

Clinical Competency Calculator

Software Requirements Specification
Draft Document

Dylan Colburn
Jacq Lee
Tyler Muessig
Tyler Price

v0.0.4
Dec. 12, 2024

Contents

List of Figures	ii
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Client information	1
1.4 Users	1
1.5 Definitions	1
1.6 Acronyms and abbreviations	2
1.7 References	3
2 Overall description	3
2.1 Product perspective	3
2.2 Product functions	3
2.2.1 Student functions	3
2.2.2 Rater functions	4
2.2.3 Administrator functions	4
2.3 User characteristics	4
2.4 Constraints	4
2.5 Assumptions	4
3 Specific requirements	4
3.1 External interface requirements	4
3.1.1 User interfaces	4
3.1.2 Hardware interfaces	4
3.1.3 Software interfaces	5
3.2 Functional requirements	5
3.2.1 User authentication and authorization	5
3.2.2 Feedback form request and submission	5
3.2.3 Form response quality control	7
3.2.4 Full report	9
3.2.5 Other functional requirements	9
3.3 Non-functional requirements	9
3.3.1 Performance requirements	9
3.3.2 Design constraints	10
3.3.3 Regulatory compliance	10
3.3.4 Backup and recovery	10
3.3.5 User training and documentation	10
Appendices	11
A Software engineering tools	11
A.1 Version control	11
A.2 Issue tracking	11
A.3 Automated testing	11

List of Figures

3.1	Use case diagram for the Clinical Competency Calculator.	6
3.2	Functional requirements sorted topologically by dependency.	7
3.3	Robustness diagram describing the process of a rater filling in a feedback form.	8
3.4	Sequence diagram describing the process of a rater filling in a feedback form.	8

1 Introduction

1.1 Purpose

The purpose of the software requirements document is to create a comprehensive outline of the Clinical Competency Calculator (CCC) for the development team, project advisors, faculty advisors, stakeholders, and members of the Hershey Medical team that will be maintaining this software. This outline will describe the functions and high-level requirements of the CCC. The target audience is expected to know technical terminology related to software development.

1.2 Scope

This document specifies the CCC as software designed to be used for the appraisal of **students** undergoing **clerkship** at Hershey Medical, as is to be utilized by the Department of Family Medicine. The CCC will provide a means of collecting feedback from **raters** on the **competencies** of students and performing useful, actionable analyses thereon. The CCC is not designed to interface with existing Hershey Medical systems or engage in information exchange therewith. The CCC is also not designed for utilization by other departments within Hershey Medical; should this software be used outside of the stated scope, further requirement elicitation should be carried out and relevant issues addressed.

1.3 Client information

The development of the CCC is being sponsored by Anthony Dambro (adambro@pennstatehealth.psu.edu) from the Milton S. Hershey Medical Center Department of Family Medicine.

1.4 Users

Our intended audience for the CCC would be the students undergoing clerkship, raters providing feedback on students' competencies, and **administrators** of the entire system and its data.

1.5 Definitions

The definitions for terms required to properly interpret this document are listed below. The first mention of each term in this document (excluding this subsection) is highlighted in teal and boldface, and is a hyperlink that will redirect to this subsection when clicked.

Words listed in uppercase indicating requirement levels (MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY) are adapted from RFC 2119 and follows the specification by RFC 8174 that these defined special meanings only apply to uppercase usage of these terms.

In general, the definitions of terms used in this document not listed below conform to the definitions provided in IEEE Std 610.12-1990.

- 1.5.1. **administrator**: A user with elevated privileges who can perform administrative tasks, with similarly elevated authority with respect to the clerkship and Hershey Medical as a whole. Such users include the Assistant Dean for Education in the Clinical Learning Environment,

Clerkship Directors, Director of the Phase 3 UME-GME Transition, and the Associate Dean for Student Assessment and Program Evaluation. 1, 3–5, 7, 9, 10

- 1.5.2. **clerkship:** A period of training where medical students practice medicine under the supervision of experienced physicians. Students work directly with patients, conduct exams, and begin to manage cases. 1, 3
- 1.5.3. **competency:** An observable ability of a health professional, integrating multiple components such as knowledge, skills, values, and attitudes. Since competencies are observable, they can be measured and assessed to ensure their acquisition. 1, 3, 5, 9
- 1.5.4. **Entrustable Professional Activity:** Units of professional practice, defined as tasks or responsibilities that trainees are entrusted to perform unsupervised once they have attained sufficient specific competence. EPAs are independently executable, observable, and measurable in their process and outcome, and, therefore, suitable for entrustment decisions. 3, 9
- 1.5.5. **feedback form:** The digital form through which raters will assess students' competencies. 3–5, 7, *see 1.5.10 rater & 1.5.11 student*
- 1.5.6. **form response:** A particular instance of feedback that a rater provides by a single submission of a completed feedback form. 4, 5, 7, 9, *see 1.5.5 feedback form*
- 1.5.7. **full report:** A report of a specific student's competencies, including a summary of the student's achievement within a specific EPA, their strengths within that EPA, and areas/suggestions for improvement. 3, 9, *see 1.5.3 competency & 1.5.4 Entrustable Professional Activity*
- 1.5.8. **MUST:** This word, or the terms "REQUIRED" or "SHALL", means that the definition is an absolute requirement of the specification. 5, 7, 9
- 1.5.9. **MUST NOT:** This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification. 7, 9
- 1.5.10. **rater:** Any individual providing feedback on a student's level of competency. This encompasses faculty, nurses, clerkship advisors, and patients. 1, 3–5, 7, 9, 10
- 1.5.11. **student:** The medical student being assessed through the Clinical Competency Calculator. 1, 3–5, 7, 9, 10
- 1.5.12. **user:** Any person who interacts with the product, namely the students, raters, and administrators. 4, 5

1.6 Acronyms and abbreviations

The definitions of acronyms and abbreviations required to properly interpret this document are listed below. The first mention of each abbreviation in this document (excluding this subsection) will be accompanied by its full long form.

- 1.6.1. **EPA:** Entrustable Professional Activity 3, 9, *see 1.5.4 Entrustable Professional Activity*
- 1.6.2. **CCC:** Clinical Competency Calculator 1, 3–5
- 1.6.3. **CSV:** comma-separated values 9
- 1.6.4. **GUI:** graphical user interface 4

- 1.6.5. **HTML:** Hypertext Markup Language 9
- 1.6.6. **PDF:** Portable Document Format 9
- 1.6.7. **PSU:** Pennsylvania State University 4
- 1.6.8. **RDBMS:** relational database management system 5

1.7 References

Core Entrustable Professional Activities For Entering Residency: Faculty And Learners' Guide. Association of American Medical Colleges, Washington, D.C. Accessed: Nov. 27, 2024. [Online]. Available: https://store.aamc.org/downloadable/download/sample/sample_id/66/%20

IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990, IEEE Standards Board, Sep. 1990. Accessed: Nov. 27, 2024. [Online]. Available: https://www.informatik.htw-dresden.de/~hauptman/SEI/IEEE_Standard_Glossary_of_Software_Engineering_Terminology%20.pdf

Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, RFC Editor, Mar. 1997. Accessed: Nov. 27, 2024. [Online]. Available: <https://www.rfc-editor.org/info/rfc2119>

Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, RFC 8174, RFC Editor, May. 2017. Accessed: Nov. 27, 2024. [Online]. Available: <https://www.rfc-editor.org/info/rfc8174>

2 Overall description

2.1 Product perspective

The CCC is a grading tool to be used to assess the competencies of students undergoing clerkship. The CCC aims to replace and improve upon the preexisting method of collecting feedback from raters by aggregating more useful feedback and providing more insightful analyses. Via this system, students and administrators will have the ability to review student performance as reported by raters. Administrators will also be able to filter low-quality feedback. All of these options will be implemented to allow students, raters, and administrators to access them via any device that has a modern web browser, such as smartphones, laptops, and desktop computers.

2.2 Product functions

2.2.1 Student functions

Students must be able to request a specific rater to fill out a **feedback form** for themselves. Students must also be able to view a **full report** of their performance, which displays where they land along the **Entrustable Professional Activities (EPAs)** and allows the student to view specific comments or other relevant feedback that supports the specific assessment.

2.2.2 Rater functions

Raters must be able to fill out new feedback forms by accepting requests from students.

2.2.3 Administrator functions

Administrators must be able to view all feedback forms submitted, along with the rater who submitted the forms. Administrators must be able to view the full report of any student. Administrators must also be able to audit and reject **form responses** that are of poor quality or otherwise not useful.

2.3 User characteristics

The **users** of the CCC will have a basic sense of computer literacy including minimal keyboard skills and experience interacting with a browser application prior. No advanced knowledge of computer systems is needed. A user should be able to learn the user interface swiftly. Additionally, raters must have sufficient professional knowledge regarding the questions that are asked in the feedback form and students must be able to understand the feedback and analysis given.

2.4 Constraints

Any device with access to a modern web browser should be able to access and run this application assuming that the device has a monitor/screen to view/interact with the graphical user interface (GUI). An internet connection is required to access the web application.

2.5 Assumptions

The students, raters, and administrators are assumed always to have the correct information to access their respective accounts and the hardware capability to input that information. It is also assumed that the database being used is or will be heavily adopted and will likely have support in the near future to ensure an up-to-date system. The application is also assumed to be deployed in an environment with a stable network.

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

The user interface will be a web application that can be accessed through a modern web browser.

3.1.2 Hardware interfaces

The hardware interface will be any device, such as a computer or equivalent, that is capable of accessing the internet through a modern web browser.

3.1.3 Software interfaces

3.1.3.1 User authentication and authorization

The system must allow user authentication and authorization features to ensure proper access. The system will interface with the Pennsylvania State University (PSU) authentication system for user role validation, which includes password access and two-factor authentication. Users will only be able to view data and perform functions as permitted by their authorization level.

3.1.3.2 Database management system

The CCC must interface with a database management system to store and retrieve data. The database will be used to store form response data, user data, and other relevant information.

The database system should be a relational database management system (RDBMS) such as MySQL or PostgreSQL.

The database must be able to retain data for a minimum of two (2) years.

3.1.3.3 Web application framework

The main interface of the CCC will be a web application. The web application will be built using a modern web application framework such as React.

3.1.3.4 Server

The server will be responsible for handling requests from the web application and interfacing with the database. The server will be hosted by Hershey Medical.

3.2 Functional requirements

In this section, “the system” refers generally to the CCC. A use case diagram for the system is provided in Figure 3.1 (p. 6).

3.2.1 User authentication and authorization

- 3.2.1.1. The system **MUST** allow each user to authenticate and log in to access their role-based permissions.

Importance: ■ HIGH

- 3.2.1.2. Administrators **MUST** be able to manage user roles for students, raters, and other administrators.

Importance: ■ HIGH

Dependencies: 3.2.1.1

3.2.2 Feedback form request and submission

- 3.2.2.1. The system **MUST** provide a feedback form for raters to assess the competencies of a particular student.

Importance: ■ HIGH

- 3.2.2.2. The rater **MUST** be able to submit assessments of students via a feedback form.

Importance: ■ HIGH

Dependencies: 3.2.2.1

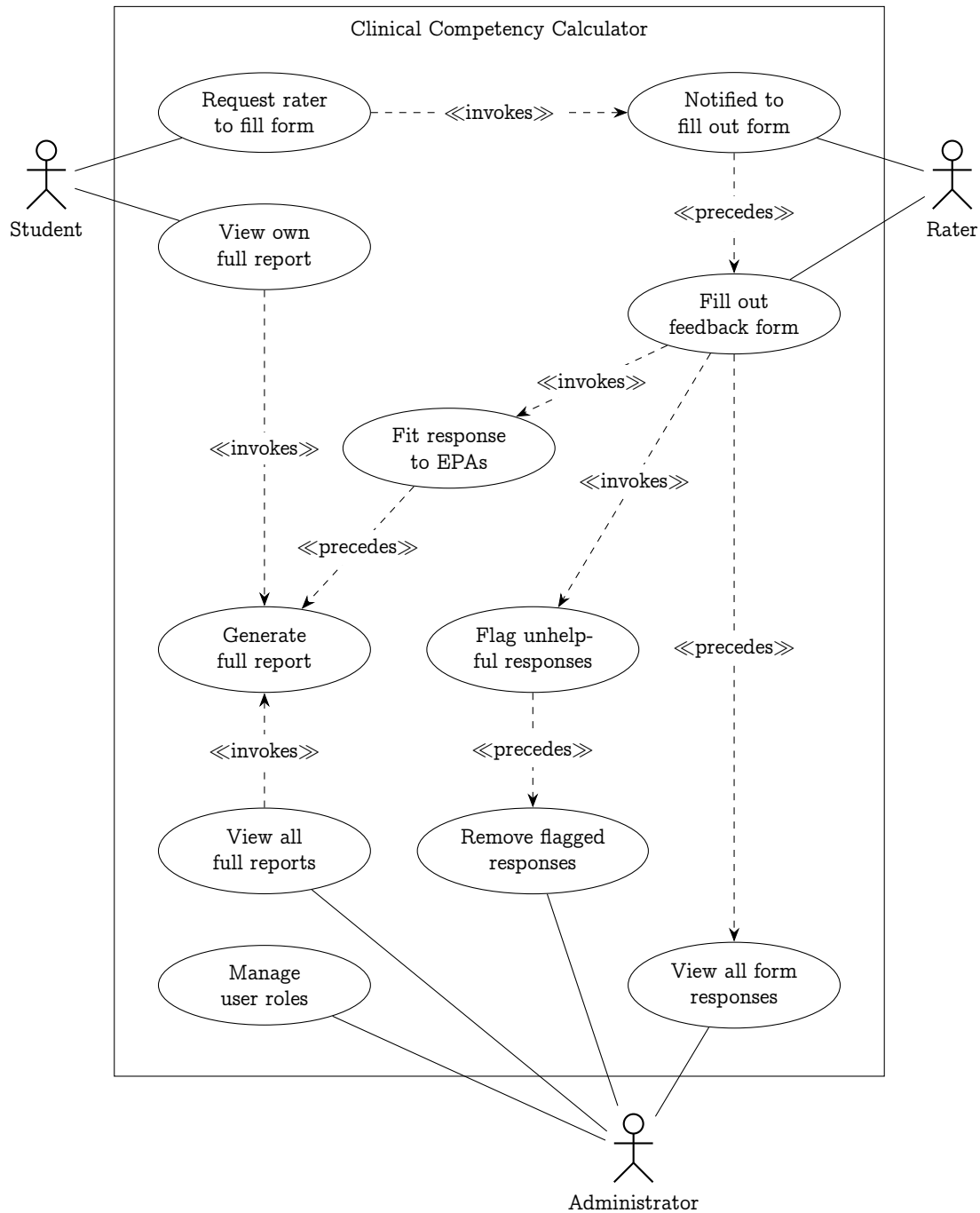


Figure 3.1: Use case diagram for the Clinical Competency Calculator.

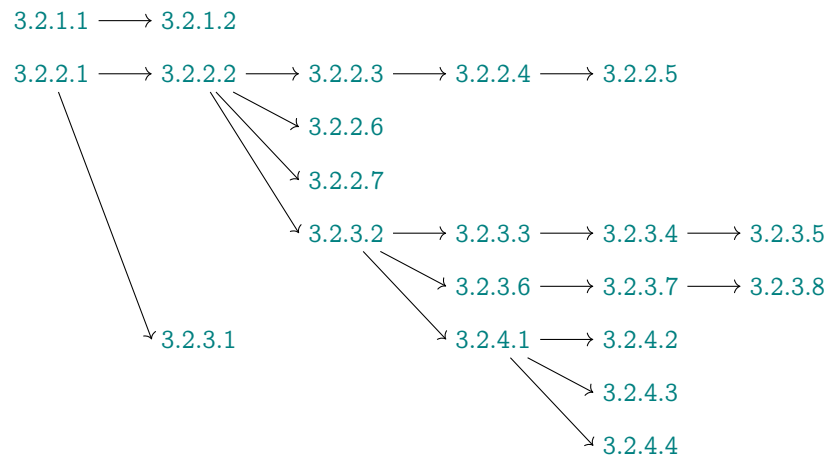


Figure 3.2: *Functional requirements sorted topologically by dependency. Requirements without dependents and/or dependencies are not included.*

- 3.2.2.3. The student **MUST** be able to request a rater to fill out a feedback form.
Importance: ■ HIGH *Dependencies:* 3.2.2.2
- 3.2.2.4. The system **MUST** notify raters when a student requests feedback.
Importance: ■ MEDIUM *Dependencies:* 3.2.2.3
- 3.2.2.5. The system **MUST** remind raters if feedback forms are not completed within a specified time.
Importance: ■ MEDIUM *Dependencies:* 3.2.2.4
- 3.2.2.6. The system **MUST** notify students when a rater has completed a feedback form for them.
Importance: ■ LOW *Dependencies:* 3.2.2.2
- 3.2.2.7. Students **MUST NOT** be able to view the identity of the rater of any form response.
Importance: ■ HIGH *Dependencies:* 3.2.2.2

3.2.3 Form response quality control

The functional requirements in this subsection are related to the auditing and control of the quality of form responses received from raters. During the process of a rater filling in a feedback form, if the rater attempts to submit a poor form response, the system will prompt the rater with suggestions to improve their response. Upon submission, the system will then further audit the form response for quality and flag the rater if necessary.

- 3.2.3.1. The system **MUST** prompt the rater for better feedback based on gaps in form responses received from raters.
Importance: ■ HIGH *Dependencies:* 3.2.2.1
- 3.2.3.2. The system **MUST** store form responses in a database for future access.
Importance: ■ HIGH *Dependencies:* 3.2.2.2

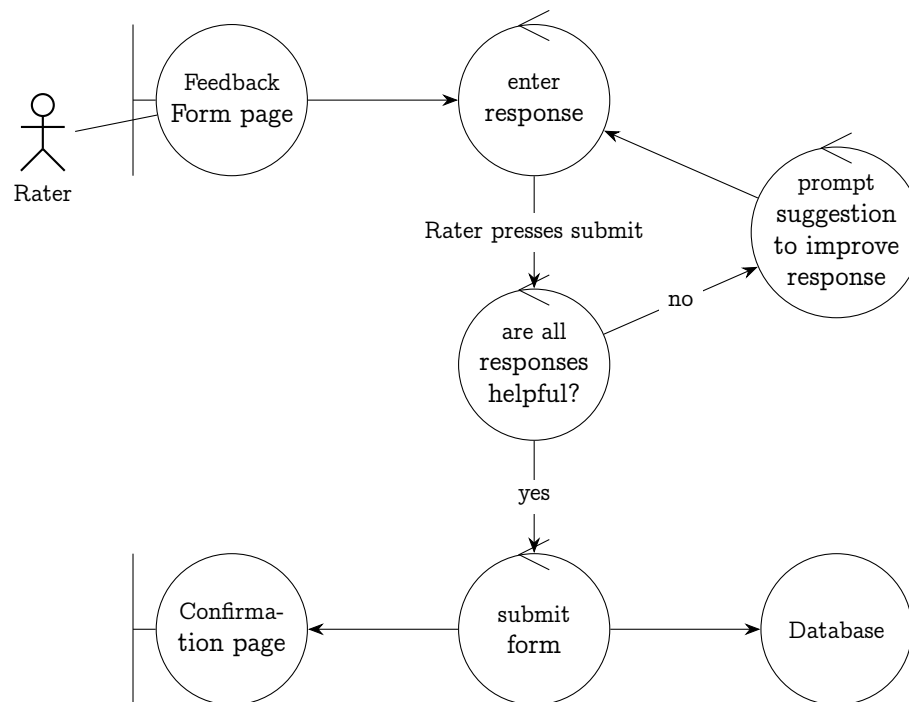


Figure 3.3: Robustness diagram describing the process of a rater filling in a feedback form.

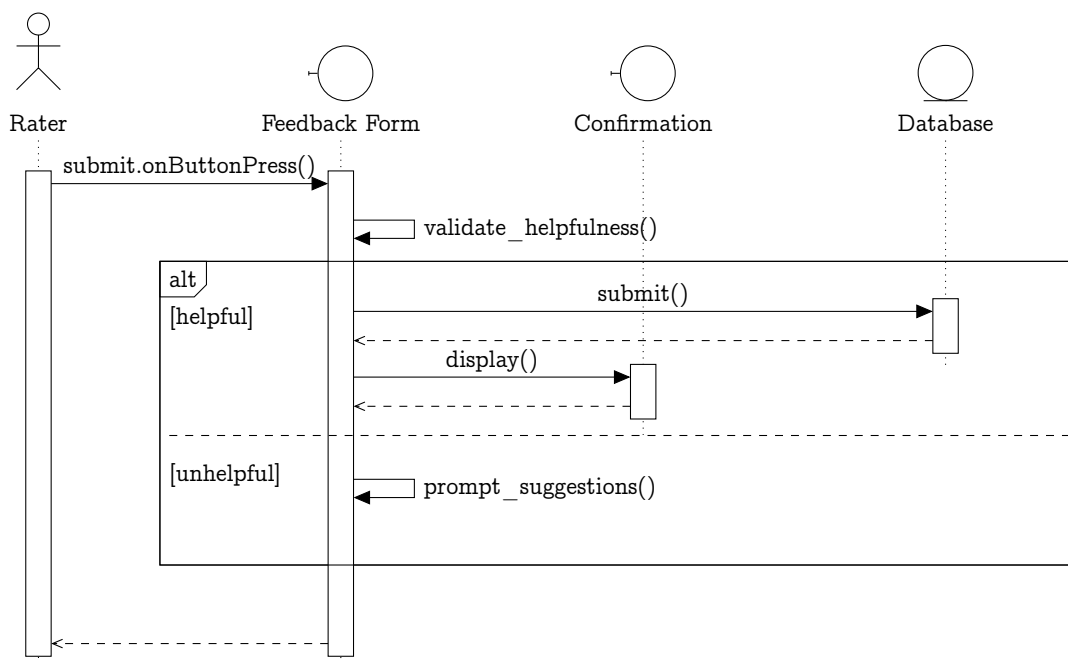


Figure 3.4: Sequence diagram describing the process of a rater filling in a feedback form.

- 3.2.3.3. Administrators **MUST** be able to view all form responses.
Importance: ■ HIGH *Dependencies:* [3.2.3.2](#)
- 3.2.3.4. Administrators **MUST** be able to view who submitted each form response.
Importance: ■ HIGH *Dependencies:* [3.2.3.3](#)
- 3.2.3.5. Administrators **MUST** be able to filter form responses by rater and student.
Importance: ■ HIGH *Dependencies:* [3.2.3.4](#)
- 3.2.3.6. The system **MUST** be able to detect low-quality and unhelpful form responses, flag the corresponding responses and raters, and notify administrators.
Importance: ■ MEDIUM *Dependencies:* [3.2.3.2](#)
- 3.2.3.7. Administrators **MUST** be able to view raters flagged for low-quality or unhelpful form responses.
Importance: ■ MEDIUM *Dependencies:* [3.2.3.6](#)
- 3.2.3.8. Administrators **MUST** be able to filter and remove low-quality or unhelpful form responses.
Importance: ■ MEDIUM *Dependencies:* [3.2.3.7](#)

3.2.4 Full report

- 3.2.4.1. The system **MUST** be able to generate a full report for each specific student.
Importance: ■ HIGH *Dependencies:* [3.2.3.2](#)
- 3.2.4.2. The full report of a student **MUST** describe the level of competency they have been assessed to exhibit in accordance with EPAs.
Importance: ■ HIGH *Dependencies:* [3.2.4.1](#)
- 3.2.4.3. Students **MUST NOT** be able to view any other student's full report.
Importance: ■ HIGH *Dependencies:* [3.2.4.1](#)
- 3.2.4.4. Administrators **MUST** be able to view every student's full report.
Importance: ■ HIGH *Dependencies:* [3.2.4.1](#)

3.2.5 Other functional requirements

- 3.2.5.1. The software **MUST** be able to export data via Portable Document Format (PDF) and comma-separated values (CSV) file formats.
Importance: ■ LOW

3.3 Non-functional requirements

3.3.1 Performance requirements

The system should be able to handle concurrent usage by at least 50 users without significant degradation of performance. The system should be designed to have an uptime of 99%, equivalent to slightly more than 3.5 days of downtime per annum.

3.3.2 Design constraints

The system should be accessible via the latest versions of common web browsers such as Google Chrome, Firefox, Safari, and Microsoft Edge that support the latest version of the Hypertext Markup Language (HTML) standard as promulgated by the Web Hypertext Application Technology Working Group (which can be accessed at <https://html.spec.whatwg.org/>). The system should also be accessible from both mobile devices and desktop environments.

3.3.3 Regulatory compliance

The system must comply with all privacy regulations at Hershey Medical Center and Penn State Health as a company. The system must also comply with all relevant regulations set forth by the US Government.

3.3.4 Backup and recovery

The system should have a backup and recovery plan to protect data integrity in the event of system failure or data loss. Regular automated backups should be scheduled, with clear procedures for data restoration.

3.3.5 User training and documentation

User training and documentation should be provided to ensure that all users, including students, raters, and administrators, can navigate and use the system. This may include user manuals, FAQs, and/or training manuals.

Client signature
Anthony Dambro

Date

Appendix A

Software engineering tools

A.1 Version control

The development team will use Git for version control. The repository will be hosted on GitHub.

A.2 Issue tracking

The development team will use GitHub Issues for issue tracking. Issues will be created for bugs, feature requests, and other tasks.

A.3 Automated testing

The development team will use GitHub Actions for workflow automation. Automated tests will be run on each pull request to ensure code functionality. These tests will be written using a testing framework to ensure scalability and maintainability.