**Software Requirements Specification**

**for**

Blackbear Square

**Version 0.01 approved**

**Prepared by,**

**Team Egg**

**9/29/2022**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Troy Schotter | 9/29 | Initial creation of template document | 0.0.1 |
| Brennan Poitras | 9/30 | Creation of sections 1.1, 1.2, 1.3, 1.4, 2.3 |  |

# **Introduction**

## **Purpose [Bren**nan]

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

This document is a requirements specification for the first release of BlackBear Square. The document contains specifications for various systems including, but not limited to, the profile feature, course review feature, and the course search/filter system.

## **Document Conventions [Brennan]**

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

Sub-requirements of each system will have their own priority level and will not inherit the priority level of the overall system which will allow more flexibility when it comes to implementing features that are desired, but not essential to system functionality.

## **Intended Audience and Reading Suggestions [Brennan]**

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

This document is intended for the product manager, developers, designer, and product users. The rest of this document contains a more detailed description of the product and its functions, other software components that this product will interface with, network communication protocols, major product features, and other non-functional requirements.

Users of this product should read sections 1 and 2 of this document to get a general understanding of the product. They may read the more detailed specifications if they desire.

The product manager, developers, and designer should read all sections but mostly focus on sections 3, 4, and 5 to get a better understanding of the features that will be implemented in the product.

## **Product Scope [Brennan]**

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

BlackBear Square is a course review software that allows students to provide feedback on the courses they have taken so other students are able to make a more informed decision regarding class selection. The goal of this product is to increase student productivity and course retention rate at the University of Maine.

From a business perspective, both the University and students will benefit from this software by increasing student satisfaction and course retention rates. These factors will increase overall student retainment which will ultimately increase University profits.

## References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

<https://www.ratemyprofessors.com/>

# **Overall Description**

## **Product Perspective [Brennan]**

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

BlackBear Square is a new, self-contained product that is intended to be an improvement over current course review software such as Rate My Professor or the University of Maine course review system. As a group, we believe that these two products do not sufficiently allow students to express their thoughts about previous courses.

## **Product Functions[Graham]**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

* The system must allow users to create a registered account using their student email
* The system must allow users to search for their college and courses that college offers
* The system must allow users to create reviews for college courses
* The system must allow users to view posted reviews
* The system must allow users to like other reviews
* The system must allow both guest and signed in users to edit their own reviews

## **User Classes and Characteristics [Brennan]**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

The largest user class for this product is students enrolled in university courses. These users are expected to be the most frequent users of the software and therefore take the highest priority when considering product requirements. Students will be able to use most, if not all, system features including profile creation and access, course review creation, and the ability to read and respond to reviews of other users.

System administrators and developers are also a high priority user class but will have a separate set of requirements because these users are not concerned with reading and writing course reviews. This class will have higher security and privilege levels which will allow them to modify the content of the website and access system logs/data.

Presumably, professors will use the software to read reviews of their course. However, they will not have permission to write or respond to reviews. This is the lowest priority user class.

## **Operating Environment**

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

Blackbear Square is designed to work on phone and web browsers. It will run on 90% of browser supporting devices on the market

## **Design and Implementation Constraints [Owen**]

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

We have a few things to consider given that the functionality of our website heavily relies on user input and saving their information into a database. Security of our database as well as our system as a whole is highly valued, so developing security breach policies will be highly prioritized. Our communication protocols limit our communication to http and https. The University of Maine will be in charge of maintaining it. We also intend on developing two factor authentication through email for students who sign up.

## **User Documentation [Owen]**

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

Black Bear Square will plan on having an optional tutorial system accessible through a question mark with a closable drop down suggesting a tutorial. If selected it will present bubbles in a non obscuring fashion with a closable “x” in the corner. There will also be an accessible help option available on all pages at the bottom which will bring the user to a user support email as well as complaint or question page.

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

The application must run on a server

# **External Interface Requirements**

## **User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

<https://miro.com/welcomeonboard/U1c0dEFucXlJRlpsZWhKcHVSb1VoWlhNWlBvTUNPaFR2cVNKRmlIWkRIWkk0RjNhbDNUZEM0NjY3MHN3WGcyTXwzNDU4NzY0NTM0MDE3MzgyNDkwfDI=?share_link_id=912182037519>

## **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces [**Troy]

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

Server will need its own email account on google to send out verification emails to users. Verification email account will be a “no reply” account, it will automatically trash any email sent to it. App will need another email for official requests from users. Server will broadcast on both HTTP and HTTPS, redirecting HTTP traffic to HTTPS.

Database will be hosted on the same machine (for now), with the web server communicating to it. For security purposes the database will be hosted on different servers in the future.

When storing a review in the database associate the review with user id and course id. Anonymous reviews will be tied to a dummy user account not associated with any living person.

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## Review Posting

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.1.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

High priority

4.1.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1: The course review system shall allow a user to post a course review.

REQ-2: The course review system shall allow a user to delete one of their course reviews

REQ-3: The course review system shall allow a user to respond to a course review written by another user.

REQ-4: The course review system shall allow a user to remain anonymous when posting a course review.

REQ-5: The course review system shall allow a user to revise one of their previously written course reviews.

## Personal Profile[Brennan]

4.2.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

The profile feature will allow users to access and customize a personal profile. The profile will have information including, but not limited to, a profile picture, some personal information, and a list of past reviews written by the user. This is a high priority feature.

4.2.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

-User clicks create account -> System prompts user for email and password and stores in database

-User clicks sign in -> System prompts user for email and password

-User clicks sign out -> System signs out user and returns to the default screen

-User clicks reset password -> System prompts user for old password and new password

-User clicks any personal information field -> System allows user to enter information and stores it in the database

-User is currently on their profile page -> System will show a list of their previous reviews

4.2.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1: The profile system shall allow the user to create an account

REQ-2: The profile system shall allow the user to sign in to their account

REQ-3: The profile shall provide an error message if the user enters an invalid email

REQ-4: The profile system shall provide an error message if the user enters an invalid password

REQ-5: The profile system shall allow the user to sign out of their account

REQ-6: The profile system shall allow the user to reset their password

REQ-7: The profile system shall allow the user to upload a profile picture

REQ-8: The profile system shall allow the user to add their personal information

REQ-9: The profile system shall display all previous reviews written by the user

REQ-10: The profile system shall allow the user to delete a previously written review

# **Other Nonfunctional Requirements**

## **Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*REQ-1: The system shall be able to handle a standard workload of 1000 users at all times, 95% of the time.

REQ-2: The system shall have a response time of 2 seconds back to the user when querying the database, 95% of the time.

## **Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

REQ-1: The company shall send a security message email to all registered users within 24 hours of discovering a security breach on the server.

## **Security Requirements [**Troy]

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

REQ-1: Server will use a SSL certificate provided by LetsEncrypt.  
REQ-2: The web app server will block the account from being logged in for 10 minutes by the requesting IP address after 5 failed attempts to log into an account.

REQ-3: The web app server will block the account from being logged in for 1 hour by the requesting IP address after failing to log into an account that has been affected by Security REQ-2 in the past hour by the same IP address.

REQ-4: The system shall not store user passwords in plain-text on the database at all.

REQ-5: The remote web server shall not accept any SSH or FTP connection without private/public authentication.

REQ-6: The server shall send a registration authentication email when a user signs up with a valid .edu email address that was previously unregistered within 1 hour.

REQ-7: The server shall deny registration confirmations if the link was sent out more than 2 days ago.

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## **Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*