

CS3354 Software Engineering
Final Project Deliverable 2

Gateway

Chloe Lee
Hudson Apel
Hyrum Moses
Justin Tran
Noah Wohler
Dang Thien Nguyen
Esterban Kim

1. Delegation of tasks

Group Member	Responsibilities
Chloe Lee	GitHub First Commit, Presentation Slides
Hudson Apel	Class Diagram, Support and editing
Hyrum Moses	Test Plan/Unit Test Code
Justin Tran	Sequence Diagram, Estimation of Costs
Noah Wohler	Use Case Diagram, FP, Comparison of Work
Dang Thien Nguyen	Project Scope, Conclusion
Esteban Kim	Management, GitHub Setup, Architectural Design, Project Scheduling

2. Project Deliverable 1 Content

<https://github.com/EstebanKim11/3354-Gateway/blob/main/CS3354-ProjectDeliverable1-Gateway.pdf>

3. Project Scheduling, Cost, Effort and Pricing Estimation

3.1. Project Scheduling.

A member of the team is expected to work 8 hours per day but not during the weekends. We expect to have seven team members within the team collaborating towards the completion of a MVP prototype website.

Since we are striving to produce a MVP prototype of Gateway, we are scheduled to complete the application in one week. Multiple parts of the project can be accomplished in parallel, for example, the front-end, back-end, and database setup does not require a team to complete before the next can move forward. The initial design can be used as a means to implement the website and any major changes to the design that would impact multiple teams can be communicated.

We expect to start implementing Gateway on December 5th and final delivery of the website will be on December 9th. December 5th and 6th will be used by each group working on the front-end, back-end, and database to individually complete their implementations. December 7th will be used to bring multiple pieces together to ensure that the website is fully functional. December 8th will be used to fix any bugs or unexpected issues. On December 9th, we expect to deliver and provide a live demo of the website.

3.2. FP

	Function Category	Count	Complexity			Count × Complexity
			Simple	Average	Complex	
1	Number of user input	8	3	4	6	24
2	Number of user output	4	4	5	7	16
3	Number of user queries	2	3	4	6	6
4	Number of data files and relational tables	2	7	10	15	14
5	Number of external interfaces	1	5	7	10	5
GFP						55

Processing complexity adjustment (PCA):

- (1) **Does the system require reliable backup and recovery?**
A: 2 - moderate (backup is necessary but data is not critical)
- (2) **Are data communications required?**
A: 3 - average (yes, data must be communicated between server and clients)
- (3) **Are there distributed processing functions?**
A: 0 - no
- (4) **Is performance critical?**
A: 3 - average (performance is critical for the user experience, but a lack of performance will not cause a catastrophe)
- (5) **Will the system run in an existing, heavily utilized operational environment?**
A: 0 - no
- (6) **Does the system require online data entry?**
A: 5 - essential (yes, it is a website and all of the data will be entered online)
- (7) **Does the online data entry require the input transaction to be built over multiple screens or operations?**
A: 0 - no
- (8) **Are the master files updated online?**
A: 4 - significant (yes, the data files will be stored in an online database)
- (9) **Are the inputs, outputs, files, or inquiries complex?**
A: 1 - incidental (mostly just strings and integers)
- (10) **Is the internal processing complex?**
A: 1 - incidental (not much internal data processing, mostly just updating, searching, displaying, etc.)
- (11) **Is the code designed to be reusable?**
A: 0 - no
- (12) **Are conversion and installation included in the design?**

A: 0 - no

- (13) **Is the system designed for multiple installations in different organizations?**

A: 0 - no

- (14) **Is the application designed to facilitate change and ease of use by the user?**

A: 4 - significant (yes, the application will be designed to make it easy for people to use)

$$\text{PCA } 0.65 + 0.01(23) = 0.65 + 0.23 = \mathbf{0.88}$$

$$\text{FP} = \text{GFP} \times \text{PCA} = 55 \times 0.88 = \mathbf{48.4 \text{ FP}}$$

$$\text{E} = \text{FP}/\text{productivity} = 48.4/60 \approx 0.8067 \approx \mathbf{1 \text{ person-week}}$$

Team size = 7, so the project duration would be

$$\text{D} = \text{E} / \text{team size} = 7 \text{ days}/7 \text{ people} = \mathbf{1 \text{ day.}}$$

3.3. Estimated Cost of Hardware Products

The estimated cost of hardware products is primarily found with server costs. However, these costs can be very easily mitigated with the usage of AWS in order to host our website. This cuts down our hardware costs to a more manageable number of 1 to 3 dollars a month depending on how much server traffic we receive [1]. This leads to an estimated annual cost of hardware products to be at a maximum of \$36.

3.4. Estimated Cost of Software Products

Due to the fact that the development of our website is not expected to use any paid software products, the cost of software products used for this project will be \$0.

3.5. Estimated Cost of Personnel

For an in-house web designer, the starting rate of pay is around \$60 per hour [2, 3]. According to the Function Point calculation, with all 7 team members working, development will take about a day which is the equivalent of around 8 hours considering an average work day's length. All together, multiplying these numbers results in a total estimated cost of \$3360 overall for the website's development.

4. Test Plan

The testing for the login page consists of verifying that each component is correctly rendered on the screen, that only valid emails and passwords are able to be submitted, and that only the correct email/password combination will successfully log the user into their account. The test cases are as follows:

- Initial rendering of the page, verifying that the email input field, the password input field, and the submit button are all correctly rendered.
- Attempted submission with empty password and email fields
- Attempted submission with invalid email and empty password field
- Attempted submission with valid but incorrect email and empty password field
- Attempted submission with correct email and empty password field
- Attempted submission with empty email field and incorrect password
- Attempted submission with invalid email and incorrect password
- Attempted submission with valid email and incorrect password
- Attempted submission with correct email but incorrect password
- Attempted submission with empty email field and correct password
- Attempted submission with invalid email and correct password
- Attempted submission with valid email and correct password
- Attempted submission with correct email and correct password

5. Comparison of Work

The most similar design out there to our own project is Glassdoor. Glassdoor is a website which hosts job and salary information, company reviews, and interview questions. Users can search for a company and find detailed information about them and the jobs that they offer. Besides interview questions, users can view salary ranges, reviews, and more [4]. Glassdoor is a much broader service than our project will be. They include much more information about the job and company, whereas we will solely be focused on interview questions. Because of that focus, our service will be a greater resource for interview preparation with less clutter and more information available about specific questions. Users will be able to quickly find the questions they are looking for without being distracted by reviews and other information, making it easier to use.

6. Conclusion

In this deliverable 2, we are sticking to the plan laid out in deliverable 1, but added onto some of the original ideas. We included new features such as an upvote and downvote system, and the homepage will now show the top results to make the search process more manageable. We also analyzed the requirements for scheduling the project and estimating the cost to implement and sustain the Gateway website, which will serve as an easier way for users to prepare for potential interview questions.

7. References

- [1] "Frequently Asked Questions," *Amazon Web Services (AWS)*, 2022. [Online]. Available: <https://aws.amazon.com/getting-started/projects/host-static-website/faq/>. [Accessed: 10-Nov-2022].

- [2] Arc, “Web Developer Hourly Rate (2022),” *Arc*, 2022. [Online]. Available: <https://arc.dev/freelance-developer-rates/web>. [Accessed: 10-Nov-2022].
- [3] Ripe Media, “A web design pricing guide: This Is What You Can Expect to Pay: Ripe media,” *Marketing, Branding, Design, and Development in Los Angeles | Ripe Media*, 18-Aug-2021. [Online]. Available: <https://www.ripemedia.com/a-web-design-pricing-guide-this-is-what-you-can-expect-to-pay/>. [Accessed: 10-Nov-2022].
- [4] “About Us,” *About Us - Glassdoor About Us*, 2022. [Online]. Available: <https://www.glassdoor.com/about-us/>. [Accessed: 10-Nov-2022].

8. Presentation Slides
[CS3354 Project Slides - Google Slides](#)

9. OPTIONAL: Program Code

10. GitHub Requirements

Pushed?	GitHub Requirement
Esteban	Project Deliverable 2
Hyrum	Unit Test Code
	Implementation (Optional)
Chloe	Presentation Slides