**SRS Document Development and GitHub**

Charlie Farris

University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Dr. Charmelia Butler

November 6, 2022

**Template used for SRS document**:

Srs template-ieee · Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

<https://www.modernanalyst.com/Resources/Templates/tabid/146/ID/497/Karl-Wiegers-Software-Requirements-Specification-SRS-Template.aspx>

**Software Requirements Specification**

**for**

**Student Portal**

**Version 1.1 approved**

**Prepared by Charles Farris**

**Apple****p University**

**November 6, 2022**

**Table of Contents**

**Table of Contents 3**

**Revision History 3**

**1. Introduction 4**

1.1 Purpose 4

1.2 Document Conventions 4

1.3 Intended Audience and Reading Suggestions 4

1.4 Product Scope 4

1.5 References 5

**2. Overall Description 5**

2.1 Product Perspective 5

2.2 Product Functions 5

2.3 User Classes and Characteristics 6

2.4 Operating Environment 6

2.5 Design and Implementation Constraints 6

2.6 User Documentation 7

2.7 Assumptions and Dependencies 7

**3. External Interface Requirements 7**

3.1 User Interfaces 7

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 8

**4. System Features 8**

4.1 New User Registration 8

4.2 System Login 8

4.3 Course Enrollment 9

**5. Other Nonfunctional Requirements 10**

5.1 Performance Requirements 10

5.2 Safety Requirements 10

5.3 Security Requirements 10

5.4 Software Quality Attributes 10

5.5 Business Rules 10

**6. Other Requirements 11**

**Appendix A: Glossary 11**

**Appendix B: Analysis Models 11**

**Appendix C: To Be Determined List 11**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Charles Farris | 11/04/2022 | Initial document | 1.0 |
| Charles Farris | 11/06/2022 | Updated User Classes and Characteristics, Operating Environment, Appendix B: Analysis Models, and Appendix C: To Be Determined List | 1.1 |

1. **Introduction**

The intent of this document is to collect, analyze, and provide an in-depth insight into the complete student portal site by defining the problem statement in detail. The detailed requirements of the student portal site are provided in this document. This will is a new student portal and, as such, will not be replacing or upgrading existing functionality.

**1.1 Purpose**

The purpose of this document is to gather and analyze the ideas that have been presented to define a student portal site and its requirements with respect to the students. In addition, an attempt will be made to anticipate and sort out how this product will be utilized so as to gain a better understanding of the overall project, capture and define concepts for later development, and document any other ideas that are to be considered but may be abandoned later as development proceeds.

* 1. **Document Conventions**

Every requirement statement is to have its own starting priority (HIGH, MEDIUM, LOW).

**1.3 Intended Audience and Reading Suggestions**

The intended audience for this document includes the developers, project manager(s), product manager(s), testers, end-users (students), and the academic staff. Reading suggestions will be provided in a later version.

**1.4 Product Scope**

This document specifies functional and non-functional requirements for the student portal site. The various required functionality that will be implemented on the site will be defined within this document.

**1.5 References**

Functional vs. non functional requirements. (2020, April 28). *GeeksforGeeks*. Retrieved from

<https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>.

Hagh, F. (2020, July 2). *SRS – Software Requirements Specification*. NASA Software

Engineering Handbook (Ver C). Retrieved from <https://swehb.nasa.gov/display/SWEHBVC/SRS+-+Software+Requirements+Specification>.

1. **Overall Description**
   1. **Product Perspective**

This software system will be an online academic portal for the Applepuniversity to manage their academic needs online. The system is to be a simple and intuitive system that will cater to the academic enrollment needs of students. Allowing users to view and manage their enrollment in available courses.

**2.2 Product Functions**

The functions for this product will allow the user to:

* Create a unique account and profile
* Log into the student portal securely once their account is set up
* Allow users to manage their enrollment across the three semesters
  + Enroll in available courses
  + Add to or drop from course waitlists
  + Notify users on waitlists when they are eligible to enroll in the pending course

**2.3 User Classes and Characteristics**

The classes to be used in the product for the student portal functions are as follows:

* Auth()
* Register()
* Login()
* Profiles()
* Enroll()
* Enrollment()
* Security()
* User()

This is not to be considered an exhaustive list of classes that will be utilized, as the need for new classes may arise as the product is developed.

**2.4 Operating Environment**

The student portal site will support and function properly in the following web browsers:

* High Priorities
  + Google Chrome (latest version) — Windows, Mac OS X, iOS, Android
  + Microsoft Edge (latest version) — Windows, Android
  + Mozilla Firefox (latest version) — Windows, Android
  + Apple Safari (latest version) — Mac OS X, iOS
* MEDIUM priorities
  + Microsoft Internet Explorer 11+ — Windows
* LOW priority
  + Other non-mentioned browsers
    - However, the pages may not display as designed or provide the best user experience.

**2.5 Design and Implementation Constraints**

PHP 7.x will be utilized for student portal site functionalities

MariaDB 10.x will be utilized for the database to store the user registration, login, profile, and class enrollment information.

**2.6 User Documentation**

All user documentation, including online help tutorials and user manuals, will be provided as functionality is released.

**2.7 Assumptions and Dependencies**

Not applicable.

1. **External Interface Requirements**

**3.1 User Interfaces**

The user interfaces will consist of a registration page, login page, user home page, user profile page, and class enrollment page. These user interfaces are not necessarily the only interfaces that will be developed and utilized, as the need for new interfaces may arise as the product is developed.

**3.2 Hardware Interfaces**

Not applicable.

**3.3 Software Interfaces**

1. The registration page will communicate with the database to capture initial user information, including login id (email address) and password.
2. The login page will communicate with the database to verify and validate the user prior to providing access to the student portal.
3. The user profile page will communicate with the database to update user information, including but not limited to name, phone, and email.
4. The class enrollment page will communicate with the database to provide a list of available courses and allow users to enroll in, be placed on a waitlist for, or drop courses.

**3.4 Communications Interfaces**

The HTTPS protocol will be utilized to help secure the student portal site. In addition, all communications between the web server and the database will utilize secure protocols.

1. **System Features**

**4.1 New User Registration (HIGH priority)**

4.1.1 Description

The new user registration functionality will provide the user with the means to create an account and profile on the student portal. To accomplish this, the user will be presented with a web page that will prompt them for the required information.

4.1.2 Stimulus/Response Sequences

The student is the user actor in this scenario registering for an account and profile on the student portal site. The user will be presented with a menu on the main home page that contains a Register option.

4.1.3 Functional Requirements

REQ-1: Registration page must require name, mailing address, phone, social security number, email address, and password.

REQ-2: Each user’s identifier (email address) must be unique within the system.

**4.2 System Login (HIGH priority)**

4.2.1 Description

The system login functionality will provide the user with the means to access the student portal after a successful registration. To accomplish this, the user will be presented with a web page that will prompt them for their user ID (email address) and password.

4.2.2 Stimulus/Response Sequences

The student is the user actor in this scenario logging into the student portal site. The user will be presented with a menu on the main home page that contains a login option.

4.2.3 Functional Requirements

REQ-1: The login page must prompt for the user ID and password.

REQ-2: User ID and password combination must be verified and validated against the database.

**4.3 Course Enrollment (HIGH priority)**

4.3.1 Description

The course enrollment functionality will provide the user with the means to view and enroll in available courses on the student portal. To accomplish this, the user will be presented with a web page that will prompt them for the semester they wish to enroll in, then a list of available courses for that semester. If a class that a student wishes to enroll in is full, they will be prompted to be added to the waitlist.

4.3.2 Stimulus/Response Sequences

The student is the user actor in this scenario enrolling in courses on the student portal site. The user will be presented with a menu on the user’s home page that contains an Enroll option.

4.3.3 Functional Requirements

REQ-1: The enrollment page must prompt for the semester (spring/summer/fall), then prompt for available courses for that semester.

REQ-2: Users must have the ability to add themselves to a waitlist if the course maximum enrollment has been reached.

REQ-3: The user must have the ability to drop a class they previously enrolled in.

REQ-4: Each course shall have a maximum number of enrollments allowed. This maximum may be different from course to course.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The student portal functionality requires standard commits to the database.

**5.2 Safety Requirements**

Not applicable.

**5.3 Security Requirements**

The site must be secured utilizing HTTPS protocol, and all passwords and social security numbers must be encrypted in the database.

**5.4 Software Quality Attributes**

1. Usability: Any registered user should be able to log into the system and enroll in courses.

2. Availability: The student portal site should be available 24/7 except during scheduled maintenance hours. To this end, the student portal must have a 3 nines uptime upon initial launch and a 4 nines uptime after the first year.

Kusnetzky, D. (2015

**5.5 Business Rules**

TBD

1. **Other Requirements**

TBD

**Appendix A: Glossary**

HTTPS: Hypertext Transfer Protocol Secure

ERD: Entity Relationship Diagram

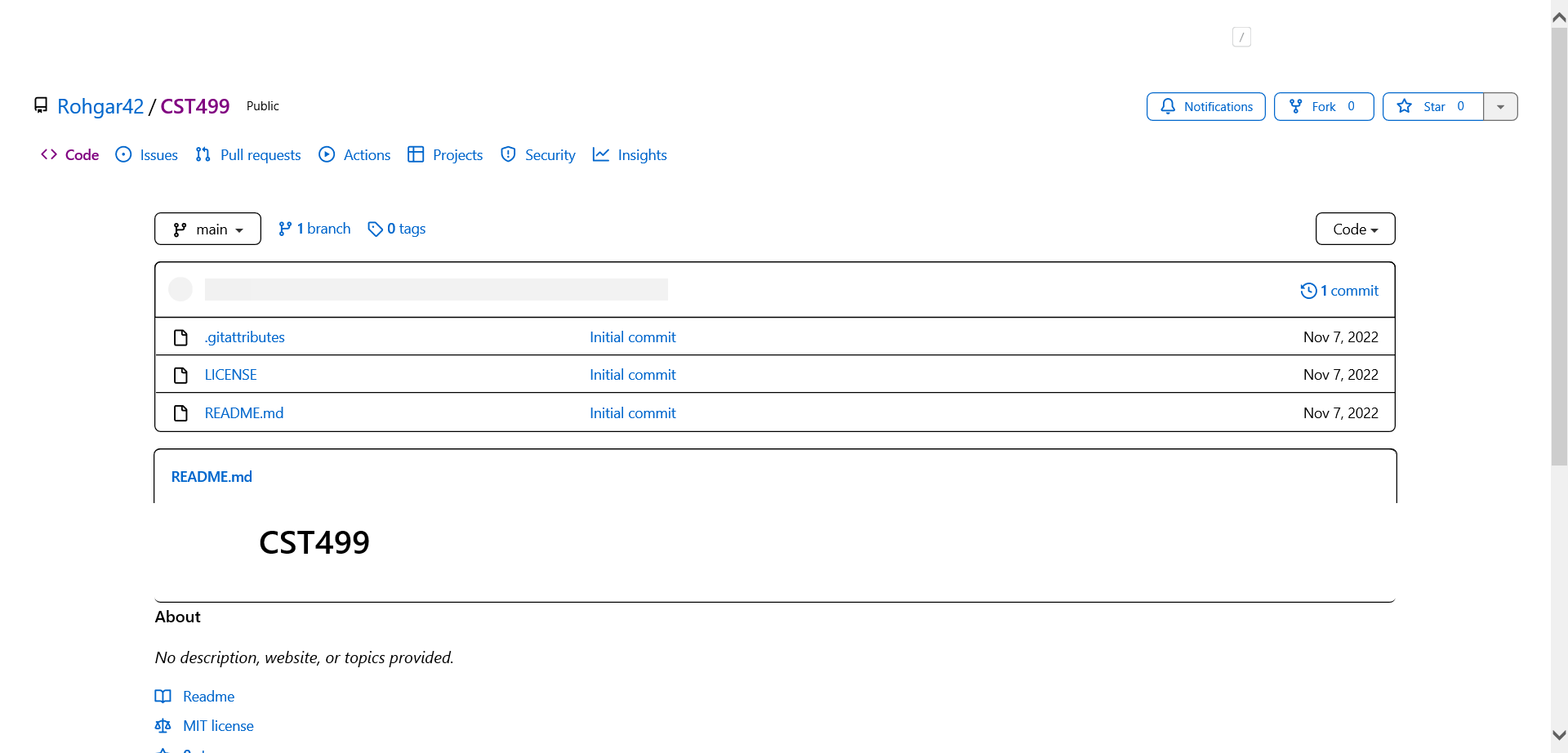
**Appendix B: Analysis Models**

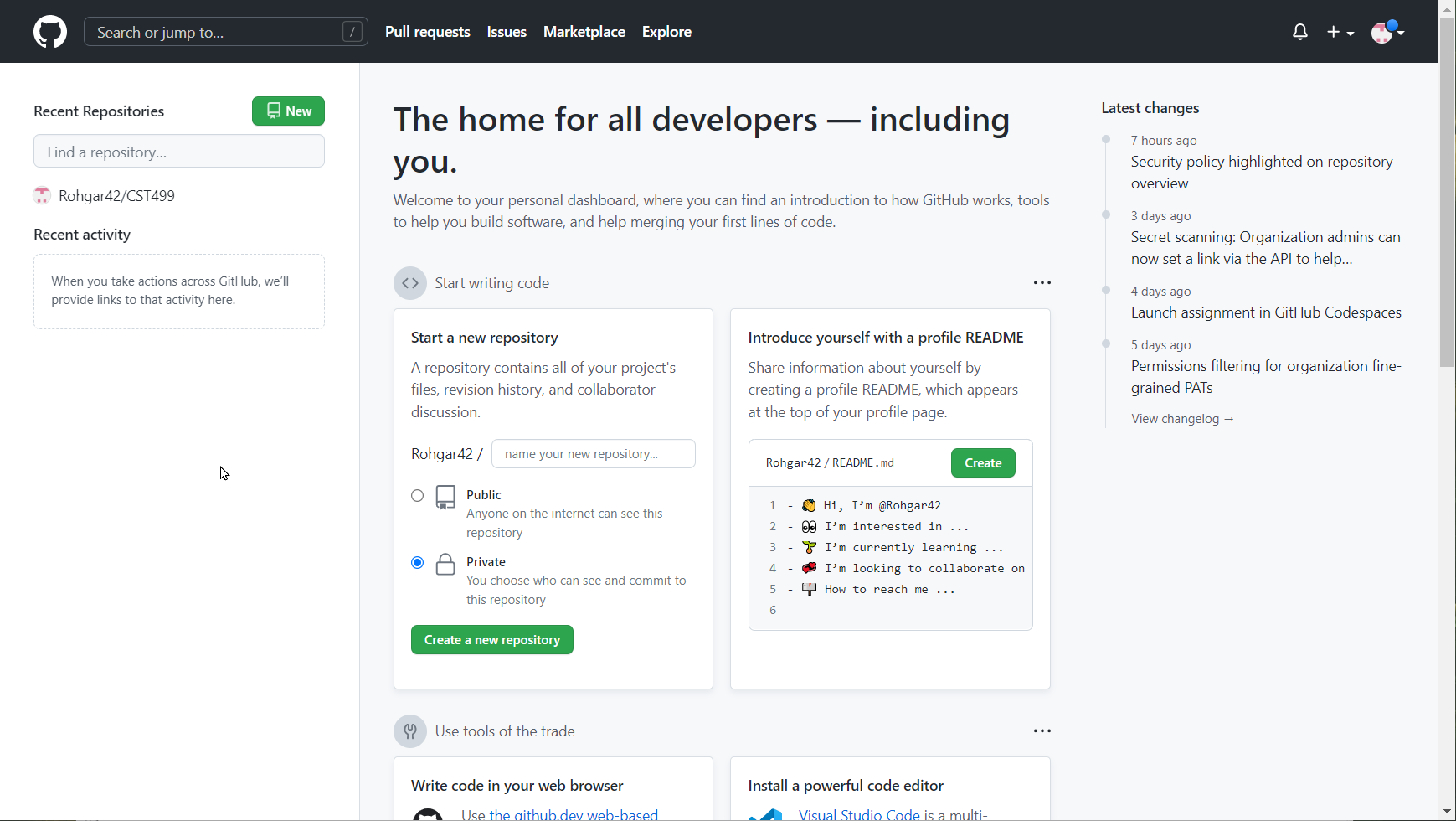
See UML Design Models document.

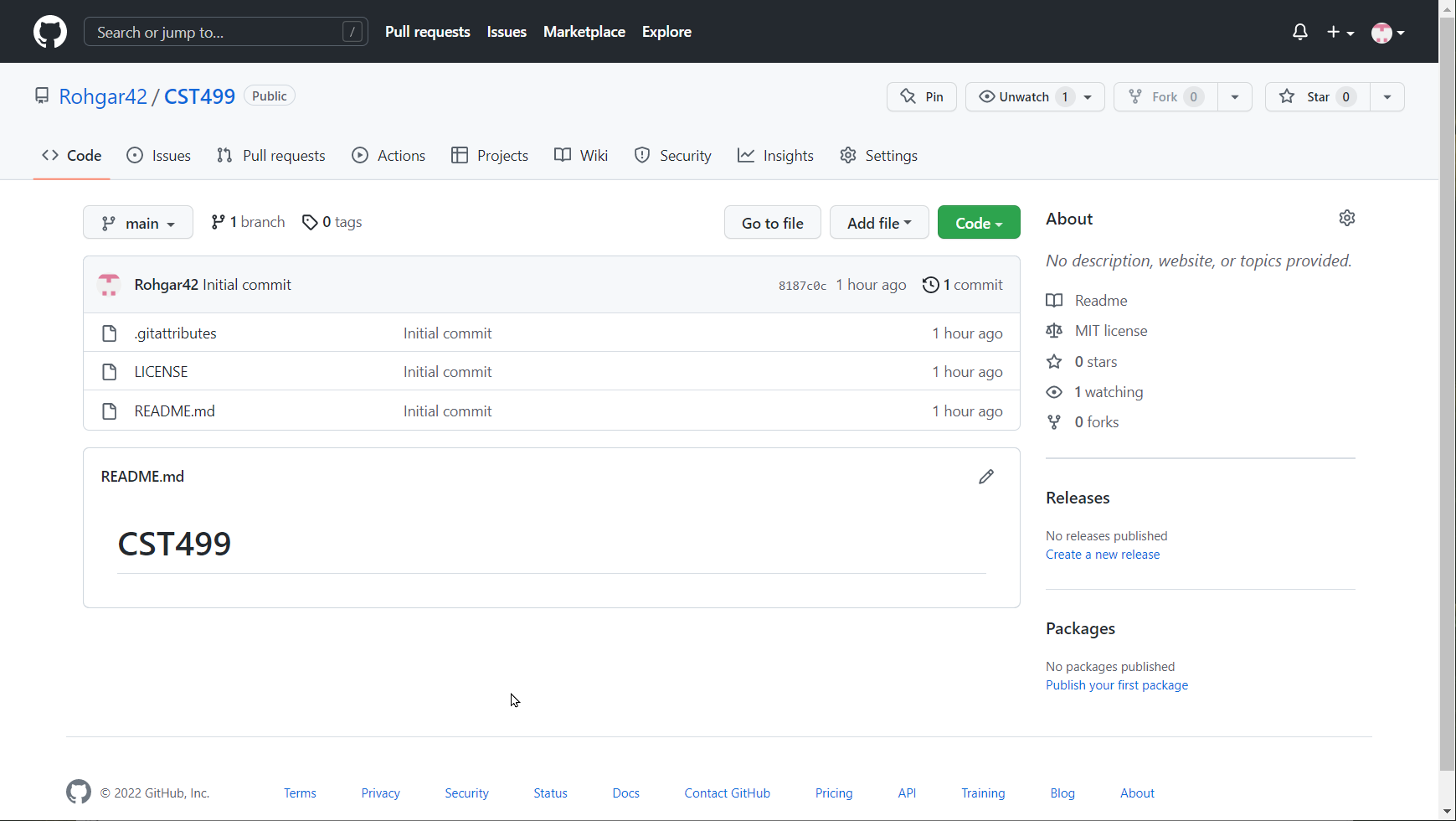
**Appendix C: To Be Determined List**

1. Business Rules: Still collecting information.
2. Other Requirements: Still collecting information.

**GitHub Screenshots**

[**https://github.com/Rohgar42/CST499**](https://github.com/Rohgar42/CST499) ****

****

****

**References**

Functional vs non functional requirements. (2020, April 28). GeeksforGeeks. Retrieved from [https://www.geeksforgeeks.org/functional-vs-non-functional-requirements](https://www.geeksforgeeks.org/functional-vs-non-functional-requirements%20/)

Hagh, F. (2020, July 2). SRS – Software Requirements Specification. NASA Software Engineering Handbook (Ver C). Retrieved from <https://swehb.nasa.gov/display/SWEHBVC/SRS+-Software+Requirements+Specification>

Karl , W. (1999). *Karl wiegers’ software requirements specification (Srs) template*. <https://www.modernanalyst.com/Resources/Templates/tabid/146/ID/497/Karl-Wiegers-Software-Requirements-Specification-SRS-Template.aspx>

Kusnetzky, B. D. & 06/01/2015. (n.d.). *How many nines of uptime do you need? -*. Virtualization Review. Retrieved November 6, 2022, from <https://virtualizationreview.com/articles/2015/06/01/how-many-nines-of-uptime-do-you-need.aspx>