track grades in a central, easily accessible location that also calculates averages automatically
- There are peak times during the year that the tool will see higher usage (e.g. when grades are released)
- Students are interested in the distribution of grades on a curve

The team will consider these factors in the development of the solution.



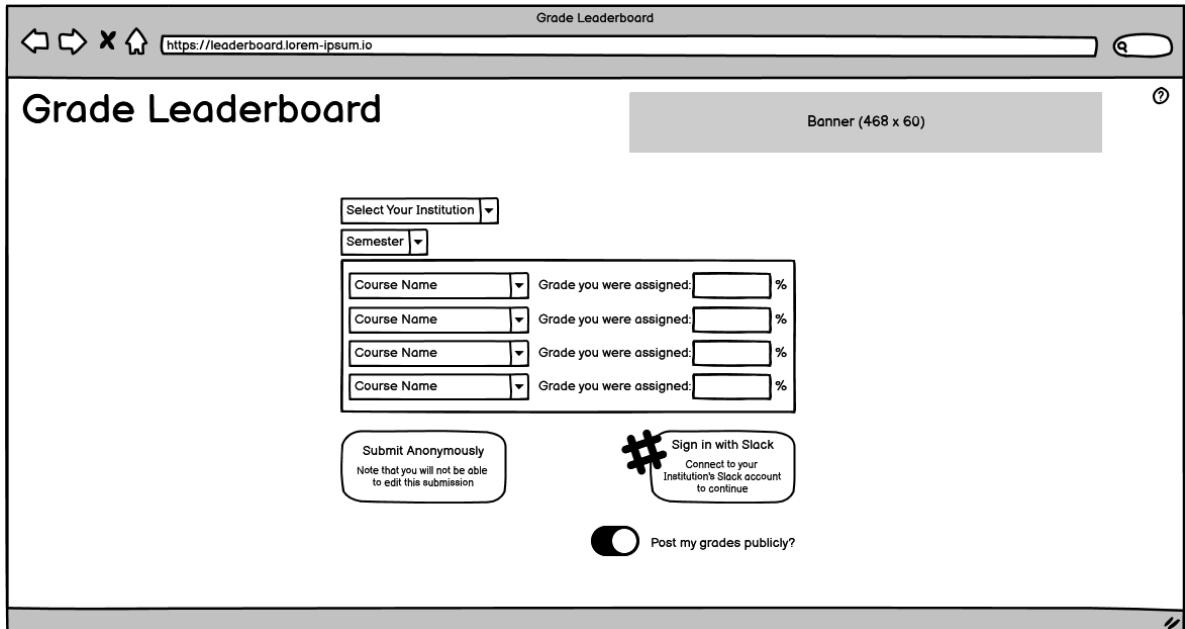
**Figure 8.** Initial storyboard for the persona “Competitive Topper”



**Figure 9.** Initial storyboard for the persona “Casual Checker”

### 3.3 Wireframes

To further illustrate how the application will work, an initial wireframe was created to show how an entry page to capture grades might look like. This wireframe was built using a wireframing tool (Balsamiq)[?] and will be iterated on as we move towards building the solution.



**Figure 10.** Early wireframe sketch

Based on this very early wireframe, the team identified important questions to consider for further iteration.

- Should users enter to see their own overview of grades, or an aggregate?
- How might users authenticate into the system?
- How does an aggregated view look?
- What other information might we need to collect from users?

As a result of these questions and collaboration further iterations were made to the early wireframe. As well, initial versions of a login page, a report view, and a grade leaderboard were developed by the group.

Grade Leaderboard

<https://leaderboard.lorem-ipsum.io>

# Grade Leaderboard

Ad Banner (468 x 60)

Semester ▾

Course Name ▾	Grade you were assigned: [ ] %
Course Name ▾	Grade you were assigned: [ ] %
Course Name ▾	Grade you were assigned: [ ] %
Course Name ▾	Grade you were assigned: [ ] %

Submit Anonymously  
Note that you will not be able to edit this submission

# Sign in with Slack  
Connect to your Institution's Slack account to continue

Post my grades publicly?

Figure 11. Wireframe after group iteration

Grade Leaderboard

<https://leaderboard.lorem-ipsum.io>

# Grade Leaderboard

Ad Banner (468 x 60)

### Sign Up with Email

\* Name: [ ]

\* Email: [ ]

\* Password: [ ] ⓘ

\* Re-type password: [ ]

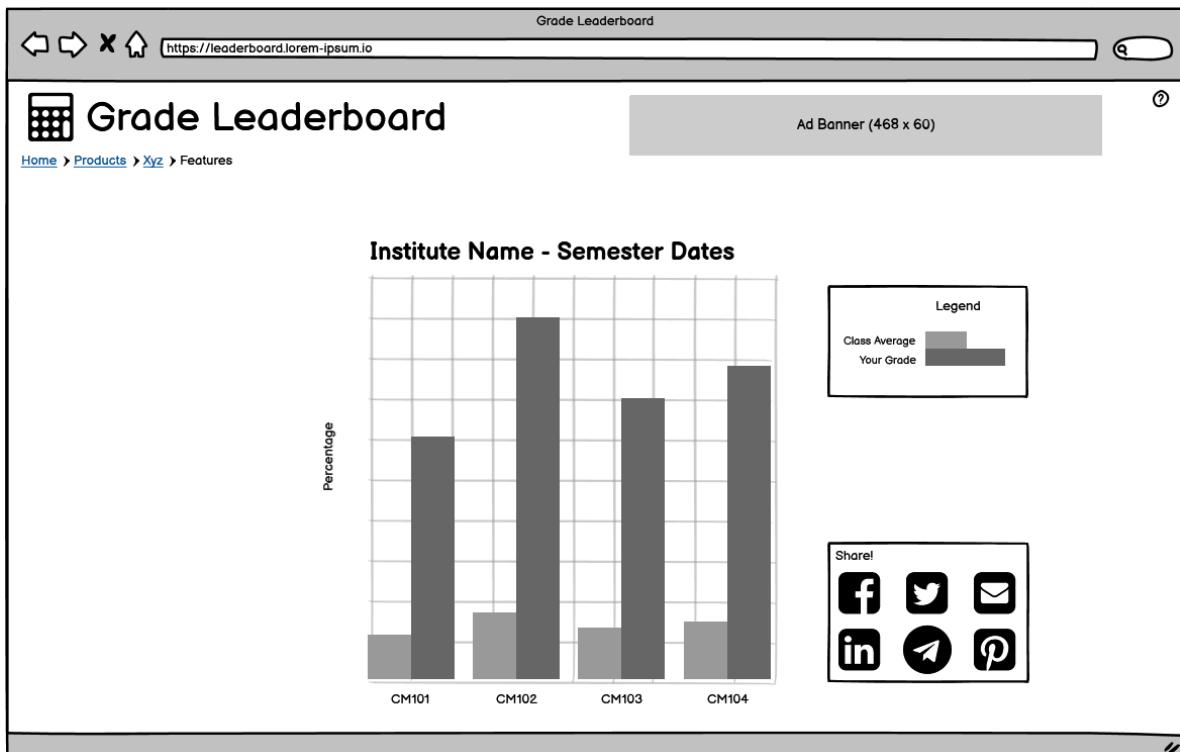
Select your Institution ▾

I agree to the [Terms of Use](#) and [Privacy Policy](#).

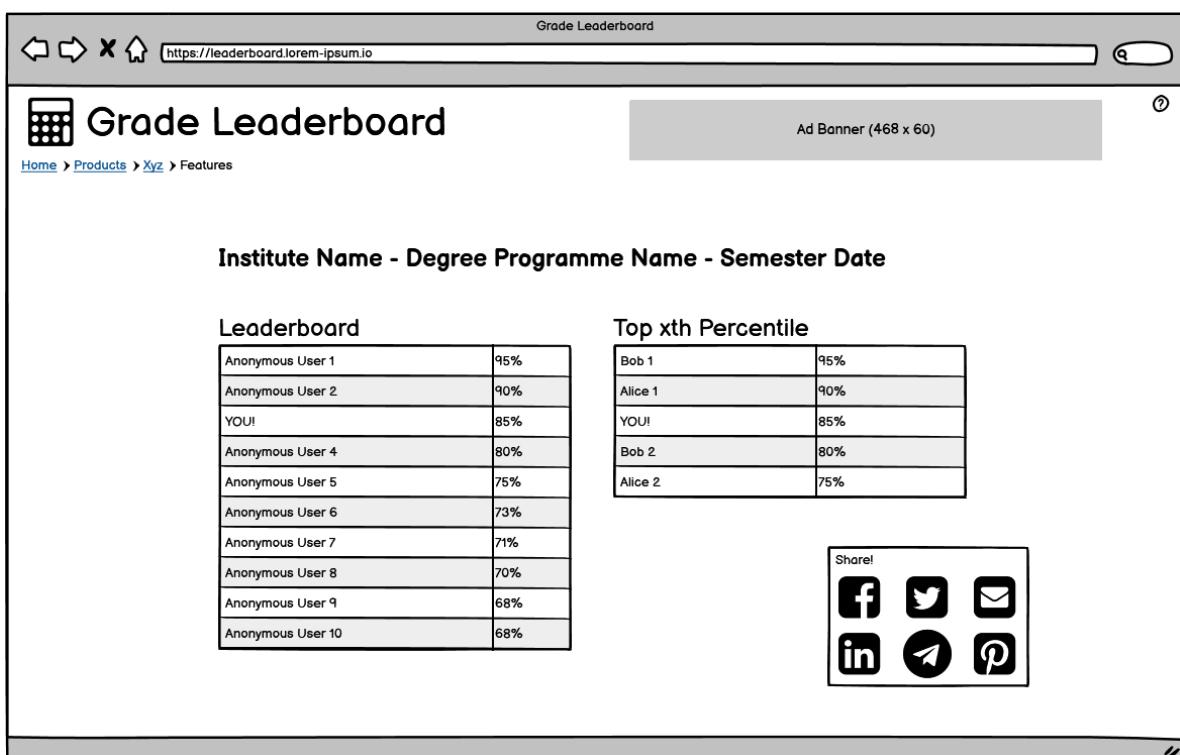
[Learn more](#)

# Sign in with Slack  
Connect to your Institution's Slack account to continue

Figure 12. Initial signin page wireframe



**Figure 13.** Early collaboration on report wireframe

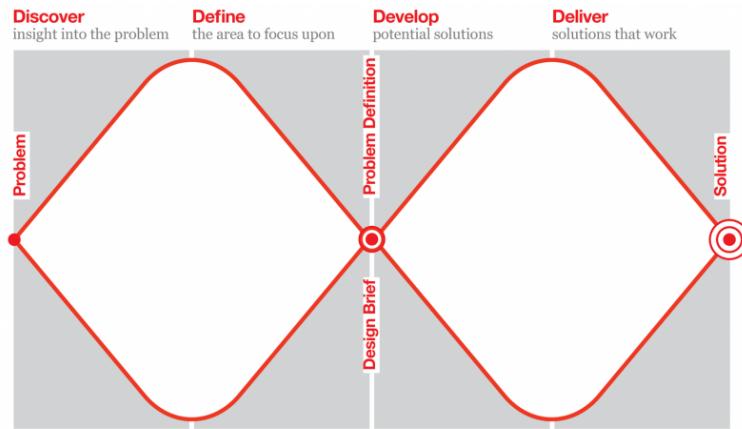


**Figure 14.** Early leaderboard wireframe

## 4 Design

### 4.1 Iterative Design

Requirements gathering for this project are based on a classical “double diamond” approach. In order to create this proposal, the project team has iterated once internally to gather a potential set of very high-level critical functionality to inform further data gathering. We explored potential topics of interest, narrowed it down to the topic of creating a grade comparison tool for university students, and then explored the problem space.



**Figure 15.** Double Diamond Design Approach

The reason why we chose the double diamond method is because it is widely used in Design Thinking [?] approaches when project ambiguity is high. The team felt this approach would lead to the highest level of clarity quickly, and would fit into the given timelines by the course schedule and the available capacities of the team.

We considered alternatives such as the Google Design Sprint, but eventually decided not to pursue this as we were not in the need of a 5-day, rapid iteration approach for our solution. We might use it partially during more concrete stages of the project to accelerate progress. We also considered "Research in the wild" (RITW) to find disruptive behaviour-changing ideas, however this approach felt inappropriate as our problem space did not indicate a fundamental challenge to existing technology, but much rather the fulfilment of a basic need of university students often not provided by universities.

### 4.2 Discovery

As background for how we arrived at our final idea, we initially conducted a number of broader exercises to discover potential problem spaces. The team conducted a “Crazy 8” exercise in which each team member developed at least 8 new ideas in a time span of a minute for each idea. This laid out a large space of potential ideas the team could work on.

## Brainstorming Session 09.11.2020

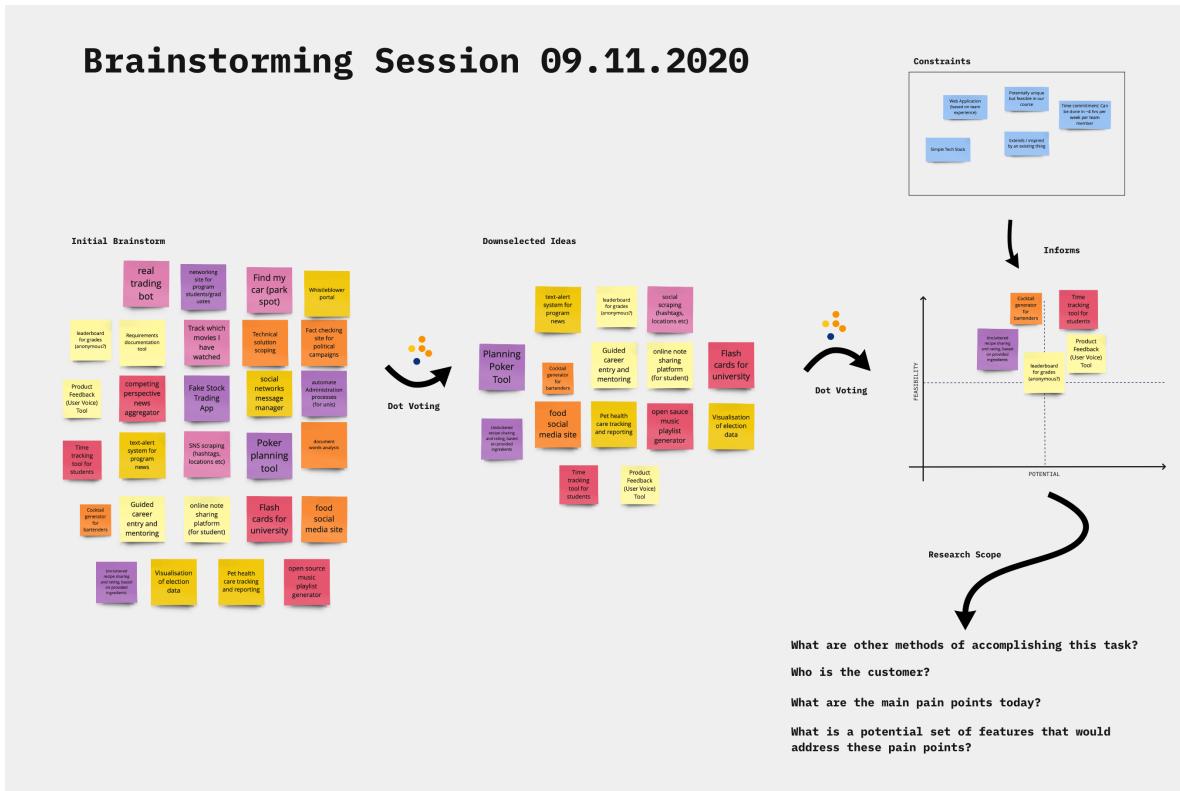
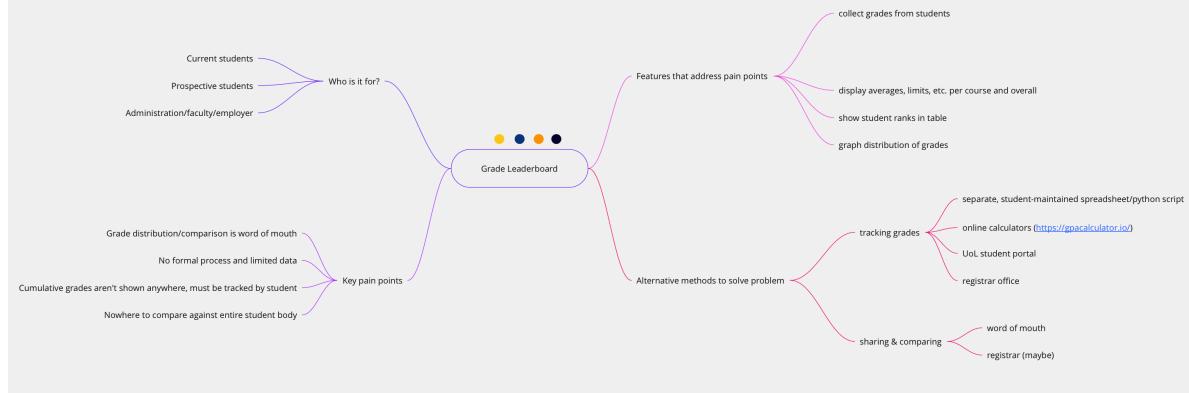


Figure 16. Illustration of initial discovery exercise

Next, we used dot-based voting to narrow down the ideas to those with the highest assumed potential for success and highest interest for the group to further explore.

Once we had narrowed the ideas down from over 40 to just 5, we decided to explore the potential problem space for each via mind maps, exploring the potential competitive landscape, possible functionality and target users.

## Project Selected 16.11.2020



**Figure 17.** Mindmap for the winning idea

To further explore the problem space of student grading, we began by brainstorming potential functionality.

### 4.3 High-Level Requirements Funnel

Our approach to gathering requirements will follow these phases, based on the approach for data gathering and requirements analysis described in [?].

1. **Data Gathering:** We first use two methods for data recording, namely *questionnaires* and *interviews*. These will serve as initial data input for the project. This will be aided by further desk research to inform the problem space, such as academic papers in the area.
2. **Data Analysis:** Analysis of data recorded in this manner will occur based on the analytic framework of *content analysis* and *grounded theory*. We will both develop quantitative and qualitative analysis based on the recorded data. We aim to formulate our **core hypotheses** around this analysis. At the point of writing this paper, only quantitative data had been gathered, with more input following in early 2021.
3. **Formulate requirements:** Finally, we develop requirements in the form of epics and user stories. To aid this process, we decided to use the frameworks of *contextual inquiry*, which allows us to discover requirements that inspire *joy of use* and *joy of life*, and the 7 product dimensions [?], which allow us to ensure broad coverage of important product attributes.

## 4.4 Initial Epics

Based on the survey results and research to date, we have identified initial epics, formulated in story format with acceptance criteria. The acceptance criteria would be further broken down into user stories that fulfill those criteria.

**Capture Grades** As a student, I want to enter a module grade, so that I can contribute to the data-base and help other students.

### Acceptance Criteria

- Only authenticated users can provide a grade
- Grades can be submitted anonymously
- Capturing a grade is fast and requires little input
- One student can only capture one grade per module
- Students can update a previously captured grade

**View Grades** As a student, I want to view all the grades for a specific module, so that I can see how I compare with other students.

### Acceptance Criteria

- Students can see their position on a leaderboard for the entire program
- Students can see their position on a leaderboard for a single module
- Students can see the distribution of grades for the entire program
- Students can see the distribution of grades for a single module

**Manage Personal Grades** As a student, I want to capture and manage all my grades in my program, so that I can see my program progress and calculate my final grade.

### Acceptance Criteria

- Students can view all their program grades in a single overview
- Students can assign modules to semesters that they plan to take
- Grades are used to calculate a program average using the correct weighting and credits
- Students can see their program progress as a percentage

**Submit Module Retrospectives** As a student, I want to submit a module retrospective, so that I can inform other students about my experience.

### **Acceptance Criteria**

- Students can enter a free-text retrospective assigned to a specific module
- Students can only enter one retrospective per module
- Students can assign ratings to effort and difficulty of a specific module
- Students can see retrospectives entered by them

**View Module Retrospectives** As a student, I want to read module retrospectives submitted by other students, so that I can make an informed choice about future modules.

### **Acceptance Criteria**

- Students can see a list of modules and sort it by rating criteria
- Students can read retrospectives entered for a specific module

As part of development, we will prioritize these epics and likely deliver on a subset of them in the initial release.

## **4.5 [ARJUN] Justification Of Design Decisions**

## **5 Testing**

### **5.1 General Methodology**

The team decided that the most appropriate evaluation type is in a **controlled settings directly involving users**. This means having users peruse the product under observation, describing their experience and providing live feedback. This will be the most efficient approach from a planning perspective.

A further reason this is more appropriate as opposed to *natural settings* is that grades are not something students submit and engage with on a daily basis, hence the opportunities to observe students in this process are rare. This will result in certain downsides of the controlled setting taking precedence, specifically not being able to identify behaviours of users that may occur when unsupervised, which may be central to the grades process. A future iteration may employ screen recording (such as Hotjar) or A/B testing to identify more hidden behaviours. An UMUX-Lite method for evaluating usability will be employed based on [?].

We will test the produced application at the end of each cycle with 1-3 real student users, recording the outcomes and discussing the results along four categories:

1. **Likes:** What did users enjoy about the product?
2. **Problems:** What problems did users encounter while using the product?

3. **Ideas:** What are new impulses that users provided while using the product?
4. **Questions:** What questions did users ask while using the product?

## 5.2 Formative Testing

### 5.2.1 Identifying Users

In order to identify users, we employ the technique of personas to help identify the right kind of user. We also employ *scenarios* and user fiction to describe the kind of experience we envision for end users.

The underlying hypothesis for our personas is that some users will be more interested in such a tool than others, and do so for different purposes. Our hypothesis is based on the observation that among our own cohort of students in the program, there is a smaller subset of users that are highly active in the student community, generally reporting better grades and mentoring other students, and appreciate some competition. We derive that there must also exist a majority of more casual users who are not as active, but still interested in aspects of grade comparison.

- **Competitive Topper:** Students who are generally top-of-the-class, who enjoy competition and are eager to hit the top of the leader board.
- **Casual Checker:** Students who are not very interested in competitive aspects, but would still like to know where they stand and if they are within the expected range of grades.

In order to actually identify users that match these personas, we are including specific questions in this regard in our survey to validate the underlying hypothesis.

### 5.2.2 Results

## 5.3 Summative Testing

### 5.3.1 [BRAD] Automated Acceptance Testing

As all members of the team were contributing to the Github repository, we wanted to implement some repeatable testing that could be quickly run to ensure the application was continuing to perform as expected. To accomplish this we implemented the Taiko browser automation library [?] and Gauge automation framework [?]. Taiko tests were written to simulate human use of our application. We also endeavoured to make the steps as small and modular as possible, for more effective re-use. We wrote tests that covered a wide range of actions that a user would be expected to be able to try to do, including:

- clicking the log in button
- entering login information as retrieved from system variables
- selecting a course or semester by name

- entering a value in a text field
- waiting for certain text to be visible
- checking that the user avatar was visible in the navigation bar
- checking that the user avatar was visible in the leaderboard

We also wrote Taiko steps to validate that the application was responding as intended to both valid and invalid entries from the automated "user". This included validations that error messages were being displayed when appropriate, and a step to make sure that sorting of the users was being performed in the manner we intended it to be done. The tests were written to allow arguments to be passed if necessary, enabling us to quickly and easily change some of the test conditions for more in-depth testing.

We wrote Gauge automation test specifications to run multiple Taiko test steps. This allowed us to repeatedly test the application in a more end to end manner by re-running all the acceptance tests after changing the application. This provided a greater confidence that the change made to the code had not affected another part of the code unintentionally.

Gauge automation was configured to test a number of scenarios including:

- end to end logging into the application through Slack
- selecting a course and session, entering valid grades, and confirming that the grade was accepted
- selecting a course and session, entering invalid grades, and confirming that error messages were visible to the user
- editing a previously entered grade and confirming that change to the database was successful

Much of the testing was done with the software in "headless" mode which made the operation very fast. The generated logs allowed the developer to quickly confirm that the application was still performing as expected, as shown in ?? or if something was wrong, as shown in ??.

**Figure 18.** Successful Gauge Test Log

```
PS C:\Users\brad\GitHub\uol-agile-group-project\test> npm test

> gauge-taiko-template@ test C:\Users\brad\GitHub\uol-agile-group-project\test
> gauge run specs/

# Test Gradez
## View default page    P
## Sign in with Slack   P P P P P P P
## View personal grades page  P P
## Try to add invalid grades to Graphics Programming in October 2021 and verify they were not accepted
P P P P P P P P P P
## Try to add grade that was already entered and verify that module is not available in list  P P
## Edit Existing Grade Entered In Previous Step      P P P P P P P P P
## Confirm tied grades are sorted by timestamp      P P P
## Verify user avatar exists in navbar        P P
## Verify user avatar exists in leaderboard    P P P
## Verify Cumulative Grade and Completion Rate are visible    P P P P

Successfully generated html-report to => C:\Users\brad\GitHub\uol-agile-group-project\test\reports\ht
1

Specifications: 1 executed      1 passed      0 failed      0 skipped
Scenarios:      10 executed     10 passed     0 failed      0 skipped

Total time taken: 42.607s
```

**Figure 19.** Failed Gauge Test Log

```
PS C:\Users\brad\GitHub\uol-agile-group-project\test> npm test

> gauge-taiko-template@ test C:\Users\brad\GitHub\uol-agile-group-project\test
> gauge run specs

# Test Gradez
## View default page    P
## Sign in with Slack   P P P P P P P
## View personal grades page  P P
## Try to add invalid grades to Graphics Programming in October 2021 and verify they were not accepted
Failed Step: Select "Module" "CM2030"
Specification: specs\leaderboard.spec:23
Error Message: Error: Option CM2030 not available in drop down
Stacktrace:
Error: Option CM2030 not available in drop down
    at node_modules\taiko\lib\elements\dropDown.js:171:15
    at async doActionAwaitingNavigation (node_modules\taiko\lib\doActionAwaitingNavigation.js:125:5)
    at async DropDown.select (node_modules\taiko\lib\elements\dropDown.js:125:5)
    at async DropDownWrapper.select (node_modules\taiko\lib\elementWrapper\dropDownWrapper.js:136:5)
    at async Object.<anonymous> (tests\leaderboard_step_implementations.js:136:5)
## Try to add grade that was already entered and verify that module is not available in list  P P
Failed Step: Select "Module" "CM2030"
Specification: specs\leaderboard.spec:39
Error Message: Error: Option CM2030 not available in drop down
Stacktrace:
Error: Option CM2030 not available in drop down
    at node_modules\taiko\lib\elements\dropDown.js:171:15
    at async doActionAwaitingNavigation (node_modules\taiko\lib\doActionAwaitingNavigation.js:125:5)
    at async DropDown.select (node_modules\taiko\lib\elements\dropDown.js:125:5)
    at async DropDownWrapper.select (node_modules\taiko\lib\elementWrapper\dropDownWrapper.js:136:5)
    at async Object.<anonymous> (tests\leaderboard_step_implementations.js:136:5)
## Edit Existing Grade Entered In Previous Step      P P P P P P P P P P
## Confirm tied grades are sorted by timestamp       P P P
## Verify user avatar exists in navbar        P P
## Verify user avatar exists in leaderboard     P P P
## Verify Cumulative Grade and Completion Rate are visible  P P P P

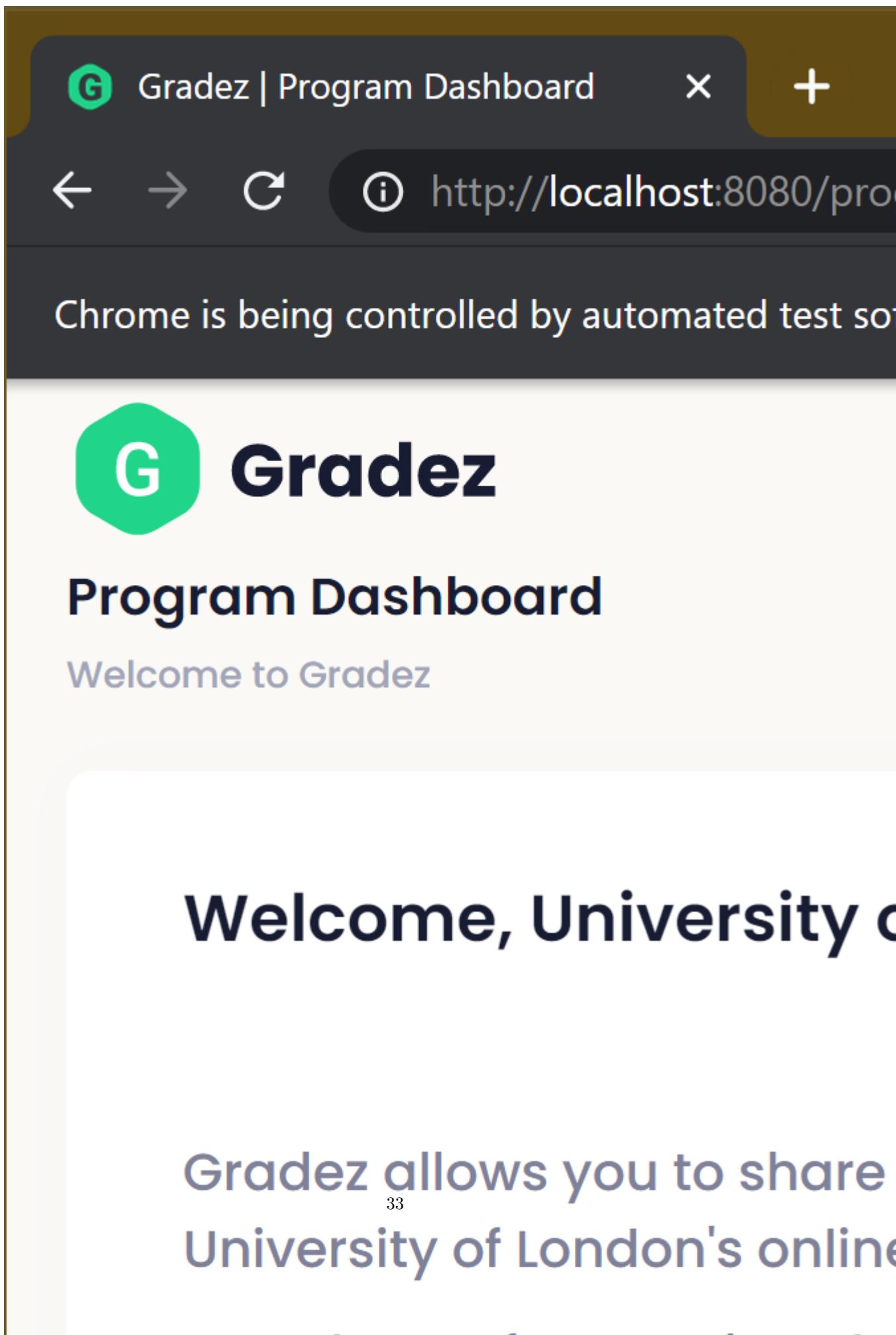
Successfully generated html-report to => C:\Users\brad\GitHub\uol-agile-group-project\test\reports\html
1

Specifications: 1 executed      0 passed      1 failed      0 skipped
Scenarios:      10 executed     8 passed      2 failed      0 skipped

Total time taken: 37.808s
npm ERR! Test failed. See above for more details.
```

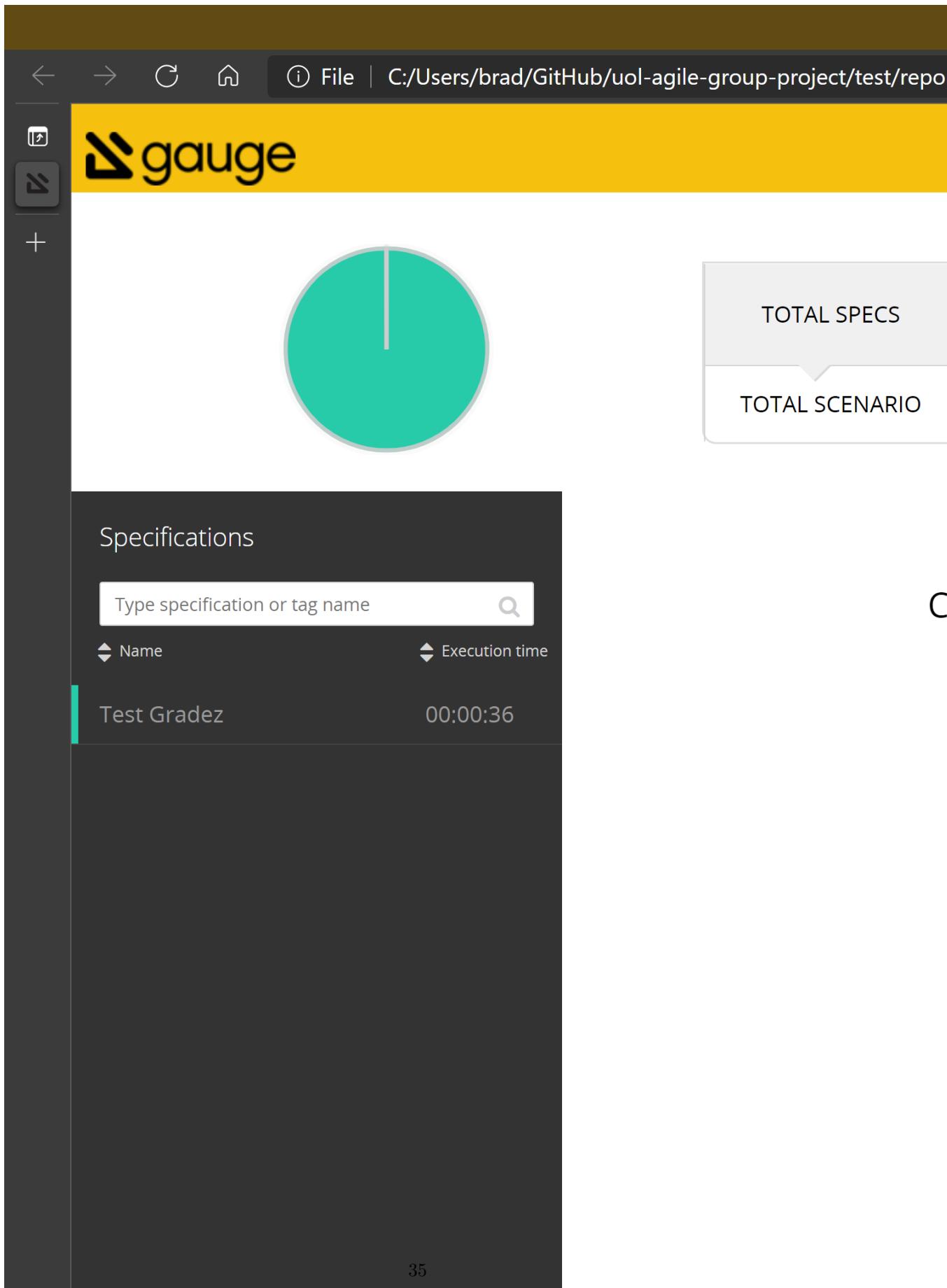
If necessary the software could be configured to show the automated browser window, as shown in ??.

**Figure 20.** Automated Browser Testing



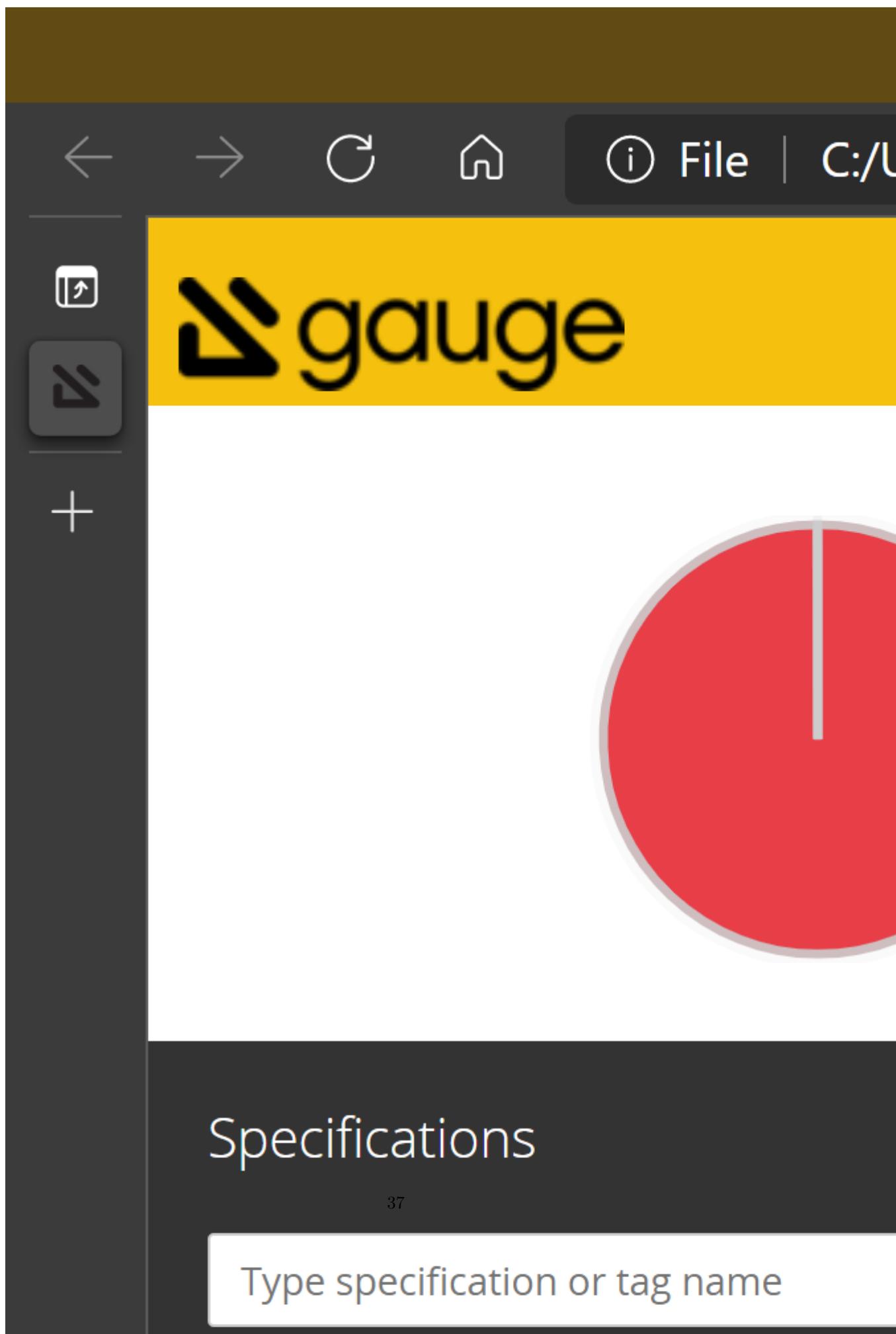
This allowed the developer to watch as Gauge implemented the Taiko automation steps. Gauge also provides a testing report in HTML format, a sample of which is provided in ??.

**Figure 21.** Successful Gauge Testing Report

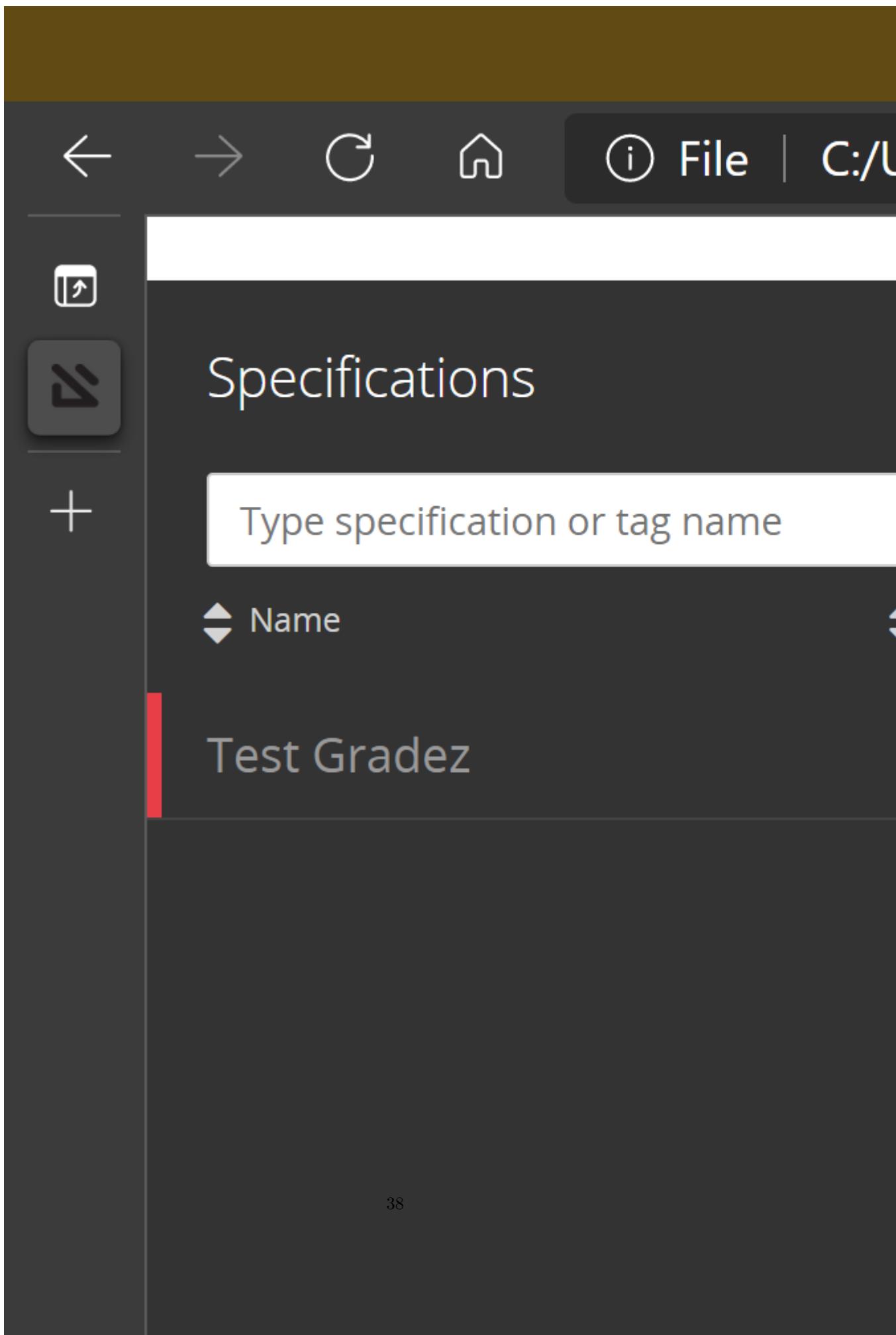


When the testing fails the report indicates this, as shown in ???. Additionally Gauge will take screenshots of the automated browser window to aid the developer in determining the cause of the failure, as shown in ??.

**Figure 22.** Failed Gauge Testing Report



**Figure 23.** Failed Gauge Testing Report - Detailed View



By using the automation testing software we were able to run acceptance tests on the application repeatedly throughout the development process. Additionally by standardizing the tests we removed some of the aspects of human error from the testing by better ensuring that the same tests were run each time. Knowing that the software would provide a quick log and a detailed report in the event of a failure gave us a better sense of confidence that whatever change we were implementing to the code was not causing the application to fail in a way that we were not focusing on at that time.

### 5.3.2 User Feedback

We tested the first iteration of the application with four real-world student users. Test subjects were asked to use the application with basic guidance and record observations, using the previously outlined UMUX-Lite approach. The key learnings from this initial user testing are summarised below, with a full list available in the appendix.

- Because pages look similar, some navigational feedback is necessary to indicate where users are. They felt lost.
- It wasn't clear what functionality is available to logged in users vs. public users.
- It wasn't clear that some grades are anonymous, these need to be marked more clearly and present a default image next to them
- The login procedure isn't very friendly to use, and overall the product looks dull and uninviting.

A number of smaller usability issues were also mentioned, which we considered as scope for the next iteration. We decided to capture all feedback items and prioritize them by impact and effort for further development.

## 5.4 Error Handling

We have implemented two kinds of error handling in this MVP version of the application. The first is on a technical level, where we have used `try-catch` blocks in the code, as shown in [??](#). This block ensures that a user cannot add a duplicate record to the grades database, and provides a useful error message back to the user. This approach is used throughout the application on all routes.

The UI prevents users from entering a duplicate grade by not offering modules that already have submitted grades. However, we have added an additional layer of error-handling and used the `try-catch` block to provide a useful error message - in case a user does somehow manage to submit a duplicate grade. Based on [\[?\]](#), we ensured that the button labels are action-oriented and the errors clearly indicate what happened, and what the user can do next.

---

**Listing 1** Using try-catch blocks for error handling

---

```
1 app.post("/addgrade", checkAuth, checkPermission, async (req, res) => {
2     try {
3         let params = [req.body.course_id, req.body.semester, req.user.id,
4             ↪ req.body.grade, !!req.body.anonymous];
5         let sql = `
6             INSERT INTO grades(course_id, study_session_id, user_id, grade,
7             ↪ anonymous)
8                 VALUES (?, ?, ?, ?, ?);
9         var [results, _] = await db.query(sql, params);
10        res.redirect(req.baseUrl + "?addResult=success");
11    } catch (error) {
12        console.log(error);
13        if (error.code === "ER_DUP_ENTRY") {
14            res.redirect(req.baseUrl + "?addResult=You already have a
15            ↪ grade for module ${req.body.course_id}. You may edit an
16            ↪ existing grade");
17        }
18    }
19 });
20 );
```

---

## 6 System Development

### 6.1 Technical Approach

### 6.2 Relevant Code Extracts

---

**Listing 2**

---

```
1
```

---

### 6.3 File Structure

The application uses a file structure as shown in ??, following a standard pattern for a Node application. The structure separates configuration files, database scripts, middleware (where all the more specific application logic lives), a public folder that contains all the assets need to be served across pages such as images and client-side styling and scripts. Further, the actual application routes are defined in their own folder, and the actual application pages reside in a folder for views.

---

**Listing 3**

---

---

**TREE**

---

## 6.4 Difficulty And Novelty Of Technical Challenge

We judged the technical difficulty of this application in terms of the skill levels in the team. We chose an approach that was level-appropriate for the varying backgrounds in the team, and included some additional challenges to learn new skills.

1. We introduced a more modern routes logic than we learnt in the program's "Databases, Networks and the Web" module, by using promise functions. This was new to most team members.
2. We used Taiko and Gauge for automated browser testing and behaviour-driven development, which was new to most team members as well.
3. We used Slack-based authentication which was an additional challenge to handle security aspects and integrate with a third-party authentication API.
4. Finally, we implemented a complex theming solution that required the team to learn how to incorporate the existing application with a new theming solution.

We chose to use well-documented technical approaches, while the main novelty of the solution lies in the open nature of the application for users, using Slack authentication instead of building a silo, and providing a demo mode for first-time users and graders.

## 6.5 Cycle 1 (Sprints 1-3)

Based on the previously defined epics, we captured and worked on first user stories during the first sprint of this cycle. The goal of this cycle was:

1. Set up the basic scaffolding for the application
2. Implement all basic functionality for displaying modules and capturing grades
3. Ensure use of real-world data in the application
4. Setting up behaviour-driven unit tests

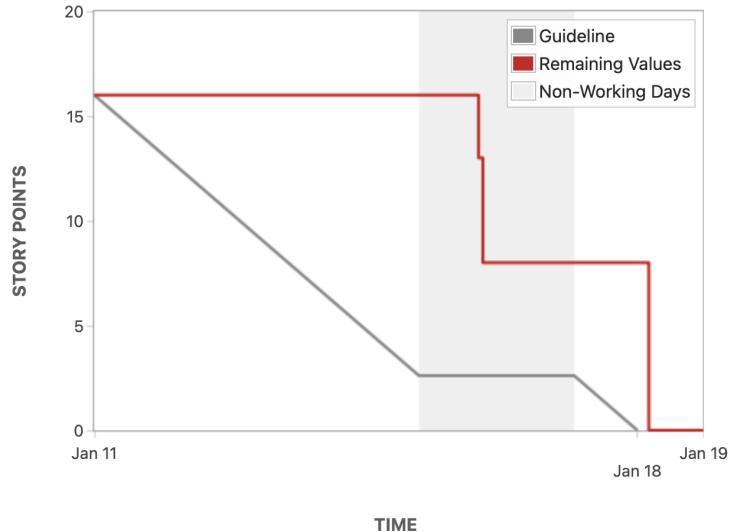
To achieve this first basic version, we decided to implement the previously defined wireframes using a standard Bootstrap theme, without too much focus on aesthetics and a higher focus on basic usability and functionality.

### 6.5.1 Sprints

We worked in 1-week sprints during most cycles and made sure to measure our sprint progress using the SCRUM methodology. We estimated captured user stories, planned them in and worked on the

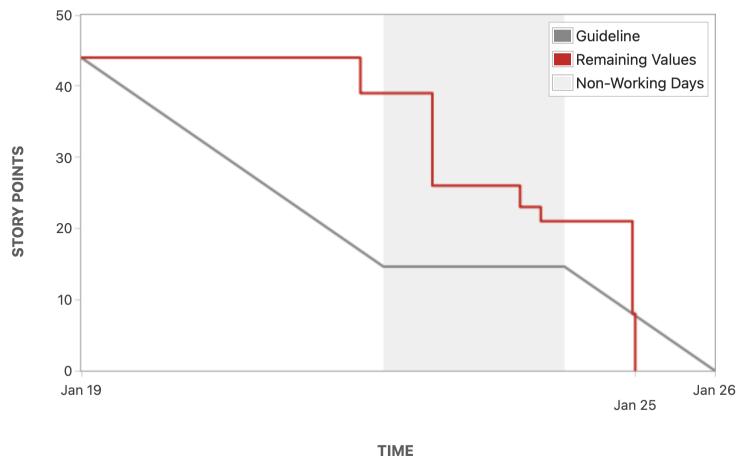
defined scope, tracking our progress using burndown charts. The first three sprints can be seen in ??, ?? and ??.

Closed sprint, ended by [Blair Currey](#) 11/Jan/21 7:34 PM - 19/Jan/21 3:51 PM  
[View linked pages](#)

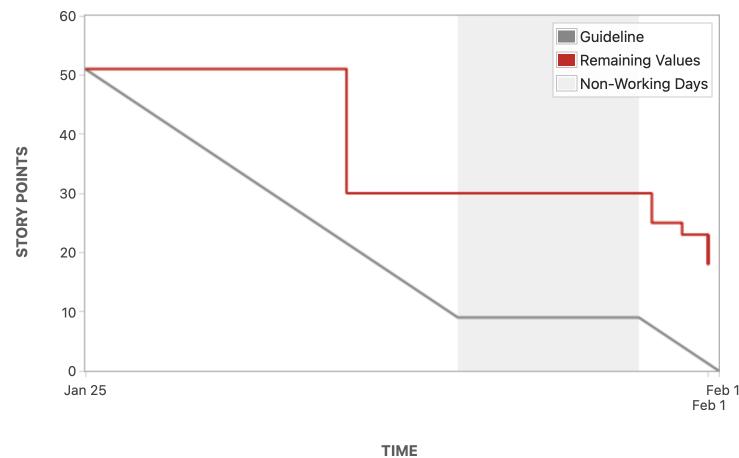


**Figure 24.** Burndown chart - Sprint 1

Closed sprint, ended by [Blair Currey](#) 19/Jan/21 3:52 PM - 25/Jan/21 6:48 PM  
[View linked pages](#)



**Figure 25.** Burndown chart - Sprint 2



**Figure 26.** Burndown chart - Sprint 3

## 6.6 Shipped Product Increment

At the end of this cycle, we had implemented a functioning version of the grades leaderboard, covering the related user stories (see Appendix A for a list).

The outcome is shown in ??, ??, ?? and ??.

A screenshot of a web-based application titled "Module Overview". The interface includes a top navigation bar with links for "Grades Leaderboard", "Module Overview", "My Grades", and a blue "Add grade" button. On the right, there's a user profile with the name "Hi, Arjun" and a "Logout" link. Below the navigation is a search bar labeled "Search". The main content area features a table with 14 rows, each representing a course. The columns are labeled "ID", "Course", "Average Grade", and "Number of submissions". The courses listed are: CM1005 (Introduction to Programming I), CM1010 (Introduction to Programming II), CM1015 (Numerical Mathematics), CM1020 (Discrete Mathematics), CM1025 (Fundamentals of Computer Science), CM1030 (How Computers Work), CM1035 (Algorithms and Data Structures I), CM1040 (Web Development), CM2005 (Object Oriented Programming), CM2030 (Graphics Programming), CM2035 (Algorithms and Data Structures II), CM2040 (Databases, Networks and the Web), and CM3070 (Final Project). The average grades range from 67 to 98, and the number of submissions ranges from 1 to 5.

ID	Course	Average Grade	Number of submissions
CM1005	<a href="#">Introduction to Programming I</a>	98	5
CM1010	<a href="#">Introduction to Programming II</a>	83	5
CM1015	<a href="#">Numerical Mathematics</a>	88	5
CM1020	<a href="#">Discrete Mathematics</a>	91	5
CM1025	<a href="#">Fundamentals of Computer Science</a>	89	5
CM1030	<a href="#">How Computers Work</a>	76	2
CM1035	<a href="#">Algorithms and Data Structures I</a>	93	5
CM1040	<a href="#">Web Development</a>	67	5
CM2005	<a href="#">Object Oriented Programming</a>	93	4
CM2030	<a href="#">Graphics Programming</a>	88	1
CM2035	<a href="#">Algorithms and Data Structures II</a>	88	3
CM2040	<a href="#">Databases, Networks and the Web</a>	93	4
CM3070	<a href="#">Final Project</a>	95	2

**Figure 27.** *Module Overview*

The screenshot shows a web application window titled "CM1005 Leaderboard". At the top, there is a navigation bar with links for "Grades Leaderboard", "Module Overview", "My Grades", and a blue "Add grade" button. To the right of the navigation is a user profile with the name "Hi, Arjun" and a "Logout" link. A search bar is located at the top right of the main content area.

**CM1005 Leaderboard**

Rank	Name	Grade
1	Alex	100
1	Brad	100
1	Blair Currey	100
4	Arjun	96
5	Hayato	93

**Figure 28.** Single Module

The screenshot shows a web application window titled "Arjun". The top navigation bar is identical to Figure 28, with "Grades Leaderboard", "Module Overview", "My Grades", and a blue "Add grade" button. The user profile shows "Hi, Arjun" and a "Logout" link. A search bar is at the top right.

**Arjun**

Session	ID	Level	Module	Grade	
April 2019	CM1005	4	Introduction to Programming I	96	<a href="#">edit</a>
April 2019	CM1015	4	Numerical Mathematics	88	<a href="#">edit</a>
October 2019	CM1010	4	Introduction to Programming II	81	<a href="#">edit</a>
October 2019	CM1020	4	Discrete Mathematics	92	<a href="#">edit</a>
October 2019	CM1025	4	Fundamentals of Computer Science	82	<a href="#">edit</a>
October 2019	CM1035	4	Algorithms and Data Structures I	86	<a href="#">edit</a>
April 2020	CM1040	4	Web Development	68	<a href="#">edit</a>
April 2020	CM2005	5	Object Oriented Programming	98	<a href="#">edit</a>
April 2020	CM2040	5	Databases, Networks and the Web	93	<a href="#">edit</a>
			Cumulative Grade	90	
			Completion Rate	38%	

**Figure 29.** Personal Grades

A screenshot of a web browser window titled "localhost". The window has a header with links: "Grades Leaderboard", "Module Overview", "My Grades", "Add grade" (which is highlighted in blue), "Hi, Arjun", and "Logout". Below the header is the title "Add grade". There are three dropdown menus: "Module" (labeled "Choose a module..."), "Session" (labeled "Choose a session..."), and "Grade" (an input field). Below these is a checkbox labeled "Do not reveal my identity in leaderboard. Keep this grade anonymous." followed by a "Submit" button.

**Figure 30.** Adding Grades

## 6.7 Cycle 2 (Sprints 4-6)

## 6.8 Cycle 3 (Sprints 7 - 9)

# 7 Analysis And Evaluation

In order to evaluate our progress and success, we decided to select specific success factors that would indicate if the product is successful following a real-world market launch. We therefore set ourselves targets for each of these factors for both a "P0" launch, which is what we intend to build as part of this course module, and long-term goals, which are longer-term goals if this product were launched in real-world conditions.

## 7.1 Critical Success Factors

Through a further brainstorming exercise, combined with some research, we identified the following success factors.

- **Reach:** The number of distinct students from the overall student population that has submitted grades. The population is defined as the number of students we can observe in the `#general` channel of the student Slack.
- **Coverage:** The number of modules that have a sufficient number of grades submitted for mean-

ingful comparison. We define "meaningful" as a minimum of 5 grade submissions for a single module.

- **Repeat:** The number of students who submitted at least two grades in two distinct terms.
- **Adoption:** The number of academic institutions and their respective student bodies that have adopted this product.
- **Rating:** The average of all user-ratings in app stores where the product is available.
- **Usability:** Usability score measured with UMUX-Lite method
- **Usefulness:** Usefulness score measured with UMUX-Lite method

## 7.2 Measurement Of Success & Failure

As noted, we set ourselves targets for the initial launch as well as longer-term goals.

	P0 Target	Long Term Target
<b>Reach</b>	2%	10%
<b>Coverage</b>	20%	100%
<b>Repeat</b>	n/a	20%
<b>Adoption</b>	1	20
<b>Rating</b>	n/a	4.5
<b>Usability</b>	40%	60%
<b>Usefulness</b>	40%	60%

## 7.3 UI Evaluation

## 7.4 Accessibility Audit

## 7.5 User Safety & Security

# 8 Individual Reflection

# Appendices

## A User Guide

## B User Stories

## C User Feedback

## D Git History

The following output shows a log of work done on this assignment as it was committed to our private Git repository. This is generated by `git log --color --graph --oneline`.

```
* 100bc1c (HEAD -> main, origin/main, origin/HEAD) Final version midterm
* 35c8b91 Added compelled pdf after last revision
* c8aae6f Added citations for section 1
* 1f9f666 Added citations and usability evaluation
* f296c91 Updated with compiled PDF
* fddbd84 Added more detail to initial epics - final candidate 2
* 0d49a0a Midterm final candidate
* 60b1a21 added some wireframes to section 5
* 94e4e7f Updated midterm section 6
* 48b7b4a updated wireframes
* feeeeaa5 Update leaderboard_wireframes.bmp
* 8f28ebb added the rest of 12-21 notes
* 0d09fae added my meeting notes
* 354fc35 new revision from group session
* 0cf3b52 updated overleaf as images were not visible or outdated
* 62ca2e7 new wireframes, renamed source file
* 5995e79 Added framing to SWOT/STEEPLE and added author names.
* 1a29062 Reviewed and restructured midterm LaTeX file. Edited sections 2-5 and
→ created tables. Incorporated storyboard graphics.
* f4465f9 Create casual_checker_1.png
* de57acb Create competitive_topper_1.png
* b32c653 export of gantt used in proposal
* 41852f6 Some rough ideas for reports to the user
* b7dea47 Modified wireframe
* 9f86020 Added account log in and sign in pages
```

```
* 54bf1e4 initial attempt at wireframe prototype
* 1f79184 Amended minutes
* 39db9d6 Minutes from 2020-12-10 session
* 04fa4e1 Notes 2020-12-07
* 1dc2d5b added updated midterm report pdf
* 5f165af Merge branch 'main' of
→ https://github.com/BlairCurrey/uol-agile-group-project into main
| \
| * f35e4a3 Added a couple of comments on the meeting
| * d70fb30 Fix date
| * c2d0c6a Merge branch 'main' of
→ https://github.com/BlairCurrey/uol-agile-group-project into main
| | \
| | * 273900f Updated notes 3-Dec-2020
| | * 0f60790 Uploaded summary of the book
| * | 03b91ca Capture ideation board after the meeting
| | /
| * abaac74 Update TG1_G6_GroupMeetings.md
| * a2e19a4 Updating exports of Miro
* | 51d1a3a Add STEEPLE & changed market analysis sectioning
| /
* 6424a17 added methodology for gathering competitors
* 1022228 images for market research
* e358d8f added Market Research
* 03e7384 Setup agenda for next meeting
* 0e21f30 Updated notes 27-11-2020
* effc3bd Added Tex Template for Midterm
* 23a65f7 Updated meeting outcomes November 23
* 29ab345 Update TG1_G6_GroupMeetings.md
* 347db17 2020-11-19 check-in
* 4f45b8a added 2020-11-12 meeting
* 7556267 corrected date for second meeting
* 859b63f Documentation for Weekly Working Session 2020-11-09
* 9b7188d Added Meeting Notes 2020-11-05 and updated README
* f461ecf Initial commit
```