Curtin University

6/1/2016

Final Report

Project Design and Management 300

HOOLI XYZ

Connor Beardsmore – 15504319
Brendan Lally – 18407220
Adam Brown – 18338993
Jordan Chou – 18348691
Edward Lim - 18339145

Science and Engineering
CURTIN UNIVERSITY

Abstract

This report has been generated to cover all aspects of the development of an online web platform for Curtin associates to collaborate on ideas and innovations. The report has been developed for the project client to address the key aspects of the project implementation, design and other project management issues. Outlined in this report is the project background, design and implementation decisions, full documentation for each Sprint iteration and a post-mortem discussion on the challenges faced.

The online platform was developed utilizing the Django framework, a high-level python web framework. Issue tracking was performed via the Pivotal Tracker platform, with version control also being applied as required. The back-end database of the platform was implemented via the SQLite database management system.

The project utilized Agile techniques and the SCRUM framework to ensure its success. SCRUM uses an iterative and incremental delivery solution to help deliver projects on track with all functionality implemented. The use of SCRUM ensured that the project was a success and that the team functioned efficiently and productively. The resulting solution delivered provides all the functionality requested by the project client in the requirements document. In addition to the functionality, the produced platform is fully polished with an easy to use and aesthetic user interface.

Table of Contents

Table of Figures

1.0 Introduction and Objectives

The main purpose of this report is to illustrate the process employed in developing our project solution for the Curtin Ideas platform. The report provides a detailed overview of the projects design and exists to summarize the teams progress over the previous four months. The primary audience of this report is the project client, in addition to anybody interested in understanding the SCRUM framework and its implementation on an academic group project.

The objective of the project was to deliver a working solution for the Curtin Ideas online platform. That is, a solution that has completely implemented all functionality defined by the project requirements. A secondary objective was for all team members to gain experience with both web development and the SCRUM Agile development technique.

The project was heavily constrained by stringent time requirements and cost, as there was no budget at all allocate for the platform. The scope of the project relates to the development of the platform, with anything outside of the solution development considered out-of-scope.

The platform developed utilized the SCRUM framework and an Agile project management technique. This incremental ideology of the SCRUM framework saw the project delivered in 4 Sprints of two to three weeks long. In each sprint, an overall plan of the goals for that sprint was developed, weekly SCRUM meetings were held and a sprint retrospective was written and discussed to enhance the future sprint progress.

This report outlines the entire process of developing the Curtin Ideas platform from planning stages to the final delivery. The project background is discussed, examining the project management approach taken and assessing the tools utilized for the project. The entire product backlog is listed, with each sprints backlog attached in the sprint reports within Appendix A. Design and implementation details of the system are heavily discussed, with every aspect of the platform being justified. Finally, an honest retrospective on the entire process is conversed, with the entire project being summarized.

2.0 Background

2.1 Project Management Approach

The project management approach utilized throughout the project was the SCRUM framework, which is based around an iterative Agile technique. One project member was chosen as the SCRUM master to ensure the project stayed on track and to shield members from external interferences. The SCRUM framework was followed, with the team holding planning and retrospective meetings for every sprint in addition to the weekly SCRUM meetings held (Twalib, 2016).

The foundation of this approach was centered around communication. Communication is vital to the success of any project (Schalwbe, 2006) and thus, we aimed to guarantee that there was never a lack of communication at any stage throughout the process of developing the final product. This communication was achieved via the prior mentioned SCRUM meetings.

Early planning of the solution was key to the projects overall success, as it provided a vision for the final product. The original plan was discussed at great length by the team and is discussed both in section 2.4's project mockup and the extensive design details in section 4.1. Planning was performed in a whiteboard session with numerous possible solutions considered.

2.2 Software Solutions Considered

The solution required is similar to other, similar solutions currently available. Thus, while planning our solution, we researched these solutions to determine which elements of these solutions we valued. The two similar systems the team analyzed during planning were the Reddit website and the HackerNews website.

Both Reddit and HackerNews are solutions extremely similar to ours, with the ability to post submissions and a solid voting system. The simple header and footer on HackerNews makes navigation throughout the site extremely easy and we aspired to make our solution comparably easy to navigate. The primary aspect of the Reddit site we liked was the voting system. The voting buttons are organized aesthetically and are simple for users to use. We desired our solution to have similar navigation and voting systems to these well-defined solutions and this early analysis helped guide our planning process.

2.3 Tools Utilized

The tools considered for use throughout the project were centered around the following categories:

- Communication
- Task management
- Development
- Documentation

The tools considered for communication included *Facebook Messenger* and *Slack* for group messaging in addition to *Hangouts* and *Skype* for video conferencing. Due to some group members being unfamiliar with *Slack* we decided to go with the more familiar messaging platform and hence selected *Facebook Messenge*r. As for the video conferencing solution, we discovered that Skype required premium access for group video calls. As a result of this, we found that Skype did not fit our usage requirements. On the other hand, *Google Hangouts* is a free service provided by Google and is easily accessible. Both of the choices we made regarding communication tools are cross-platform and worked well with our group as members used a variety of devices from Android to iOS.

Utilizing Facebook Messenger allowed us to stay in contact with each other throughout the project duration. The push-notifications allowed members to quickly respond to questions which increased our productivity and helped reduce stress as we were able to stay up to date during the development process. The usage of Google Hangouts also increased productivity greatly. Our primary use for Google Hangouts was as an informal conferencing tool. We found that, due to the tools we have chosen and the way we allocated tasks, some members did not know how to do some things that others knew how to. When such a situation occurred we found that using Hangouts as a way to explain step-by-step tasks, worked well. The ability to share-screen in Hangouts made it an excellent learning tool and allowed more knowledgeable group members to guide and teach others.

The tools considered for task management were chosen due to their suitability for use with the SCRUM methodology. Software solutions considered included *Atlassian's JIRA, Pivotal Lab's Pivotal Tracker* and *Trello*. The group began to use JIRA but discovered that the student trial was not sufficient for the project. As the project owner had given us access to *Pivotal Lab's Pivotal Tracker* (Pivotal Tracker, 2016) we decided to use that platform as an alternative. However, the group found *Pivotal Tracker* a little

cumbersome to use over time and we decided to additionally use *Trello* to track the tasks.

The decision on which tools to employ for the actual development of the platform was crucial to the success of the project as a whole. The tools we considered included *PHP, C# ASP.NET, Django,* and *Node.js*. Database management solutions that we considered *included MongoDB, MySQL, phpMyAdmin and sqlite3*. The majority of team members had no prior experience in web development so regardless of the chosen platform, there was a large learning curve to overcome. After researching online, we opted for *sqlite3*, the simplest of the database tools. In order to connect to our database, perform business login and serve the frontend we decided to develop the platform using the *Django* framework, due to its relatively simple learning curve. Several group members desired to learn *Python*, with the *Django* framework also providing this possibility. Both the *Django* framework and *sqlite3* systems are open source, ensuring that no additional cost is passed onto the project client.

2.4 Project Mockup

The majority of the early project planning was performed during a whiteboard session, with numerous solutions covered and discussed. The majority of these ideas were discarded early on and thus are of little relevance to this report. The following project mockup shown in Figure 1 however, was our final project mockup. This mockup was completed in the early stages and guided our design and styling for the entire project. Figure 2 displays a more advanced mockup that was produced slightly later in the planning process.

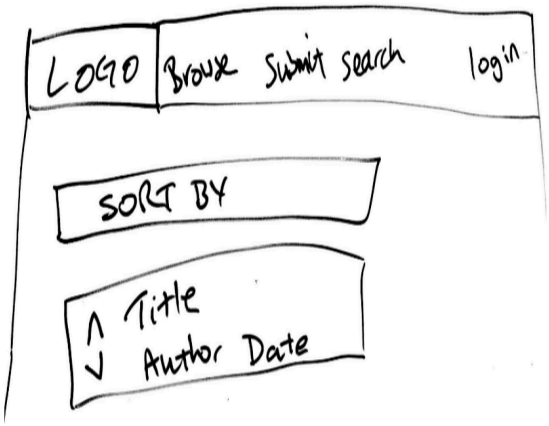


Figure 1: Project Mockup

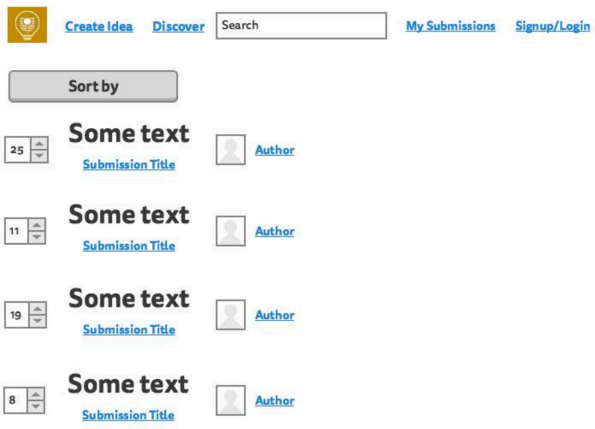


Figure 2: Advanced Project Mockup

3.0 Product Backlog

Upon receiving the project requirements from the client, the requirements were converted into a collection of 45 user stories. These stories were further broken down into groupings for simplicity and to improve modularity when the stories were implemented. The following major groupings were applied to the user stories:

- Account
- Administration
- Commenting
- Posting Submissions
- Viewing Submissions
- Voting
- Rewards

Each user story was allocated into a specific grouping, with each sprint generally focusing on completing one or two groups of stories. The group allocations for all user stories are displayed below. For a more detailed overview of the product backlog and a breakdown of stories completed per sprint, see the attached file exported from *Pivotal Tracker* in Appendix B.

3.1 Account

- As a User, I want to login to the platform so that I can post a submission.
- As a User, I want to logout of my account so that nobody else can access my account.
- As a User, I want to create an account so that I can use the platform.
- As a User, I want to be able to change my password so that I can keep my account secure.
- As a User, I want to access user profile so that I can view specific user's details.

3.2 Administration

- As an Admin, I want access to a Admin platform so that I can perform administrative actions.

- As an Admin, I want to create user profiles so that users can use my site effectively.
- As an Admin, I want to edit submissions so that I can remove offensive language.
- As an Admin, I want to remove submissions so that I can remove inappropriate submissions
- As an Admin, I want an Admin tag so that other Users see my account as an Admin.

3.3 Commenting

- As a User, I want to delete my improvements so that I can remove my improvement.
- As a User, I want to edit my improvements so that I can fix spelling errors.
- As a User, I want to delete my comments so that I can remove my comment.
- As a User, I want to edit my comments so that I can fix spelling errors.
- As a User, I want to suggest improvements to a submission so that I can improve upon an idea.
- As a User, I want to comment on a submission so that I can voice my support of the idea.
- As an Admin, I want to be able to delete comments because they may contain inappropriate material.
- As an Admin, I want to edit comments so that I can remove offensive language.
- As an Admin, I want to delete improvements so that I can remove inappropriate improvements.

3.4 Posting Submissions

- As a User, I want to post a submission so that I can put my idea onto the platform.
- As a User, I want to delete my submission so that I can remove my submission.
- As a User, I want to edit my submission so that I can fix spelling errors.
- As a User, I want to add images to my submission so that I can add relevant images to my submission.
- As a User, I want to add videos to my submission so that I can add relevant videos to my submission.

3.5 Viewing Submissions

- As a User, I want to view existing submissions so that I can see all submissions posted.
- As a User, I want to search all submissions so that I can find one relevant to my field of study.
- As a User, I want to sort submissions based on popularity so that I can see the most popular submissions.
- As a User, I want to sort submissions based on date so that I can see the newest submissions.
- As a User, I want to sort submissions based on views so that I can see the most popular submissions.
- As a User, I want to categorize my submissions so that other users can find it easily.
- As a User, I want to view submissions based on category so that I can view relevant submissions.

3.6 Voting

- As a User, I want to down-vote on submissions so that I can acknowledge submissions I dislike.
- As a User, I want to up-vote on submissions so that I can support submissions I like.
- As a User, I want to see the total votes for a submission so that I can see how popular it is.
- As a User, I want to up-vote on improvements so that I can support improvements I like.
- As a User, I want to down-vote on improvements so that I can acknowledge improvements I dislike.
- As a User, I want to only vote once for a submission so that voting is fair and equal for all users.
- As a User, I want to un-vote on submissions so that I can change my mind on voting.

3.7 Rewards

- As a User, I want other users to see my level when I post so that I can show off my level.
- As a User, I want to gain levels based on my points so that I am rewarded for utilizing the platform.
- As a User, I want to gain points when my submission receives votes so that I am rewarded for good work.
- As a User, I want to gain points for posting improvements so that I am encouraged to post again.
- As a User, I want to gain points for posting submissions so that I am encouraged to post again.
- As a User, I want to gain points for voting on submissions so that I am encourages to vote again.

3.8 Pivotal Tracker Analytics

The following figures illustrate the progress through the product backlog over the period of the project. Figure 3 shows the cumulative flow of points, while Figure 4 and Figure 5 display the project burnup and burndown respectively.

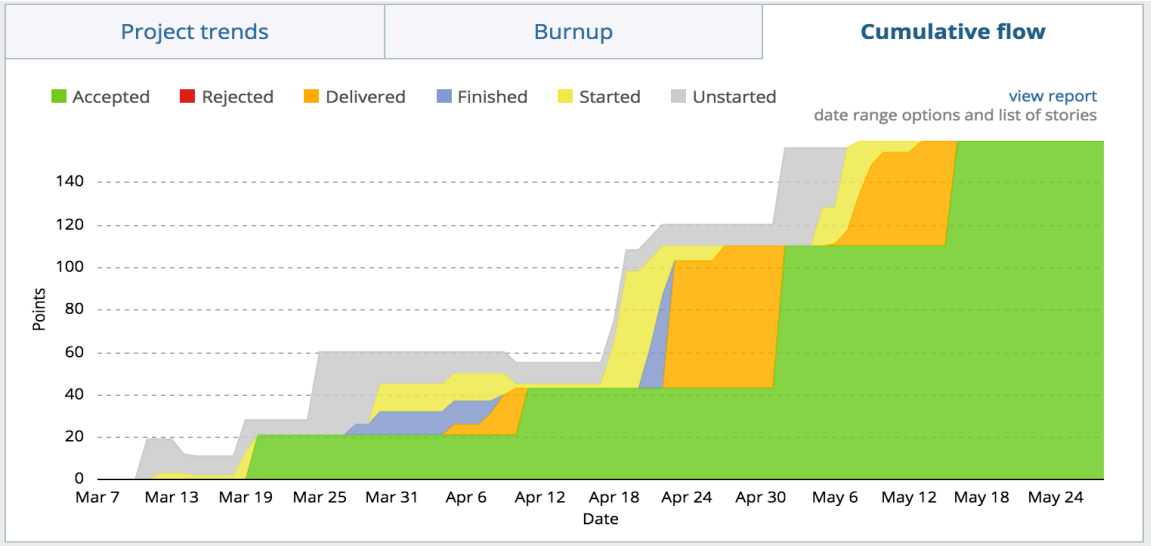


Figure 3: Cumulative Flow

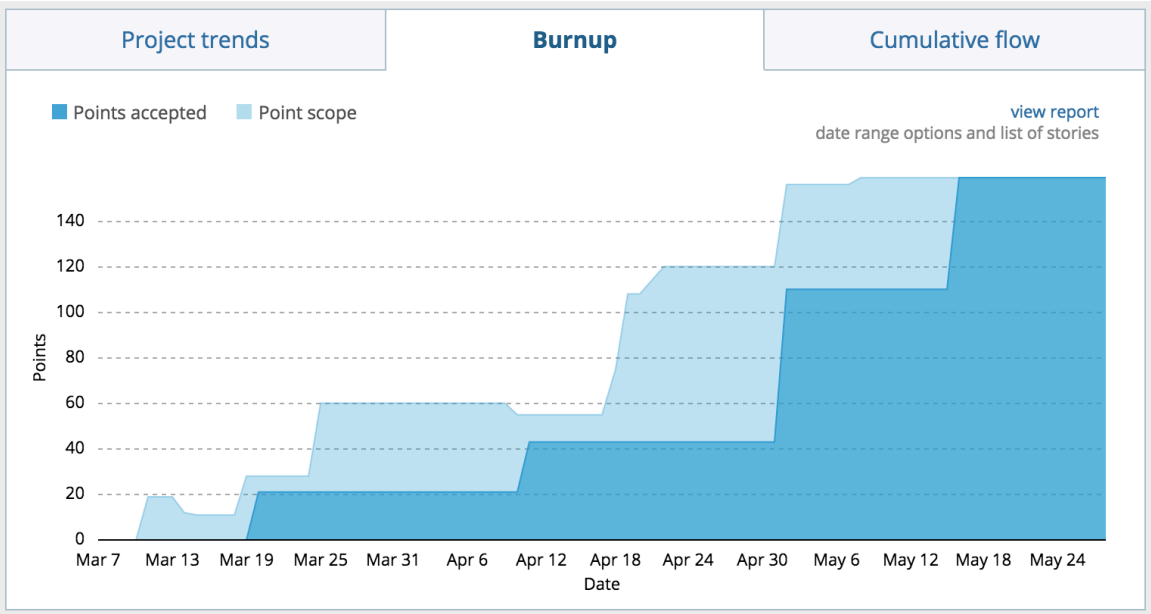


Figure 4: Project Burnup

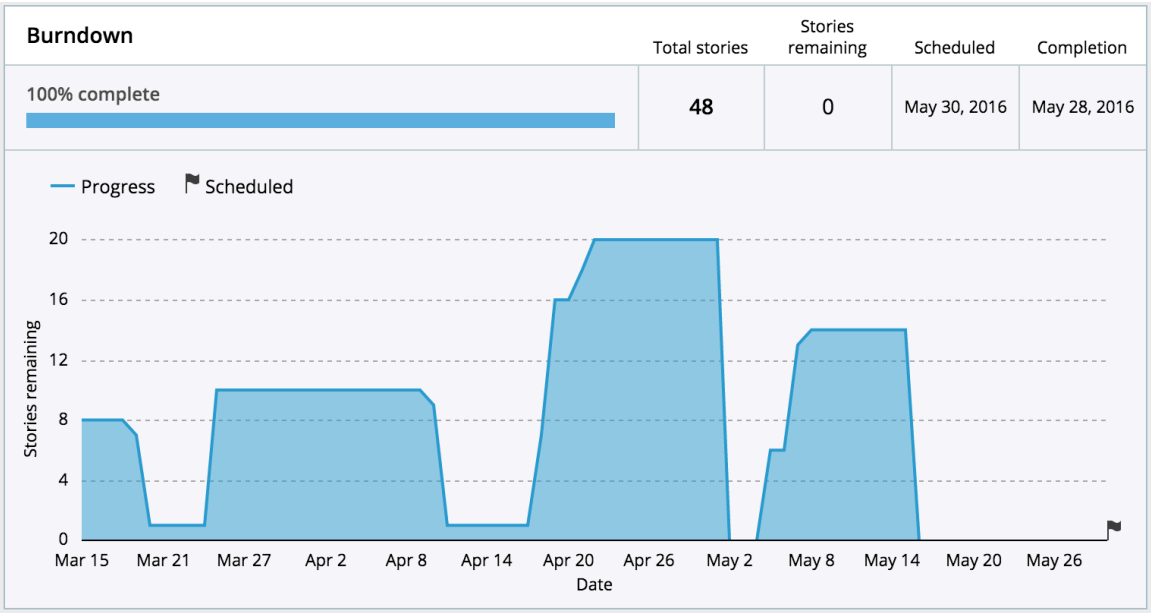


Figure 5: Project Burndown

4.0 Overall Design

This section of the report illustrates the overall design details and the implementation of the project solution. The design details consider both the back-end and front-end user interface while the implementation details cover all of the user story groupings discussed in the product backlog.

4.1 Design Details

As mentioned within the report background, it was decided that the platform would be implemented using the Django web framework. This would allow for easy-to-use and powerful admin features, along with well-defined methods to implement all the functionality deemed necessary. Django's model system seemed very applicable for creating objects such as users, submissions and comments which were central to the overall design. An overview of all the models utilized in the back-end system is illustrated in Figure 6.

4.1.1 Back-end

One of the major design issues that was central to the project as a whole was the idea of the submission. The submission model needed to store information about a user-generated idea including the title, author, category, the idea itself and links to external webpages. Every feature of the platform in some way interacts with the submission, be that viewing submissions or suggesting improvements for them. Additionally, submissions had to be both editable and removable by the correctly privileged users/admin.

We wanted a user to be able to create an account, storing basic personal information that could be used to identify the user. We also desired a user to be able to participate in posting submissions and both commenting and voting on them. The choice was made that a user would have to be currently logged onto the system to be able to submit, comment and vote on any submissions. However, submissions could be viewed by anyone of the general public without requiring a registered account, as requested by the project client.

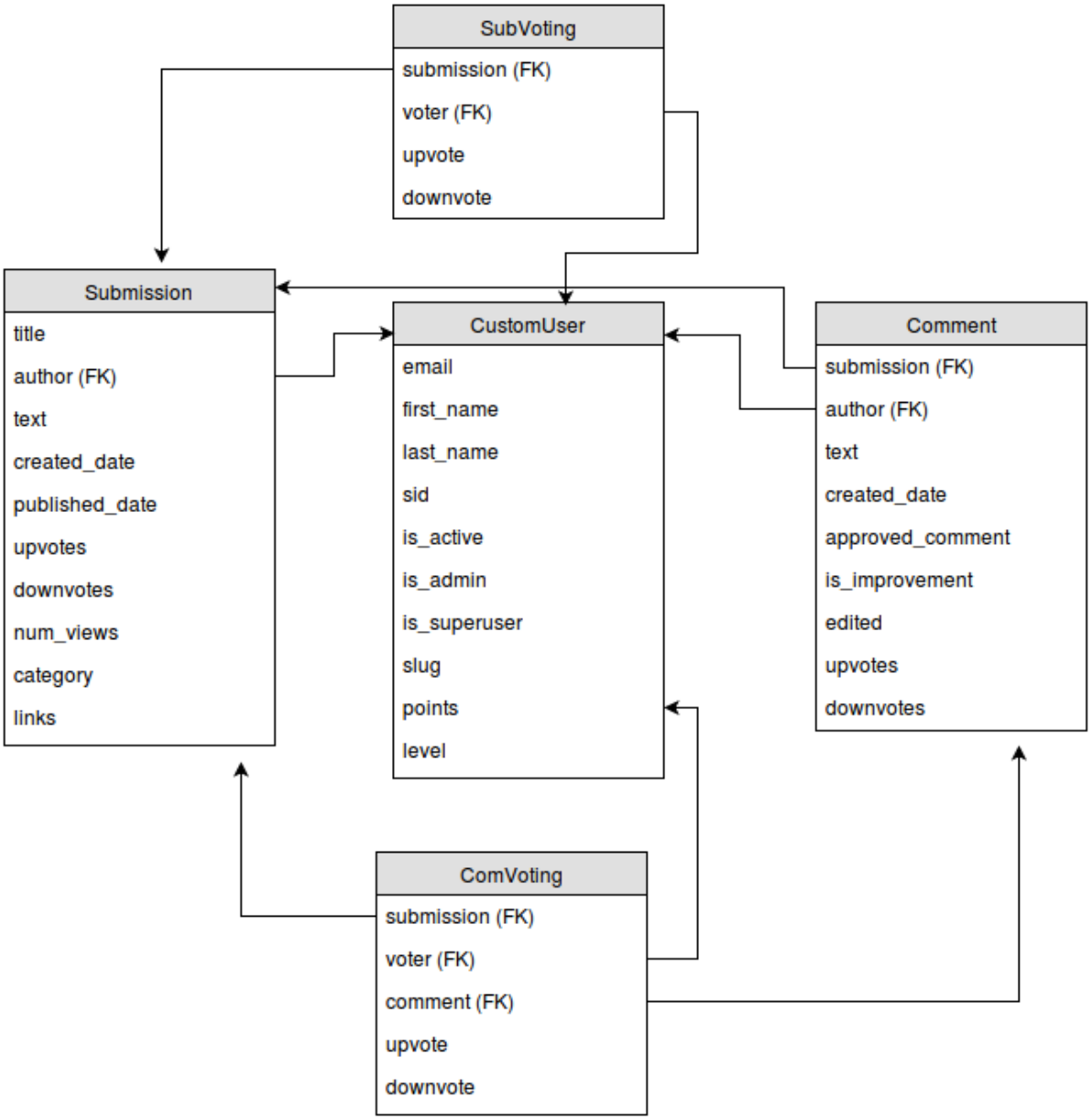


Figure 6: Django Models Overview

Comments and improvements were added for users to be able to give and receive feedback on ideas that were submitted. This feature fits with the whole collaborative ideology of the platform. Improvements enhanced the idea of comments, allowing users to provide constructive criticism on how to improve a submitted idea.

4.1.2 User Interface

The user interface design went through numerous iterations before eventually settling on the final design. The main aims of the interface design were to create a sleek, modern and polished interface. This interface was required to be both easy to use and easy to navigate to ensure users enjoyed utilizing the platform.

For the first sprint we began using the Bootstrap theme darkly CSS (Park, 2016) as a base outline for the styling, color scheme and UI elements of the platform. This was later modified, but it gave us a general idea of what the website would look like and allowed us to begin creating the templates for the webpages. We started with a simple navigation bar in the header of all of the pages to allow users to easily navigate the website.

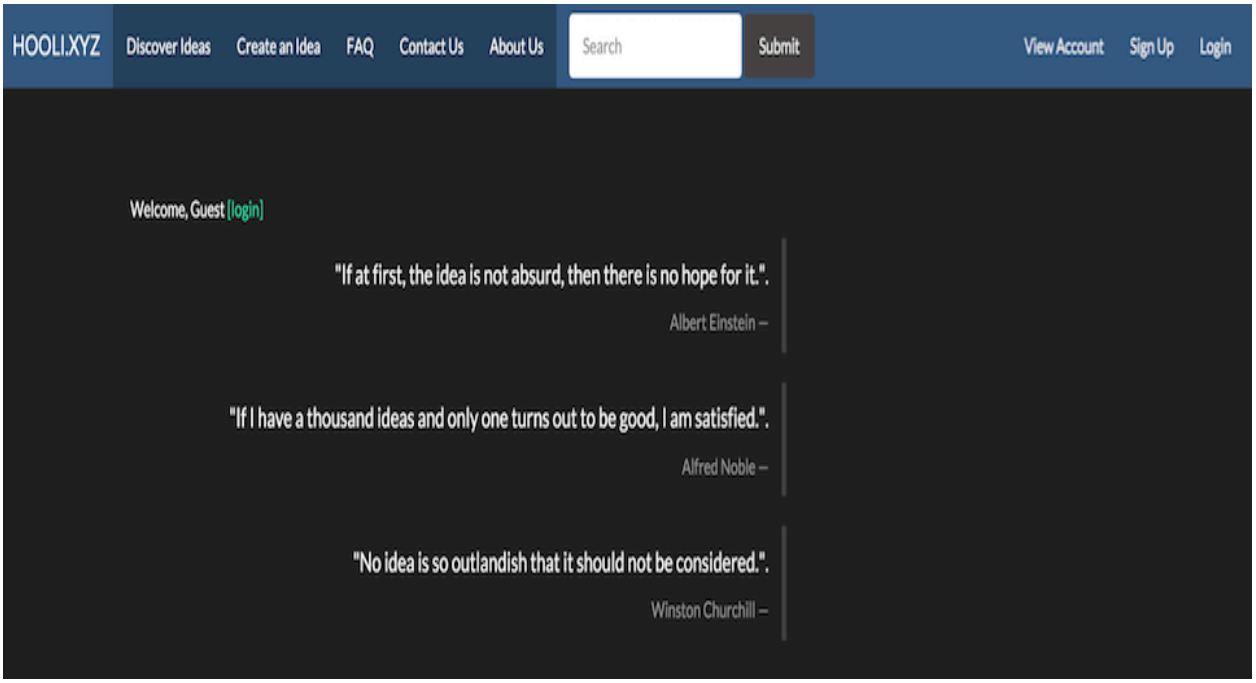


Figure 7: Sprint 1 UI

In the second sprint we added a footer to all of the pages to further increase the accessibility of the website. This ensured that users no longer had to scroll back to the top of pages to navigate. The navigation bar text was also simplified to reduce the amount of text that users needed to read when navigating, thus increasing the user-friendliness of the prototype.

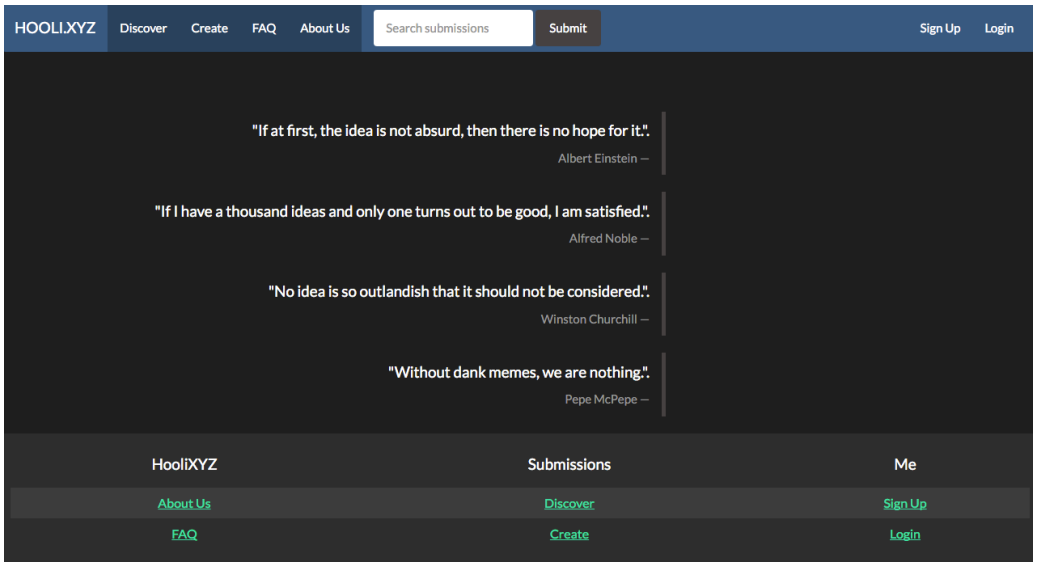


Figure 8: Sprint 2 UI

During Sprint 3 we altered the color scheme of the website so that it better resembled the styling of the official Curtin website. This also included adding a logo that incorporated the Curtin logo and changing the webpage name from HooliXYZ to CurtinIdeas. Various other UI elements were added to the Discover Ideas page, including simple buttons to sort the submissions.

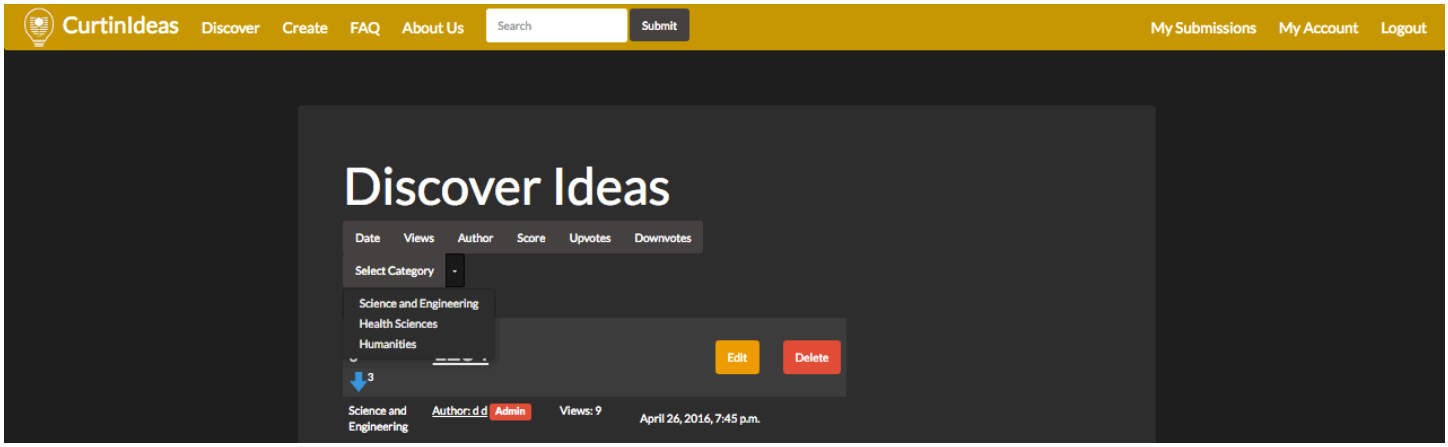


Figure 9: Sprint 3 UI

For the fourth sprint, we created our own local CSS file based on the one we were using before. This allowed much greater control over the user interface. For example, we could now actively modify the styling of the table elements used to list submissions. A background image was also added to the website to further relate the platform back to Curtin and certain elements were made opaque to allow the background to be seen.

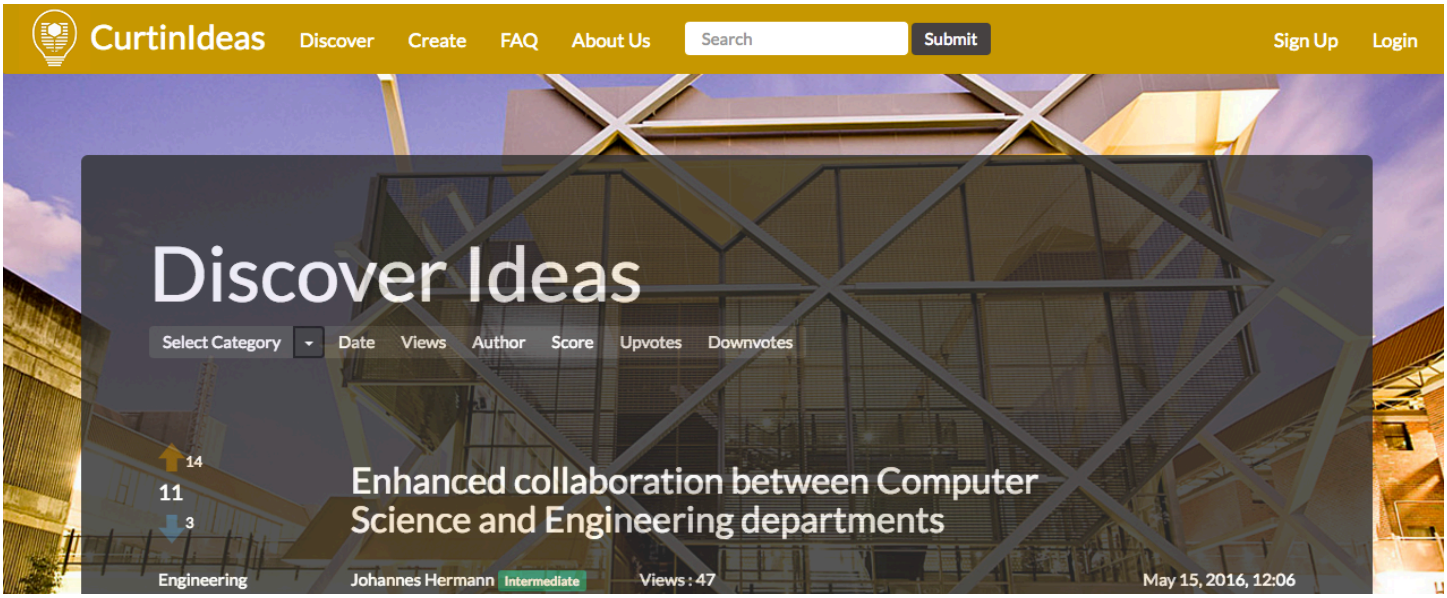


Figure 10: Sprint 4 UI

Figure 10 displays the final home page of the website after all user interface elements have been designed. This image incorporates the background image that was added during Sprint 4.

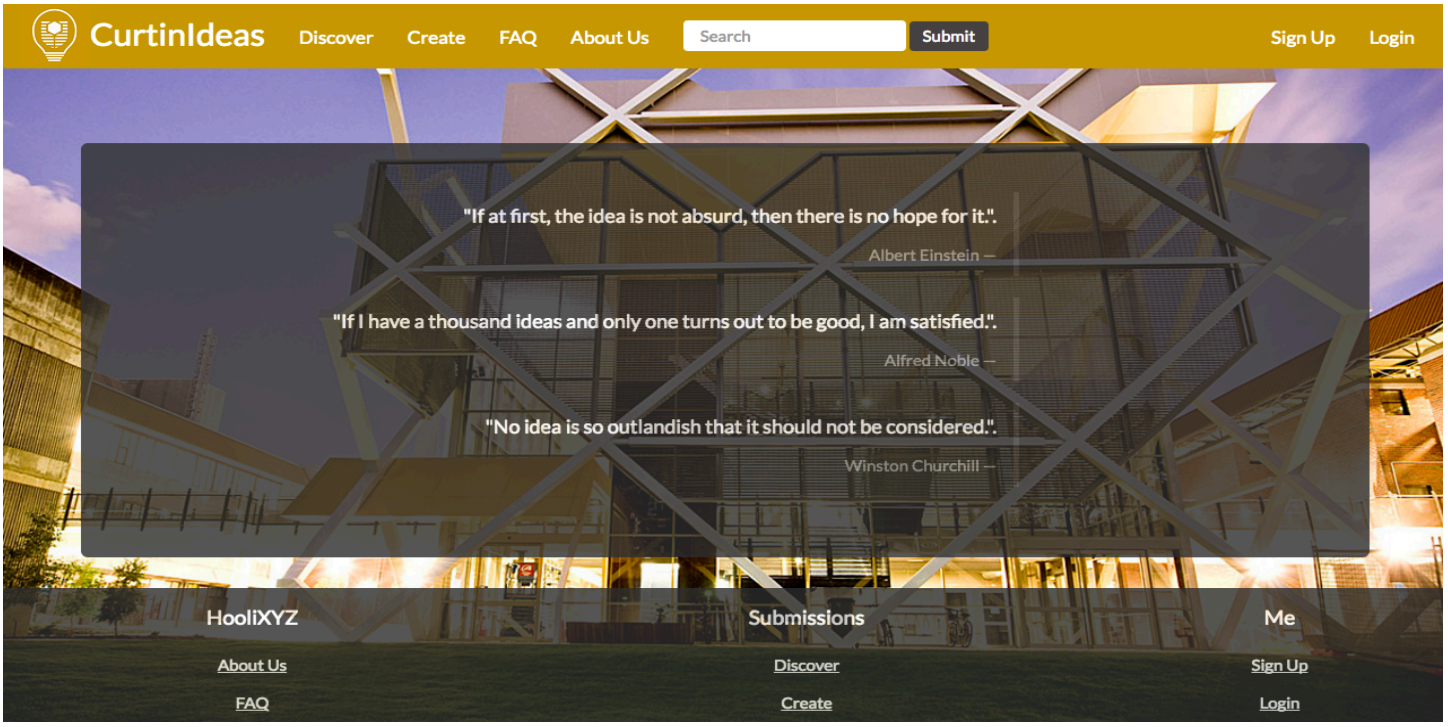


Figure 11: Final UI Design

4.2 Implementation Details

As discussed previously, all user stories were categorized into different groupings. The task of implementing the stories within these groupings were distributed to different group members, typically in pairs. A Fibonacci points system was used to estimate the effort required to complete the tasks. An effort of 1 or 2 typically meant the task was fairly straightforward to implement. An effort of 3 meant that more time was required and some research had to be undertaken to successfully implement the task. The 5 effort rating was the maximum effort estimation used in the project, implying that the feature required more thought and often multiple iterations to successfully implement.
Over the life of the project, there were minimal changes to the account creation form and overall functionality, with only a few user interface improvements being added later on. All of the planned functionality from the user stories was successfully implemented by the date of final release. Each of the user story groupings had specific implementation details.

4.1.2 User Accounts

The initial plan was to implement the administration and basic user account functionality to form the underlying framework of the project. The tasks required to set up user accounts included: creating a user model which in Django is used to represent a database object with object fields, creating a login form for users to enter the required personal information and adding logout and password changing capabilities. These tasks were given effort ratings of 2 and 3. An account creation form was made to allow the user to input the required information such as name, email, student/staff id and password (Figure 12). All the remaining information required by the system is automatically added to the created user object which is then stored in the back-end database.

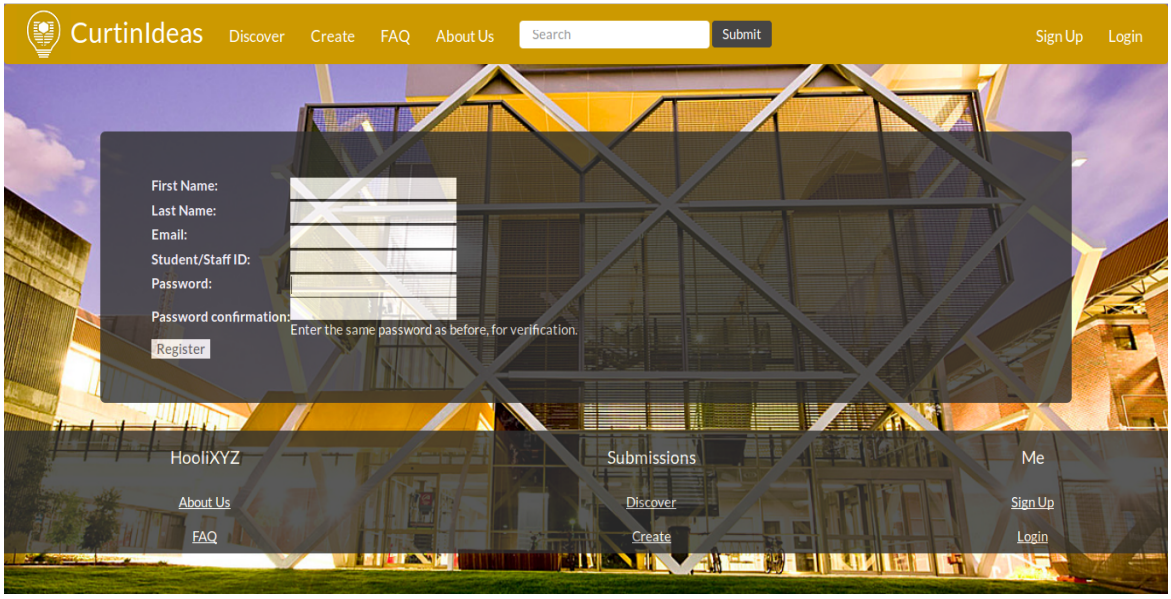


Figure 12: User Creation Form

Another feature relating to user accounts was the ability to display each user's profile information in a user profile (Figure 13). The task breakdown for adding the user profile included: creating a template to display the relevant user information, adding a progress bar to track points and linking these attributes to the database to allow them to be dynamically updated. The overall effort estimation for the profile implementation was 5.

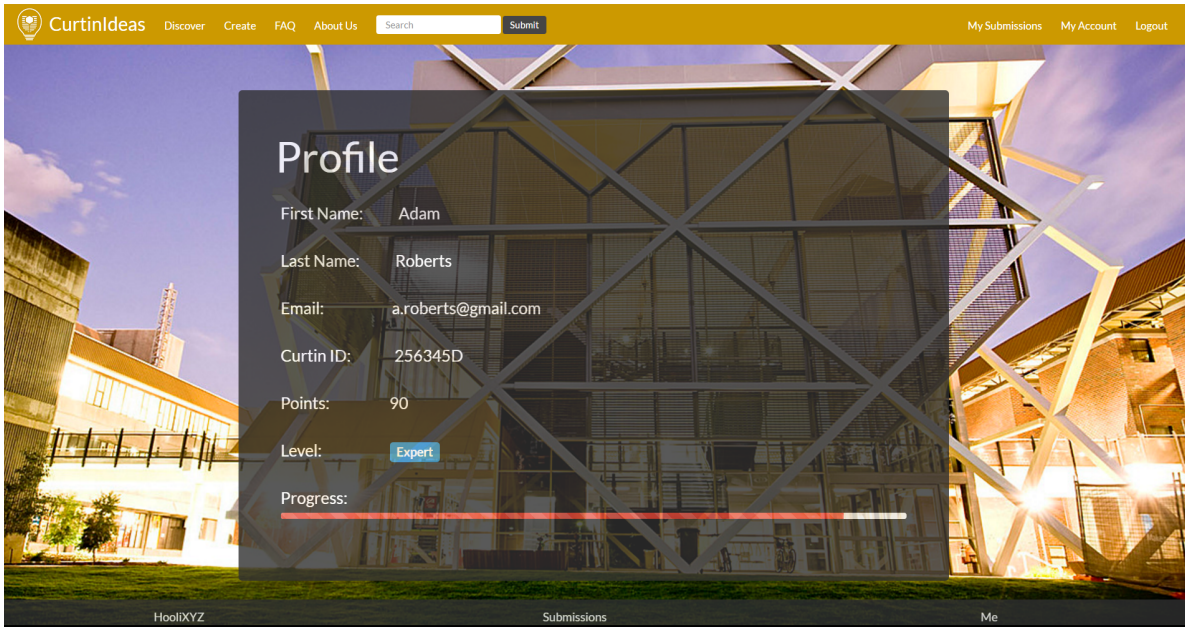


Figure 13: User Profile Display

4.1.2 Administration

The default Django administration platform was used to allow administrators to perform tasks such as creating a user, editing and removing submissions. This platform is in-built into the Django framework and little work was required to integrate this functionality seamlessly into our platform.

4.1.3 Submissions

Implementing the submission was arguably the most imperative and time consuming portion of the project. The tasks for implementing the submission functionality were once again broken down so there were separate pairs working on certain tasks. These tasks included: creating a submission model, creating a submission form, updating the database with submission objects, adding edit and remove buttons, creating a page to list submissions and adding a sorting function. The effort required to create the submission model and the submission form was listed at 3. Likewise, deletion of submissions was quite easily implemented, with a rating of 2. The task of implementing the editing features to work appropriately received an effort rating of 5 and was initially the most challenging feature to implement.

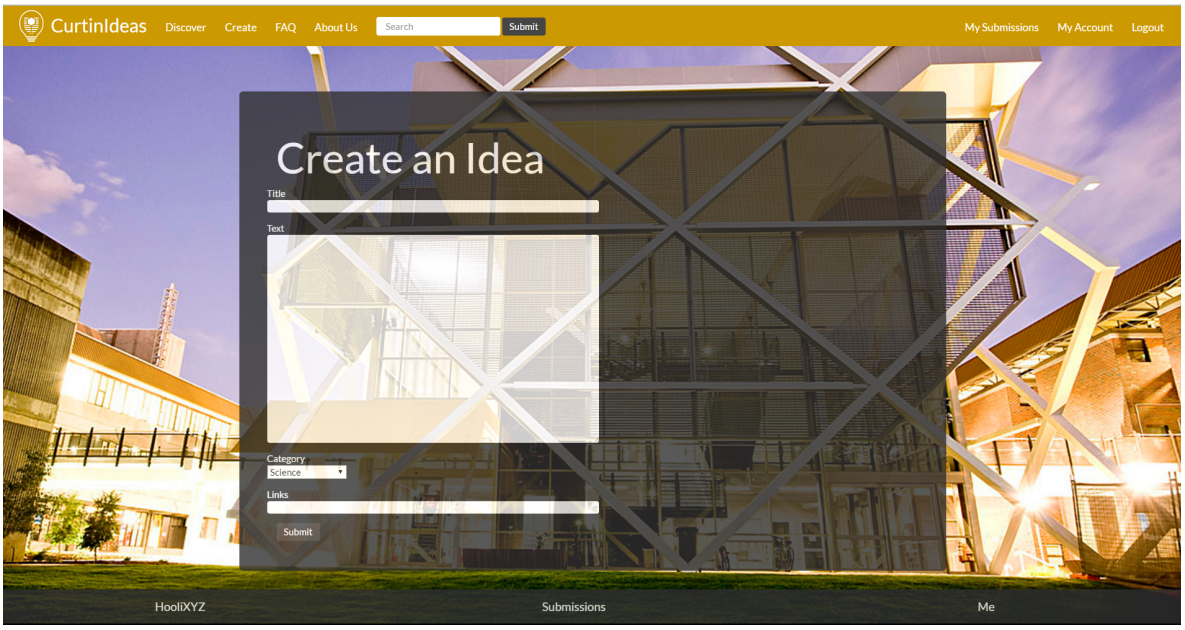


Figure 14: Create Submission Form

The submission model was created to represent a submission, holding all the required information in the database. The subsequent task was to develop a submission form (Figure 14) for a submission to be added to the platform. This form was reused for editing a submission by populating the fields with the current submission and then re-submitting, reducing the amount of duplicate work performed.

Deletion of a submission was restricted to the author of the post or an administrator and was straightforward to implement. The object related to that particular submission was simply removed from the database and the new submission page loaded.

Having the ability to embed photos and videos within a submission was originally planned but was modified after concerns about the time this would take. After deliberation with the project client, the decision was made to instead allow only links to images and videos instead of the embedded alternative. Despite this change, the overall functionality is not modified.

In terms of viewing submissions, any person can browse and read submissions without requiring a user account as requested by the client. A page was created to display the list of current submissions showing the submission title, author, number of views it has received, the date it was posted, the submission category and the number of votes (Figure 15). Additional functionality was added in order to sort the submissions by different criteria and to search submissions based on a keyword(s). Whilst this feature was easily implemented, it provided a large amount of functionality to the user.

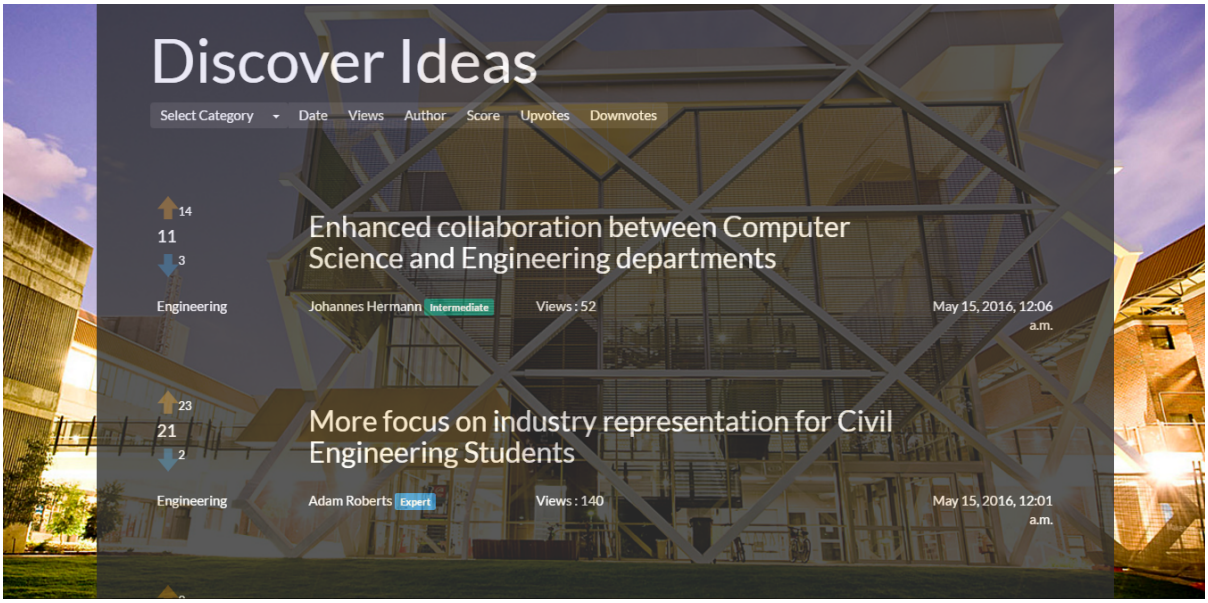


Figure 15: Submission Listing

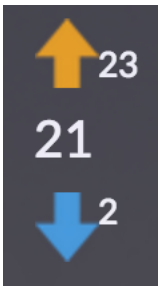
4.1.4 Comments and Improvements

Each submission had the ability to receive both comments and improvements from other users, be they student or staff members. The task breakdown for handling comments included creating a comment model, creating a form to handle the comment input and linking the comment object to the target submission object. At this stage of the project, a lot of similar functionality had already been added for other sections and so the efforts given for the implementation reflected this, with generally only 2 - 3 effort points estimated. To declare whether a comment is also an improvement, a primitive tick-box was added to indicate if the user intends for their comment to be an improvement.

The editing and deleting functionality for comments and improvements was similar to that for editing and deleting a submission. The initial form used for adding a comment was populated with the original comment and then re-submitted after any edits were completed. These simple modifications required little effort due to the underlying functionality already existing in the system.

4.1.5 Voting

The voting functionality was implemented for the submissions, comments and improvements. The tasks that were assigned included: adding clickable arrow images to represent the up-voting and down-voting features (Figure 16), calculating and updating the overall vote total by subtracting the total down-votes from the total up-votes and greying out a user's ability to prevent them voting multiple times on the same submission or comment. The effort estimation for the voting tasks varied from 2 to 3 points for displaying/calculating the totals and the main functionality to a higher effort of 5 points for only allowing a user to vote once per submission.



Voting was restricted to users with accounts and only one vote was permitted per user per submission or comment. The voting model was initially implemented as two fields - up-vote and down-vote - in the submission and comment model before being moved to their own individual models. The decision to move the voting functionality to a separate model was based on the ability to extend voting functionality further in the future.

Figure 16: Voting Buttons

4.1.6 Rewards

The reward system and the gamification of the platform was the final feature to be implemented after all other features were completed and fully functional. As previously discussed, the reward system was designed to 'reward' users for participating in the generation and discussion of ideas, whether that be by creating submissions or voting/commenting on other user's submissions. The plan was to implement a points system to keep track of how active a user is by allocating points per submission, comment or vote. The allocation of points for specific events was discussed and set as follows:

Task Performed	Points Received
Posting a Submission	4
Posting an Improvement	3
Posting a Comment	2
Voting on a Submission	1
Voting on an Improvement	1
Voting on a Comment	1

Figure 17: Points Allocation

Initially a user starts at the base level with 0 points. Every 20 points gained granted a new level, up to a total of 100 points. This was implemented by the addition of two fields to each user model to track both their points and their current level. The system currently only has 5 levels displayed below, but is designed to be easily extended if required.

Level	Points
Beginner	0 – 19
Novice	20 – 39
Intermediate	40 – 59
Advanced	60 – 79
Expert	80 - 100

Figure 18: Platform Levels

5.0 Project Review

5.1 What Went Right

Many aspects of the project went well for the group and contributed greatly to the overall success of the project. Throughout the sprints, the morale of the team was high and the work ethic of each team member was sufficient. Being able to choose our team members contributed greatly to this as group members were already familiar to each other well before the start of the project.

The aspect that helped the most in the product development was the high level of communication between all team members during sprints. All members attended the weekly SCRUM meetings that were held and these worked to guide the team in completing their designated stories for each sprint. In addition to these formal meetings, communication was also performed via a *Facebook* messaging group. This allowed us to keep in touch with one another and relay information quickly when it was needed.

Another team-based aspect of the project that went well was our utilization of pair programming. Every story was allocated not just to an individual, but to a pair of individuals. This greatly enhanced performance as members could bounce ideas off one another and troubleshoot bugs and issues in tandem. As a result, this contributed to the high morale of the group and resulted in high all-round performance and productivity. This also resulted in no major issues arising in the weekly SCRUM meetings.

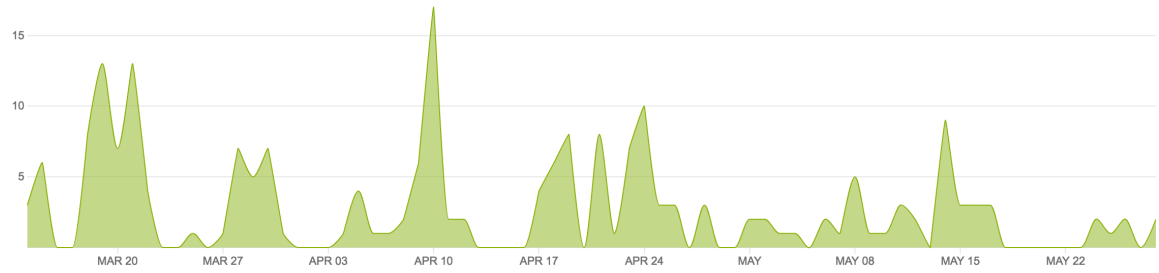


Figure 19: Version Control Commits

Numerous tools utilized throughout the project worked in our favor during product development. The use of version control via the *BitBucket* platform enabled team members to concurrently work on the project without their work interfering with one another. The commits pushed to this platform are shown in Figure 19 and analysis of this helps us to analyze our group and individual contributions over time. It also

enabled us to rollback and help solve any regression errors inserted into the source code. The issue tracking tool *Pivotal Tracker* worked to keep the team on track and provide a metric on how our performance was progressing and whether we were on track or not. Without this, we expect we would not have fully implemented the solution by the final release date.

The final feature of the project that went well was the choice of web development tools employed. The use of the easy-to-learn Django web framework for the development platform meant that team members could learn the relevant skills to perform their required work with ease. It also enabled us to use the PythonAnywhere tool for hosting the site, providing users and the project client with easy access, comparatively to some other hosting solutions. All in all, the majority of aspects of the project went extremely well and the project smoothness as a result of this contributed strongly to our overall performance.

5.2 What Went Wrong

Despite the majority of aspects being productive during the project, some aspects of the project could have been performed better. The major issue was the team's unfamiliarity with both the SCRUM process and web development projects in general. This project was the first undertaken for many of the team members and thus the learning curve was relatively steep. This was shown by our weak performance in Sprints one and two. Our team also had no experience with web development frameworks and thus learning these technologies slowed our progress significantly in the early weeks. This was unavoidable however and we believe we dealt well with this minor setback.

Not having undertaken any previous projects however also led to aspects of poor project management at times. The most significant of these would have been the poor time management shown occasionally. This become a concern at times when members had external commitments and thus productivity was greatly reduced during these periods. The experience learned from these mistakes will be vital in future projects as we further enhance and hone our time management skills.

5.3 What Was Learned

The team has learned a lot from the project and grown significantly as a result. Coming into the project, the team had no experience with project management, web development or the Agile and SCRUM frameworks. Coming out of the project however,

each team member now has a firm understanding of these concepts and experience in applying them to an actual real-world project.

The project management skills and time management skills that were developed will be extremely useful in future projects. Learning how to balance productivity with other curricular and extracurricular commitments is a valuable skill that any member of a project requires and hence learning these skill has been very immensely valuable. The experience gained in general group skills cannot be quantified in a metric, but is also a good skill for each team member to have acquired.

Team members gained experience in several tools and frameworks during the semester. The use of version control was crucial to the success of this sprint and now each member understands how to apply it in future roles. Similarly, the team learnt how to use issue tracking software as a project management tool to ensure the project was kept on track and was an overall success.

The major point of learning throughout the project came in terms of web development. The team developed a full-stack web platform without any prior web development knowledge and thus, everything was learned during the project itself. The Django framework is now considerably understood by the team and can be applied in the future if ever required again. The python language has also been learnt to an extent and this tool will no doubt be of use in the future to all the software engineers involved.

5.4 Future Improvements

The group has several recommendations for future participations of the Project Design and Management unit. These are issues we discovered throughout our participation of the unit and may not be relevant to certain groups and their personal group-dynamic.

Firstly, the weighting of the project was weighted heavily towards the SCRUM process and the Sprint and Final reports. We propose placing some of this weighting into the platform and its actual functionality. Thus, teams that produce a fully working and functional solution can receive marks for this. As a team that completed all functionality, this would have benefitted us significantly.

During the project, the one-on-one time with the project client was extremely limited. Through Sprint 3, only a small 10-minute window was held with the client. We feel that more time with the project client to discuss specifics on aspects such as styling and functionality implementation would benefit both our team and the final solution.

However, this may be difficult considering the time constraints of both parties and the constraints of a university unit in general.

6.0 Conclusions and Summary

The Agile project management technique utilized during the project development ensured the success of the platform and its timely final delivery to the project client. The use of the SCRUM framework also ensured this success, with its incremental approach proving effective amongst the team. The Django web development framework provided a strong foundation for the system to be designed and implemented upon, resulting in a quality, polished product. The high levels of communication between all team members also contributed to the overall success of the project as a whole.

Each team member gained a substantial amount of project experience throughout the development of the Curtin Ideas platform and this experience will be valuable in any future projects we undertake. Agile technique, the SCRUM framework and strong web development skills were gained during the project and these skills have enhanced each team member's knowledge and experience significantly. In future projects, the team will definitely push for an Agile project management technique over a traditional Waterfall model due to its overwhelming success during this project.

The following lists indicate expansions and new features that could be implemented in future releases of the product. The project client may wish to implement these recommendations into the platform over time to enhance the solution.

Expansions:

- Enhanced reward system with added rewards upon reaching certain levels
- Embedded images and videos for links in submissions
- Enhanced user profiles, enabling user to upload profile pictures and modify their details

New Features:

- Implementation of a scoreboard, listing the top 10 contributors for each category
- Department added to user profiles and on posting, to link users from departments
- Dynamic page background that change over time

7.0 References

Grellier, Jane, and Veronica Goerke. 2010. *Communication Skills Toolkit*. 1st ed. South Melbourne, Vic.: Cengage Learning.

"Hacker News." Hacker News. Accessed March 29, 2016. https://news.ycombinator.com/.

Park, Thomas. "Darkly." Bootswatch:. Accessed March 28, 2016. https://bootswatch.com/darkly/.

"Pivotal Tracker | Agile Project Management". 2016. *Pivotal Tracker*. https://www.pivotaltracker.com.

"Reddit: The Front Page of the Internet." Reddit: The Front Page of the Internet. Accessed March 29, 2016. https://www.reddit.com/.

Schwalbe, Kathy. *Information Technology Project Management.* 7th ed. Boston, MA: Thomson Course Technology, 2006.

Twalib, Salim. *"Agile Lecture."* Class lecture, Project Design and Management from Curtin University, Perth, Australia, February 29th 2016.

8.0 Appendices

8.1 Appendix A: Sprint Reports

The full Sprint reports for sprints 1 through 4 have been included as an attachment to this final report.

8.2 Appendix B: Pivotal Tracker Backlog

All data tracked within the issue-tracking tool *Pivotal Tracker* has also been included as a *csv* attachment to this final report.