496 Software Project:
Colloquium Abstract Manager

Matthew McCurdy and Andrew Dominguez

2026-02-17

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

- Client name: Selin Gursozlu
- Client title: Chair of the Undergraduate Research Colloquium committee
- Client email address: sgursozlu@loyola.edu
- Client employer: Loyola Department of Philosophy
- How you know the client: Referral from Dr. Isaacman and Dr. Bui

2 Project Description

2.1 Overview

Our client Dr. Gursozlu is asking our team to create a website to organize abstracts for the Undergraduate Research Colloquium. Students will submit abstracts to the website, and committee members will assign abstracts to reviewers who will review and provide feedback to students. Students will then be informed by the website if their abstract was accepted or not.

2.2 Key Features

- Three user types, Students, Committee Members, and Reviewers
- Submission Forum - Students will submit abstracts to this page
- Abstract list - Committee members and reviewers will be able to see a list or gallery of the submitted abstracts where they can view them and committee members can assign them to reviewers
- View pages - Each submission will have a view page where the project can be viewed by users, reviewers can leave feedback for the student authors and accept or deny the submission. This page will also display the acceptance status of the submission to the authors.

2.3 Why this Project is Interesting

We think that this project is interesting because it involves building a website with a full CRUD which would also be used by Loyola students and faculty. This will make submission and management of the Colloquium more convenient and should be of interest to anybody who intends to submit research to the Colloquium.

2.4 Areas of CS required

- Software Engineering
- Web Development
- Database Management

2.5 Potential Concerns and Questions

We are unsure if the project meets the requirements of including multiple sub disciplines of computer science, as those listed above may be too broad.

3 Comparison to Draft

Neither partner drafted a project proposal in November, the client was put in contact with Matthew. The project was chosen as neither of us had other clients we had contacted.

4 Requirements

4.1 Non-Functional Requirements

[Non-functional requirements are just as important as functional requirements. Dont forget to specify them.]

Table 1 presents the NFRs for this software project.

ID	NFR Title	Category	Description
NFR1	Page Permission	Security	Users should only be able to access pages and functions which their account has access to.
NFR2	Color Scheme	Usability	Color scheme should be a Loyola-consistent (green) color scheme.
NFR3	Include logo	Usability	The Loyola University Maryland should be present in the website.
NFR4	FAQ Page	Usability	There should be an FAQ page which answers some frequently asked questions.
NFR5	Password Hashing	Security	User passwords should be hashed when stored.

Table 1: Non-Functional requirements

4.2 Functional Requirements (User Stories)

ID	Story Title	Points	Description
U1	Login/Logout	1	As a user, I want to be able to login and logout of the website.
U2	Abstract Gallery	3	As a user, I want to be able to view a gallery of approved abstracts.
U3	Account Creation	2	As a user, I want to be able to create an account as my user type.
U4	Previous Winner Gallery	2	As a user, I want to be able to see the winners from previous years.
U5	Search Abstracts	3	As a user, I want to be able to search for words in the title of abstracts so that they are easier to find.
U6	Filter Abstracts	3	As a user, I want to be able to filter the abstracts by section so they are easier to find.
S1	Abstract Submission	5	As a student, I want to submit an abstract for review.
S2	Track Submission Status	3	As a student, I want to be able to see the status of the review for my abstract.
S3	Edit Submission	3	As a student, I want to be able to edit my submitted abstract.
S4	Abstract Comment	3	As a student, I want to be able to add a comment to an abstract in the gallery.
S5	Notifications	5	As a student/committee member, I want to receive a notification when my abstract has been reviewed/there is an abstract to review.
S6	View Feedback History	3	As a student, I want to be able to view the history of reviews on my abstract.
S7	Draft Submission	2	As a student, I want to be able to save a draft of my submission to return to later.
C1	Assign Abstracts	2	As a committee member, I want to be able to assign abstracts to reviewers.
C2	Finalize Approval	1	As a committee member, I want to be able to give final approval for abstract to be on the gallery.
C3	Approve User	2	As a committee member, I want to be able to approve new users to be students or reviewers.
C4	Committee Member Info	2	As a committee member, I want my contact information to appear on the website.
C5	Announcements	5	As a committee member, I want to be able to send announcements to all students and/or reviewers so that they can stay notified.
C6	Internal Notes	2	As a committee member, I want to be able to add notes to abstracts and feedback so that the committee can review them later.
R1	View Assigned Abstracts	1	As a reviewer, I want to be able to view the abstracts assigned to me.
R2	Give Reviewer Feedback/Approval	3	As a reviewer, I want to be able to give feedback on the reviewed abstract.
R3	Reviewer Application	2	As a reviewer, I want to be able to submit an application to apply to review.
R4	Reviewer Feedback History	3	As a reviewer, I want to be able to see previously submitted feedback for that abstract so that reviews can stay consistent.
R5	Reviewer Feedback Draft	2	As a reviewer, I want to be able to submit a draft of my review so that I can save it and return later.
A1	Committee Member Approval	1	As an admin, I want to be able to approve new committee members.
A2	Manage Accounts	3	As an admin, I want to be able to manage accounts so that inactive accounts can be deleted.
A3	Edit Page Content	5	As an admin, I want to be able to edit the content on all of the pages so that they remain up-to-date.
O1	User Dashboards	3	As a user, I want to be able to view the relevant dashboard for my user type so that I can see all of the relevant information.

Table 2: Functional requirements as User Stories.

5 System Design

5.1 Architecture

For this project, we will be using the MVC architecture. We will organize the files into three main folders: "Model", "View", "Controller". The "View" folder will have the HTML files and the CSS file for the UI designs. The "Model" folder will have the JavaScript files related to handling the connections to the MongoDB database along with using Mongoose. The "Controller" folder will have the JavaScript files that contain functions which handle requests and responses for each feature.

5.2 Diagrams

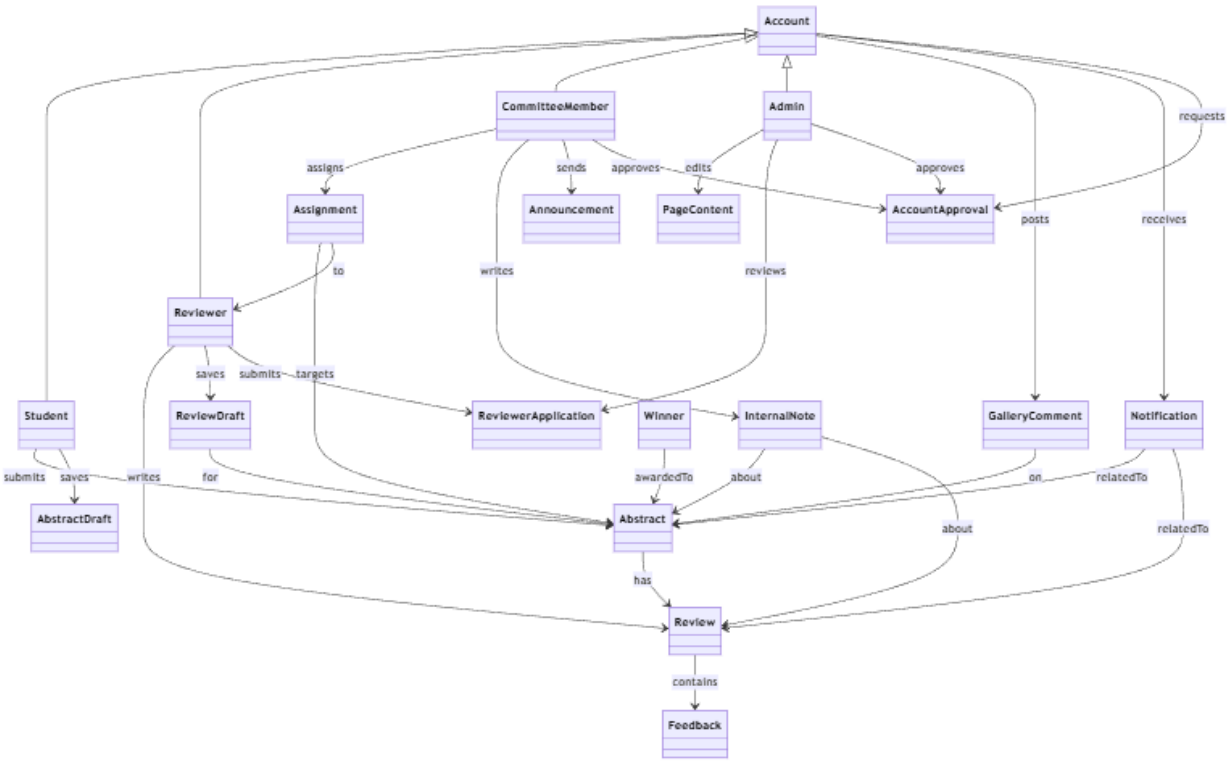

Figure 1: Pre-Iteration 1: Simplified Class Diagram

5.3 Technology

The programming languages which we will use throughout this project will be JavaScript, HTML, and CSS. We chose these languages due to prior experience and so that we can use the same language for client and server side. For the website's environment, we will be using Node.js along with Express for HTTP requests. For our database management, we will be using MongoDB. For our testing framework, we will be using Jest.

5.4 Coding Standards

Our team will follow consistent coding standards to ensure readability, maintainability, and reliability throughout development. Only code which has been tested and is working may be pushed to the repository. All commits must contain the issue number as well as a description clearly describing the changes. Any commented-code or internal developer notes will be removed before release of the software. All variable declarations will use const and let, instead of var, and complex methods will include comments explaining their purpose. File names will follow the PascalCase format, with variables and functions using camelCase, and constants being written in UPPERCASE with underscores. Test function names and descriptions must clearly state the function being tested along with the expected outcome.

5.5 Data

The abstract manager website will be comprised of students, reviewers, committee members, and an admin. The students will have a unique ID, their name, username, email, and password. The data stored for reviewers and committee members will be similar but with the addition of their subject field. The admin will be one of the committee members which will be stored as a flag in their data. The website will have an approved abstract gallery which will store the approved abstracts and make them available to view. Each abstract will contain it's title, description, last submission date, student name, subject field, all feedback given, and approval status.

5.6 UI Mocks

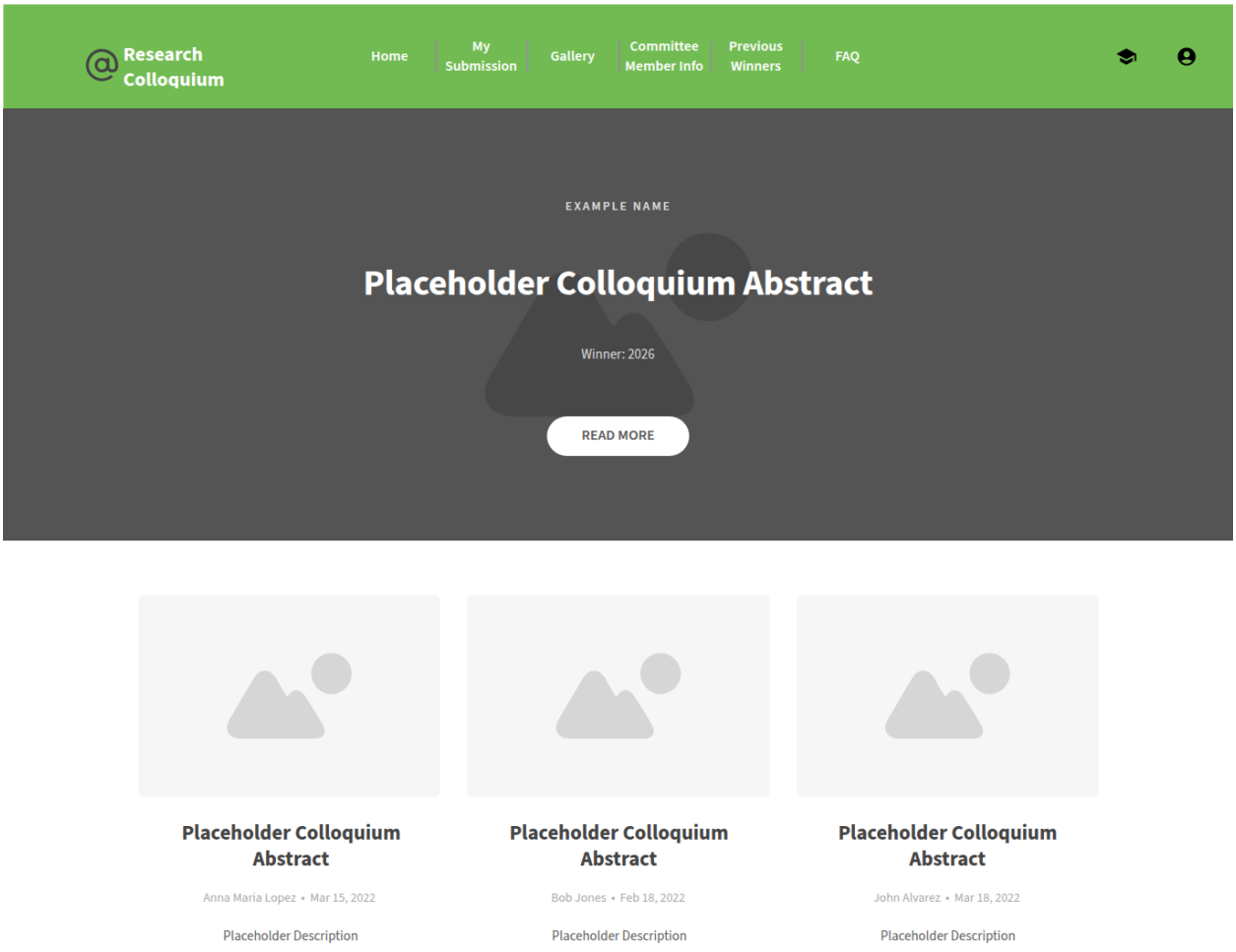

Figure 2: Mock UI of the home/landing screen of the website.

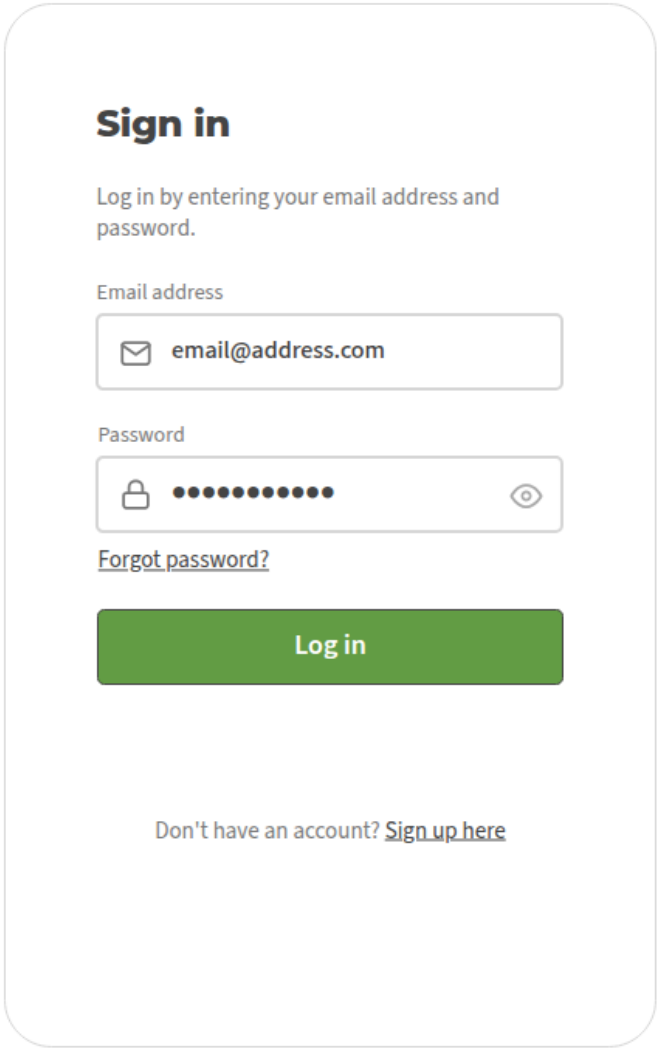

Figure 3: Mock UI of the login screen of the website.

6 Iterations

6.1 Iteration Planning

Table 3 shows the iteration planning.

It.	Dates	Stories	Points Planned	Points Done
1	01/29 - 02/10	U1 Login/Logout, U3 Account Creation, C3 Approve User, R3 Reviewer Application, A1 Committee Member Approval, A2 Manage Accounts	11	11
2	02/10 - 02/24	S1 Abstract Submission, S3 Edit Submission, S7 Draft Submission, C4 Committee Member Info, O1 User Dashboards	15	0
3	02/24 - 03/17	S5 Notifications, S6 View Feedback History, C1 Assign Abstracts, R1 View Assigned Abstracts, R2 Give Reviewer Feedback/Approval, R4 Reviewer Feedback History, R5 Reviewer Feedback Draft	19	0
4	03/17 - 03/31	U2 Abstract Gallery, U5 Search Gallery Abstracts, S2 Track Submission Status, S4 Abstract Comment, C2 Finalize Approval, U4 Previous Winner Gallery, U6 Filter Gallery Abstracts	18	0
5	03/31 - 04/14	C5 Announcements, C6 Internal Notes, A3 Edit Page Content	12	0
		Total:	75	11

Table 3: Iteration Planning for Incremental Deliveries

6.2 Iteration/Sprint 1

6.2.1 Planning

For iteration 1, we planned to complete: U1 Login/Logout, U3 Account Creation, C3 Approve User, R3 Reviewer Application, A1 Committee Member Approval, A2 Manage Accounts, for a total of 11 points. Our goal for this iteration is to implement the account functions and give a baseline structure for the website. Users should be able to create accounts and then login/logout. Reviewer accounts should be able to submit an application to review. Committee Members should be able to approve all users. In the future, this approval will lead to reviewers being allowed to review/students being allowed to submit an abstract. The admin should be able to manage all of the accounts created on the site, along with deleting any account.

6.2.2 Work Done

We completed all of the stories planned for this iteration. We have a working account creation and management system, login/logout functionality, and the ability for reviewers to submit applications. We also implemented password hashing for all of the accounts.

6.2.3 Testing Coverage

So far, we have decent coverage throughout the project. Our DAO files mostly tested with only a few lines missing between them. The controller and server functions currently have insufficient testing. On iteration 2, we plan to improve the testing coverage throughout and bring the controller and server files up to standard.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	79.57	77.16	72	82.52	
AbstractsWebsite	62.16	50	0	65.71	
server.js	62.16	50	0	65.71	32-39,46-52,65-66,70-71
AbstractsWebsite/controller	71.83	71.42	68.75	74.79	
homeController.js	71.83	71.42	68.75	74.79	19-20,62-64,72-80,113,130-153,266,327-331,412,435-448
AbstractsWebsite/model	95.06	86.25	100	100	
accountDao.js	96.49	88.23	100	100	23-24,61-62,90
applicationDao.js	91.66	82.75	100	100	10-12,34
AbstractsWebsite/routes	100	100	100	100	
htmlRoutes.js	100	100	100	100	

Figure 4: Iteration 1 test coverage report

6.2.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

In this iteration, we were struggling with the setup of the database and server which led us to make less early progress than expected.

Andrew - I personally felt that this iteration went well and we were able to complete all of the stories which we planned. Looking back on it, I should have spent more time writing tests for the files so that we can ensure they are working properly and meet the testing standards.

6.3 Iteration/Sprint 2

6.3.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

6.3.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

6.3.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

6.3.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6.4 Iteration/Sprint 3

6.4.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

6.4.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

6.4.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

6.4.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6.5 Iteration/Sprint 4

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482]

6.5.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

6.5.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

6.5.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

6.5.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6.6 Iteration/Sprint 5

6.6.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

6.6.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

6.6.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

6.6.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

7 Final Remarks

7.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project. Be concrete about your progress, you know how many story points your software is, how many points you completed (this shows your progress). You also how many points your team delivers at each iteration, therefore you can estimate how many more iterations it would take to finish the leftover points (show the math).]

7.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently? How can you do better work in future projects? You may write this as a team or per person (or both — if all your iterations were team reflections, then it would be better to write individual reflections here)]

Appendix

[Appendix section if needed]