

Software Requirements Specification

for

Would You Rather?

Version 1.0

Prepared by Tracy Hotchkiss

Group Name: Team 21

|  |  |  |
| --- | --- | --- |
| Tracy Hotchkiss | 011619865 | tracy.hotchkiss@wsu.edu |
| Kelly St.Onge | 011376758 | kelly.stonge@wsu.edu |
| Brian Hong | 011708122 | wenle.hong@wsu.edu |
| Lane Koistinen | 011707400 | lane.koistinen@wsu.edu |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
| Date: | <11/06/2020> |
|  |  |
|  |  |
|  |  |

Content

Revisions iii

1 Introduction 1

1.1 Document Purpose 2

1.2 Product Scope 2

1.3 Intended Audience and Document Overview 3

1.4 Definitions, Acronyms and Abbreviations 3

1.5 Document Conventions 4

1.6 References and Acknowledgments 4

2 Overall Description 4

2.1 Product Perspective 4

2.2 Product Functionality 4

2.3 Users and Characteristics 4

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 6

2.7 Assumptions and Dependencies 6

3 Specific Requirements 6

3.1 External Interface Requirements 6

3.2 Functional Requirements 7

3.3 Behavior Requirements 9

4 Other Non-functional Requirements 11

4.1 Performance Requirements 11

4.2 Safety and Security Requirements 11

4.3 Software Quality Attributes 11

Appendix A – Data Dictionary 13

Appendix B - Group Log 14

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.1 | Team 21 | Initial start template. | 11/04/2020 |

# Introduction

*“Would You Rather” Is a conversation party game where players take turns posing a dilemma in the form of a question beginning with “would you rather”.1 The dilemma, being a problem with two possibilities, neither of which is acceptable or preferable. Team 21 wishes to create an application for people wanting randomized generated questions. The overall goals of the system are:*

*1.    Provide access to an easy to use website of “Would you rather” questions that will be randomly generated at the push of a button.*

*2.    Allow users to respond to the “Would you rather” questions and provide fun user stats based on user responses.*

*3.    Users will be able to view the stats of all the questions in the database.*

*This document is intended to provide the purpose and requirements for the proposed application meant for key system stakeholders.*

## Document Purpose

This document Would You Rather? Version 1.0 provides the requirements for the "Would You Rather" or WYD web application. This entails the user interface and software requirements. Additionally, the intended audience and purpose for this product will be described in this document. Product parameters, goals and intended functionality will be defined to provide key stakeholders information about how the product will work.

## Product Scope

“Would You Rather” is an online service that asks users hypothetical “either-or” questions as a thought experiment and reveals the statistical values of the question ie. percentages of which choice was picked per question. This statistic is shown after the user selects a choice.. The purpose of the product is to provide conversational entertainment by offering users WYR questions at the touch of a button.

People love choice, they also love instant results. Given a simple would you rather question allows them to weigh in on what they would do consequence-free. Although there are no incorrect answers to a would you rather question, it’s interesting to see what a large population would choose given a binary choice. The benefits of this product would give the user immediate feedback to the statistics of what they and others have chosen. It would also allow users to ask their own questions that they have thought of for others to choose over. The objective of the page needs to be minimalistic in feel, with responsive interaction with the question boxes. The questions need to be applicable to most people. For instance, If we ask, “Would you rather; program in assembly or walk on Legos?” Isn’t going to achieve the results we want because most people have not programmed in assembly. However, most people understand that walking on a Lego is painful. A better question might be, “Would you rather keep repeating high school or always walk on Legos?” With better questions, we can deliver a better, more engaging product for the end-user. The goal of the product is to create a working page that is aesthetically pleasing but gives the user relevant feedback to the question asked.

## Intended Audience and Document Overview

This document was drafted as an overview for the *Would You Rather?* website project. Each section reviews a different aspect of the website specifications and functional requirements for review by the software engineering course professor for whom the project is being completed, as well as daily website users. The agreements and goals for the website development process are fully documented within this guide.

The primary content is divided into four main sections. All audiences should reference the first section prior to their reading, which provides a summary of this SRS Document and the project scope.

Sections 2-4 cover the product description, the project-specific requirements, and other related requirements, respectively. Sections 3 and 4, which primarily cover development and software engineering topics should be of particular interest to the course professor, while the product overview in section 2 is most relevant to the client.

## Definitions, Acronyms and Abbreviations

WYR - Would You Rather?

    Title of the website project.

## Document Conventions

In general, this document follows the IEEE formatting requirements. Arial font size 12 is used throughout the document. Italics used for comments. Document text is single spaced and maintains 1” margins.

## References and Acknowledgments

“Would you rather,” *Wikipedia*, 15-Sep-2020. [Online]. Available: https://en.wikipedia.org/wiki/Would\_you\_rather. [Accessed: 09-Oct-2020].

# Overall Description

## Product Perspective

This system will contain one main part, the web portal. The web portal will provide the user with a GUI to play "Would You Rather?" and view the statistics for various questions. In order for this system to work, there needs to be a way to store a list of questions and the statistics for each question. This is done by a database. The web portal will communicate with the database to retrieve questions to ask the user. It will also update the statistics in the database every time a user answers a question. Additionally, every time the user accesses the statistics page, the web portal will need to retrieve data from the database to display to the user.

## 

## Product Functionality

1. The user will be able to answer, “Would You Rather?” questions through a user interface
2. The system will store user responses as statistics of each question.
3. The system will display how all users answered a given question after the user answers a question as a percentage.
4. The user will be able to view statistics for all questions on the statistics page.

## Users and Characteristics

Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

The main users of this product will be college aged students ages 18-24, these students need only a basic knowledge of a web browser to access the site. Although not required, a user in this range might find their experience more enjoyable with some experience with pop culture, music, sports, and other relative life experience.

The next deviation might range from 16-55 year of age. Minimal browser knowledge is needed to access the product. This range is bigger than the main group because anyone with internet access can access the site. For the minimum age it may not be appropriate for teens whereas the relevance to the upper end of the spectrum may not find the questions relevant without straying from the main target market.

## Operating Environment

It is important for the site to be compatible with most browsers, the big three are Google’s Chrome, Firefox, Apple’s Safari. Microsoft’s Edge/Explorer should be included as well. This will ensure that the bulk of the market share of people who use different browsers will have access to the features of the product.

For implementation of the site and product the minimum requirements for operation would be GitHub’s minimum requirements for accessing the repository. From the repository the commits of the project are then hosted by a web host which will then host the site. Next, users will interact with the site which will access a database which can be hosted by the same host or another service of the sort.



## Design and Implementation Constraints

There are multiple constraints we have identified in the research stage of the development process, some of which have impacted the overall design of the project. The constraints we have identified include:

1. Inability to provide continual review of user-created submissions to ensure the quality and language of the content is consistent with the standards of the website app.
2. Time does not permit more advanced features, including sharing questions/program output with external applications.
3. Significant difficulties with the logistics of storing user-created submissions in a complex database outside of the user’s own machine.

After examining the impact these constraints have on our ability to develop various features, we have come to several conclusions. We shall keep the app to a simple client-based website, where the primary input for the question ranking shall be specific to each individual user. This way, the question ranking will be personal to the user and not representative of the entire user population.

Additionally, we have decided to place most focus for the project on curated questions selected by the project developers. This ensures the quality of each question and limits most security and quality concerns that may be associated with alternative approaches.

## User Documentation

The primary documentation to assist new users will be a website about-page detailing the functionality basics and developer accreditation for the Would You Rather? project. Ideally, it is unnecessary for new users to reference the about-page due to the emphasis placed on ease-of-use in the UX design effort. The about-page content details the process of submitting a question and explains how the rank of each question dynamically changes with user input. Developer accreditation will be limited to the institution of development (WSUV) and developer names. This page will be easily accessible from the main page of the website.

## Assumptions and Dependencies

For the WYR website to function properly, some minimum system requirements and user privacy expectations must be satisfied. The following list provides a brief overview of these requirements.

1. A link to the statistics page needs to be available for users to view, so users must be okay with basic usage activity being recorded and displayed. The stats on this page are dependent on the questions that will be generated.
2. Project Developers with appropriate programming skills will need to be available.
3. Users do not have internet restrictions or website filters while playing the game. Effort will not be put into avoiding such filters.
4. Users must have a stable internet connection.
5. User must be able to navigate a browser
6. Browser must support Javascript; see details in section 2.4, “Operation Environment”

# Specific Requirements

## External Interface Requirements

### User Interfaces

The users will be presented with a main page which will have an option to go to an about page, an option to go to the statistics page, and an option to generate a random WYR question. When clicking the about option, the user will be brought to a page with relevant information. From the about page interface the user will have the option to go back to the main page.

From the main page when clicking the random question, the interface will now show a randomly generated WYR question and then be given the choice to pick one of the two available options. If the user clicks on either option, the statistics for that question will show under the question. For example, what percentage of people pick one option over another option. There will now also be an option to visit the statistics page.

When users visit the statistics page from the main page, they will be presented with the stats for all the available random questions. There will also be an option to go to the main page, generate a random question, or go to the about page

### Hardware Interfaces

The application will need to recognize mouse clicks to access links and navigate the application. Supported device types would include tablets, pcs, and phones both with mouse or touch screen enabled. However, mobile devices will run the desktop version of the website, as a specific version for mobile will not be developed.

Additionally, communication with hardware devices, such as touch-screen responses and mouse-click responses, will use default communication protocols and handling. This ensures maximum compatibility with users’ devices and therefore improves the user experience.

### Software Interfaces

This system will use a database to store and retrieve data. database will store the list of possible questions and the user responses to each question. The system will retrieve questions from the database to ask the user and retrieve cumulative user responses to present statistics to the user. The system will run using JavaScript. It shall run on both Windows and Linux.

### Communications Interfaces

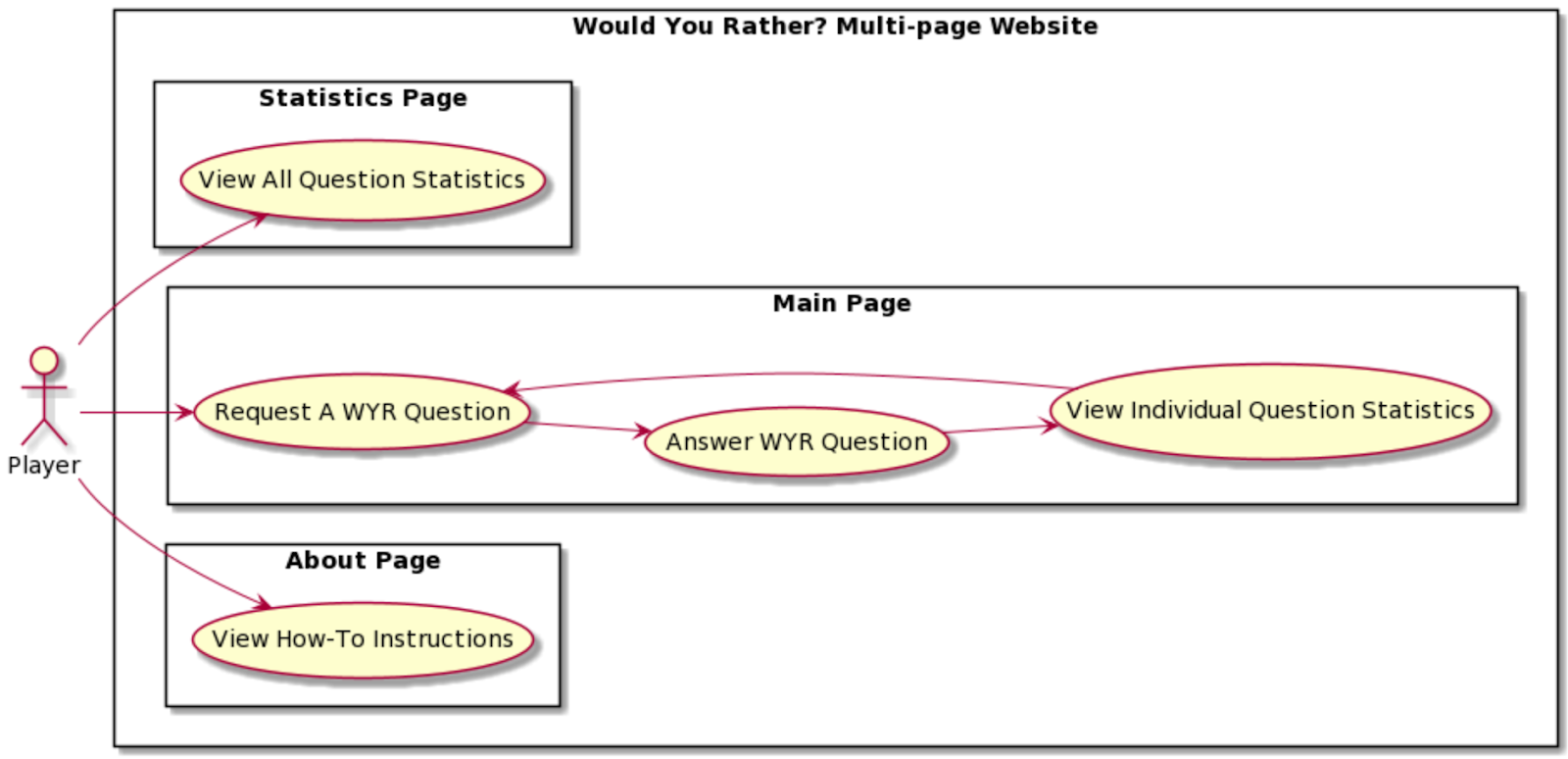
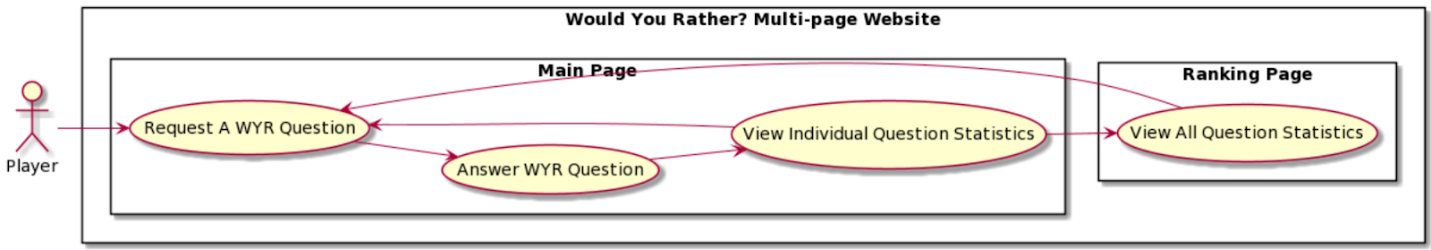
The program will run on Google Chrome, Apple Safari, Mozilla Firefox and Microsoft Edge. Communication between the user and the system will occur through a graphical user interface. This system shall use HTTP protocol for communication over the internet.

## Functional Requirements

* The user will be able to answer the “Would You Rather?” questions through a user interface
  + The system will pull a random question from the data
  + The interface will have two selections for the user to choose from, they will need to click a choice to affect the statistics of a question.
  + For aesthetics, the selection should signify a possible choice giving the user a type of interactive feedback. For instance, a red and blue button would render a darker shade when hovering over the choice. This gives the page a responsive feel.
  + The interface should also have a last and next button that will allow questions to be skipped if the user does not find the question relevant to them, or if they want to go back to a question previously asked. If they have already answered the question, they cannot skew the statistics by repeatedly answering.
* The system will store user responses as statistics of each question.
  + With each question, if given an answer, the database will track different statistics for that choice. For instance, WYR want a red car or a blue car? If the user chooses blue the tally for blue goes up and the tally for the times that question has been answered will go up as well.
* The system will display how all users answered a given question after the user answers a question as a percentage.
  + Displaying the statistics for the user will give them immediate feedback to their choice and the choices of others. For instance, if three people chose a red car and 1 chose blue, 75% would be displayed over the choice for red and 25% would show for blue.
* The user will be able to view statistics for all questions on the statistics page.
  + This page will show the results and specific statistics of each question. For instance, if a certain question didn't have as many answers yet and you wanted to go back and see its results after a period of time, you could go to the stats page and see how it’s doing.
  + For new statistics the user would need to refresh the page.
* Data retrieval
  + Pull questions from the database for the user.
  + Populate statistics for each question and statistics page

## Behavior Requirements

### Use Case View

**Use-Case Diagram Details**

**Actors**

*Player* - The player is an anonymous online user of the WYR website.

**Use-Cases**

*Request A WYR Question -* Start playing by requesting next question

*Answer WYR Questions -* Play the WYR game by answering each question.

*View Single Individual Question Statistics -* See the statistics (most liked/disliked) for an individual question after submitting your answer.

*View All Questions Statistics -* See the statistics (most liked/disliked) for all questions in the game.

*View How-To Instructions -* See basic instructions on how the WYR game works and details on the website creation.

**Systems**

*Would You Rather? Website -* The website running the WYR game.

*Main Page -* Runs the WYR game play by providing a series of questions.

*Statistics Page -* Dedicated to displaying like/dislike statistics for all questions in game.

*About Page -* Provides information on website usage and basic project accreditation.

# Other Non-functional Requirements

## Performance Requirements

For a client with stable internet, all web pages will take no longer than 2 seconds to fully load. The system is simple and should not require unnecessary loading. Updates to pages should take no longer than 2 seconds. For a simple input, the system needs to send data to the database, the database needs to update its data and then the system retrieves data from the database. All of this should take no longer than 2 seconds.

## Safety and Security Requirements

A user of a *Would You Rather* game has the right to expect data collection to be limited within the scope of game answers. This expectation is satisfied in the functional and security requirements we define for the project. We feel that external safety certification is necessary for a game of this nature.

Statistics management is a critical aspect of the security of the WYR website project. All statistics recorded will not be associated with the user who contributed the data and user profiles will not be stored. Because user profiles are not within the defined functional requirements, we will not attempt to manage this data. This prevents more severe security vulnerabilities from ever becoming possible. With this in mind, we define our security requirements as

1. Sensitive user data will not be managed or stored within the scope of this system.
2. Accordingly, all data collected will be anonymous and hosted in a location dedicated to secure data management.
3. Questions will be selected and generated in advance to avoid content containing sensitive information.

The functional requirements for the WYR website project were defined with security in mind. Data is intentionally anonymized, and all content is provided by the website owner.

## Software Quality Attributes

**Correctness:**

The correctness of our system, given the specifications, are to be operational with each piece of the system. The code needs to work not just independently but needs to be cohesive such that the final product is operational. Starting with a prototype and then scaling the features will allow each part of the system to be written, tested, and refactored so that it can maintain correctness.

**Reliability:**

The WYR system has minimal inputs for the user to interact with. Assuming that the hardware or hosting cannot be at fault, with this frequency, a higher rate of reliability and marginal error is achieved.

**Adequacy:**

The user input should be minimal where only necessary functions of the WYR system are extended to them. For instance, if someone doesn’t like a question, they cannot go into the WYR system and effectively delete the said question.

**Learnability:**

The WYR systems interface should be quite easy to learn. It is not complex enough where it needs a tutorial but needs to look and feel interactive for the user to make a decision. Specific design protocols should allow ease of use and be seamless for the user to easily understand the WYR system.

**Robustness & Maintainability:**

The WYR system like many other programs will have an ongoing method of writing, testing, and refactoring. This will ensure that the robustness and maintainability of the system are free from any occurring errors that users find or that the development team tests for.

**Readability:**

Given the specifications of the system agreed upon by the development team, the formatting of code should look uniform and match a certain style. In IntelliJ IDEA, the code can be formatted such that compiling errors will occur if the code is not in a certain style.

**Portability**:

The WYR system should be portable over most browsers in order to reach the majority market share of internet users. For minimum requirements, the development team should agree on a browser of choice for the WYR system then scale as necessary.

.

Appendix A – Data Dictionary

***Actors***

*Player - The player is an anonymous online user of the WYR website.*

***Use-Cases***

*Request A WYR Question - Start playing by requesting next question*

*Answer WYR Questions - Play the WYR game by answering each question.*

*View Single Individual Question Statistics - See the statistics (most liked/disliked) for an individual question after submitting your answer.*

*View All Questions Statistics - See the statistics (most liked/disliked) for all questions in the game.*

*View How-To Instructions - See basic instructions on how the WYR game works and details on the website creation.*

***Systems***

*Would You Rather? Website - The website running the WYR game.*

*Main Page - Runs the WYR game play by providing a series of questions.*

*Statistics Page - Dedicated to displaying like/dislike statistics for all questions in game.*

*About Page - Provides information on website usage and basic project accreditation.*

Appendix B - Group Log

Meeting Minutes

--Meeting 10/6/2020

Team Present:

Tracy Hotchkiss

    Lane Koistinen

    Kelly St.Onge

    Brian Hong

Assign Team Lead - Tracy Hotchkiss

Project Web App Ideas

* Pachinko Fake Gambling Site
* Wine Venue Database
* Where to Poo
* Math formula solver
* Word unscramble
* 21 questions?
* Would You Rather?

“Would You Rather?” Project ideas:

* Rank controversial/worst ideas
* Come up with good questions
* Allow user to submit questions
* About page

Discuss Project Ideas

* Project must be a web application
* Has a landing page which helps a user to understand what the application is about
* Besides the landing page, it has multiple other pages
* It has some data to maintain/manipulate

The Web App Idea that we have chosen is the Would You Rather? Database

* This Database will present the user with random would you rather questions
* The user will be prompted for their answer
* Answers will be stored, and stats will be available such as favorite choices or least favorites

SRS Document Plan: Going forward we will discuss and assign different sections of the document to each member of the team. As a team we will go over each section and review and edit as needed before finalizing.

To do for next meeting:

MainPage/Introduction, Update Appendix B - Tracy Hotchkiss

1.1 - Brian Hong

1.2 - Kelly St.Onge

1.3 - Lane Koistinen

--Meeting 10/13/2020

Team Present:

Derived from Discord Team Chat

The professor responded back to our project idea. It looks like we don't need to do the submit questions database. Just having the site that will produce the predefined sentences randomly and then saving and storing the stats is sufficient

Team members have reviewed the work done for the Introduction and sections 1.1 - 1.3 and provided feedback via the SRS Team 21 google doc

We have decided for now not to plan on adding item 3 of the introduction until it is clear if there will be sufficient time to do so as per the professor’s response this is not required.

To Do for next meeting:

Team members will review the next set of sections to get an idea of content to discuss for next meeting.

Team members will start thinking on which section they would like to write next.

Sections to review:

Sections: 2.1-2.4

Sections 1.1 - 1.3 will be updated on the master SRS document by Tracy.

--Meeting 10/15/2020

Team Present:

    Lane Koistinen

    Kelly St.Onge

    Brian Hong

We reviewed a little from section 2.1, we will review further at our next meeting.

We have decided to cut the meeting short and make the next one longer to discuss and review what we have written for the SRS so far. We each selected a part of section 2.

To do for next meeting:

2.1 - Brian Hong

2.2 - Brian Hong

2.3 - Kelly St.Onge

2.4 - Kelly St.Onge

2.5 - Lane Koistinen

2.6 - Lane Koistinen

SRS Document update/2.7 - Tracy Hotchkiss

--Meeting 10/20/2020

Team Present:

    Tracy Hotchkiss

    Lane Koistinen

    Kelly St.Onge

    Brian Hong

Reviewed comments and suggestions for changes to make to section 1.

Updated these sections in comments to reflect suggestions for edit.

Discussed assumptions and dependencies.

To do for next meeting:

All team members will review section 2 and comment suggestions.

3.1.1 - Tracy

3.1.2 - Tracy

3.1.3 - Brian

3.1.4 - Brian

3.2 - Kelly

3.3.1 - Lane

--Meeting 10/27/2020

Team Present:

Tracy Hotchkiss

Brian Hong

Lane Koistinen

Reviewed comments and suggestions for changes to make to section 2 and 3.

Updated suggested edits and comments.

Discussed how the database will be stored regarding section 3.1.3

Discussed minimum system requirements

Discussed assumptions and dependencies

Discussed design and implementation constraints

To do for next meeting:

4.1 - Brian

4.2 - Lane

4.3 - Kelly

1.4-1.6, Appendix B, Appendix A - Tracy

Update document Final edit is due Friday Nov. 6th – Tracy

--Meeting 11/05/2020

Team Present:

Derived from Discord Team Chat

Discussed Section 5, Appendix A, 4.3, and 2.1

Document is ready for final review and submission by team